ANL-7715

ANL-7715

The ARC System
One-dimensional Diffusion Theory Capability, DARC1D

D. E. Neal, G. K. Leaf, and A. S. Kennedy

PROPERTY OF
ARGONNE NATIONAL LAB
IDAHO LIBRARY

U of C-AUA-USAEC

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
Iowa State University
The University of Iowa
Kansas State University
The University of Kansas
Loyola University
Marquette University
Michigan State University
The University of Michigan
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame
The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
The University of Texas at Austin
Washington University
Wayne State University
The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or 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.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $6.00; Microfiche $0.95

ANL-7715
Reactor Technology

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

The ARC System
One-dimensional Diffusion Theory Capability, DARC1D

by

D. E. Neal, G. K. Leaf, and A. S. Kennedy*

Applied Mathematics Division

May 1971

*Now with the Argonne Center for Environmental Studies

PREFACE

This volume is one of a series describing the Argonne Reactor Computation (ARC) System, which has been executed on IBM System 360 computers, models 50, 75, 85, and 195, under the OS/360 operating system. The modules described herein must be used in conjunction with modules described in other volumes.

The ARC System consists of a library of computational modules for fast reactor design calculations, along with a set of system modules which provide an environment for the operation of the computational modules. The modules are used in a selected order by directive programs called "paths." Inter-module communication is through ARC System data sets. The data sets are defined such that output from one computational module can be used as input to others. This allows for the automation into a single run of what would normally be a sequence of runs.

ANL-7711 of this series contains descriptions of the ARC System history and philosophy, of the software environment in which the ARC System operates, of the system modules which constitute the software environment, and of the ARC System data sets.

A description of the ARC System standard paths can be found in ANL-7712.

The ARC System was previously described in report ANL-7332, "The Argonne Reactor Computation (ARC) System," B. J. Toppel, Ed. (November 1967).

CONTENTS OF VOLUMES IN THIS SERIES

ANL-7711 The System Aspects and Interface Data Sets of The Argonne Reactor Computation (ARC) System

ANL-7712 The ARC System Standard Paths

ANL-7713 The ARC System Neutronics Input Processor

Part One. Neutronics Input Processor Module, GNIP (NUI002)

Part Two. Geometry Data Set (GEOM) Copier, CPGEOM (AJC006)

ANL-7714 The ARC System Cross-section Homogenization and Modification Capabilities

Part One. Multigroup Cross-section Homogenization Specification Module, INHOMG (NUI001)

Part Two. Multigroup Cross-section Homogenization Calculation Module, HOMOG (NUC001)

Part Three. Material-to-Composition Cross-section Homogenization Calculation Module, HHOMOG (NUC007)

Part Four. Microscopic Cross-section Group Collapse Module, MICCOL (AJC007)

Part Five. Macroscopic Cross-section Group Collapse Module, MACCOL (AJC008)

Part Six. Run-time Microscopic Cross-section Modification Module, MODXSISO (NUI006)

ANL-7715 The ARC System One-dimensional Diffusion Theory Capability, DARC1D

Part One. One-dimensional, Multigroup Diffusion Theory Module for k-effective and Source Calculations, DIF1D (NUC002)

Part Two. One-dimensional, Multigroup Diffusion Theory Criticality Search Module, D1DSCH (NUC004)

ANL-7716 The ARC System Two-dimensional Diffusion Theory Capability, DARC2D

Part One. Two-dimensional, Multigroup Diffusion Theory Module for k-effective and Source Calculations, DIF2D (NUC005)

Part Two. Two-dimensional, Multigroup Diffusion Theory Criticality Search Module, D2DSCH (NUC006)

ANL-7717 The ARC System One-dimensional Transport Theory Capability, SNARC1D

Part One. One-dimensional, Discrete Ordinate Transport Theory Specification Module, SNIN1D (NUI005)

Part Two. One-dimensional, Discrete Ordinate Transport Theory Calculation Module, SNARC1D (NUC003)

ANL-7718 The ARC System Two-dimensional Transport Theory Capability

Part One. Two-dimensional, Discrete Ordinate Transport Theory Calculation Module, TARGET (NUC008)

Part Two. Two-dimensional, Discrete Ordinate Transport Theory Output Manipulation Module, OUTPUTR (NUE002)

ANL-7719 The ARC System One-dimensional Adjunct Calculations and Edits PARC1D (AJC003)

Part One. One-dimensional Diffusion Theory Perturbation Module, PARC1D (AJC003)

Part Two. One-dimensional Neutron Inventory Module, INVENT1D (AJC001)

Part Three. One-dimensional Neutronics Output Manipulation Module, OUTMAN1D (NUE001)

ANL-7720 The ARC System Two-dimensional Adjunct Calculations

Part One. Two-dimensional Diffusion Theory Perturbation Module, PARC2D (AJC004)

Part Two. Two-dimensional Neutron Inventory Module, INVENT2D (AJC002)

ANL-7721 The ARC System Fuel Cycle Analysis Capability, REBUS

Part One. Fuel Cycle Specification Module, INFUEL (FCI001)

Part Two. Fuel Cycle Analysis Module, FUELCY (FCC001)

ANL-7722 The ARC System Cross-section Generation Capabilities, ARC-MC^2

Part One. Eipthermal Cross-section Specification Module, INPUT

Part Two. Resonance Cross-section Calculation Module, REXSEC (CSC001)

Part Three. Nonresonant Cross-section and Fundamental-Mode Spectrum Calculation Module, SCSFM (CSC002)

Part Four. Broad-group Cross-section and Fundamental-Mode Spectrum Calculation Module, BCSFM (CSC003)

Part Five. Cross-section Data Set Editor Module, XSEDIT (CSE001)

Part Six. Delayed Data Preparation Module, DELDATA (CSI007)

The ANL topical reports ANL-7723 and ANL-7724 referenced in ANL-7713 and ANL-7717 will not be issued as part of this series.

TABLE OF CONTENTS

Page

ABSTRACT 13

PART ONE. ONE-DIMENSIONAL, MULTIGROUP DIFFUSION THEORY MODULE FOR K-EFFECTIVE AND SOURCE CALCULATIONS, DIF1D (NUC002) 15

1. DIF1D (NUC002) Introduction. 16
   a. Description of Capabilities. 16
   b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path. 17
   c. Data Sets Utilized by This Module. 20

2. DIF1D (NUC002) Description of Problem and Method of Solution . . . 21
   a. Introduction 21
   b. Derivation of the Finite-difference Equations. 21
   c. Solution of the Finite-difference Equations. 33
   d. The Solution of the Adjoint k_{eff} Calculation 46
   e. The Solution of the Source Calculation 47
   f. Output Edits 51
   g. References 63

3. DIF1D (NUC002) Description of Module Flow. 64
   a. DIF1D (NUC002) Overlay Structure Diagram 65
   b. Diagrammatic Subprogram Logic Flow 66
   c. Descriptive Subprogram Logic Flow. 69

4. DIF1D (NUC002) Subprogram Descriptions 74
   Subroutine BALCAL. 76
   Subroutine CELCAL. 78
   Subroutine CHEB. 88
   REAL Function DACOSH 90
   Subroutine DIF1D 91
   Subroutine DIMSCT. 93
   Subroutine D1DCAL. 95
   Subroutine D1DIN 96
   Subroutine D1DOUT. 97
   Subroutine ERRMSG. 100
   Subroutine EXTCAL. 102
   Subroutine FDCGEN. 103
   Subroutine FDC1D 105
   Subroutine FISCAL. 110
   Subroutine FLIP. 114
   Subroutine FLXIN 116
   Subroutine FVCAL 118
   Subroutine GETBND. 120
   Subroutine GETCHI. 123
   Subroutine IAREA 123
   Function IBAL. 125
   Function IBFJ1 128
   Subroutine IFSRC 129

TABLE OF CONTENTS (Contd.)

Page

Subroutine INDOT1. 131
Subroutine INDOT2. 138
Subroutine INDOUT. 143
Subroutine INVERT. 150
Function IPHVOL. 152
Subroutine ISOURC. 154
Subroutine ISSUBS. 158
Subroutine ISSWEP. 159
Subroutine ISUBS 161
Subroutine ISWEEP. 162
Subroutine ITSRC 165
Function IVOL. 168
Subroutine KEFCAL. 169
Subroutine LKCAL 171
Main Program for NUC002. 174
Subroutine NRMCAL. 175
Subroutine OUTERS. 180
Subroutine PAGHED. 184
Subroutine PCKSCT. 185
Subroutine PCOMP 188
Subroutine PGEOM 192
Subroutine PRTBC 194
Subroutine PWRCAL. 196
Subroutine RBLACK. 199
Subroutine RBNDC 200
Subroutine RCDEP 203
Subroutine RCOMP 208
Subroutine RDELAY. 211
Subroutine RDIST 213
Subroutine REMCAL. 215
Subroutine REORDR. 217
Subroutine RFLUX 220
Subroutine RGEOM 223
Subroutine RGROUP. 226
Subroutine RMESH 232
Subroutine RSORC 234
Subroutine RVEL. 239
Subroutine SCTCAL. 241
Subroutine SOUTER. 243
Subroutine SRCCAL. 246
Subroutine STORBC. 248
Subroutine SUMCAL. 249
Subroutine TWODCR. 252
Subroutine TWODFL. 254
Subroutine TWODPR. 256
Subroutine WFLX. 260
Subroutine WIPE. 261

TABLE OF CONTENTS (Contd.)

Page

5. DIF1D (NUC002) Common Block Descriptions 262
/BCBLK/. 263
/BLKSTR/ 263
/GNAME/. 263
/HEADER/ 264
/NAMBLK/ 264
/SINGLE/ 264

6. DIF1D (NUC002) Dynamic Storage Array Allocation. 273

7. DIF1D (NUC002) Data Set Utilization. 291

8. DIF1D (NUC002) Error Messages. 297

9. DIF1D (NUC002) Timing Information. 306

10. DIF1D (NUC002) Sample Output 308

PART TWO. ONE-DIMENSIONAL, MULTIGROUP DIFFUSION THEORY CRITICALITY SEARCH MODULE, D1DSCH (NUC004) 329

1. D1DSCH (NUC004) Introduction 330
a. Description of Capabilities. 330
b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path. 331
c. Data Sets Utilized by This Module. 334

2. D1DSCH (NUC004) Description of Problem and Method of Solution. . . 335
a. Introduction 335
b. Problem Formulation. 335
c. Method of Solution 336
d. Output Edits 339
e. References 341

3. D1DSCH (NUC004) Description of Module Flow 342
a. D1DSCH (NUC004) Overlay Structure Diagram. 343
b. Diagrammatic Subprogram Logic Flow 344
c. Descriptive Subprogram Logic Flow. 348

4. D1DSCH (NUC004) Subprogram Descriptions. 353
Subroutine CHANGE. 355
Subroutine CRTIN1. 360
Subroutine CRTIN2. 361
Subroutine DELALF. 362
Subroutine DELBSQ. 363
Subroutine DELX. 365
Subroutine DELXS 367

TABLE OF CONTENTS (Contd.)

Page

Subroutine D1DSCH. 372
Main Program for NUC004. 378
Subroutine PARAB 379
Subroutine RCORD1. 381
Subroutine RCORD2. 382
Subroutine RCORD3. 383
Subroutine RCORD4. 385
Subroutine RDBSQ 388
Subroutine RDCONC. 389
Subroutine RDDDX 390
Subroutine RDGEOM. 392
Subroutine SRCHIN. 394
Subroutine UDBHMG. 398
Subroutine UPDTXS. 403
Subroutine WRCRD1. 406
Subroutine WRCRD2. 407
Subroutine WRCRD3. 408
Subroutine WRCRD4. 410
Subroutine WSRCH 412

5. D1DSCH (NUC004) Common Block Descriptions. 419
/SINGLE/ 420

6. D1DSCH (NUC004) Dynamic Storage Allocation 430

7. D1DSCH (NUC004) Data Set Usage 451

8. D1DSCH (NUC004) Error Messages 458

9. D1DSCH (NUC004) Timing Information 472

10. D1DSCH (NUC004) Sample Output. 474

THE ARC SYSTEM ONE-DIMENSIONAL DIFFUSION THEORY CAPABILITY, DARC1D

by

D. E. Neal, G. K. Leaf, and A. S. Kennedy

ABSTRACT

This volume presents the documentation of the ARC System one-dimensional diffusion theory capability, DARC1D, comprising the k-effective and source calculation module DIF1D (NUC002) and the criticality search module D1DSCH (NUC004).

Both modules solve a finite-difference approximation to the one-dimensional multigroup diffusion equation. The DIF1D module can perform either inhomogeneous or homogeneous calculations, using either real or adjoint solutions. Four geometry types, rectangular slab, cylindrical slab, cylinder, or sphere, are permitted.

The criticality search module allows searches to a fixed value of k-effective or of the inverse reactor period α. The parameters which may be varied are composition cross section data, mesh interval length or region size, reactor buckling, and the inverse period α.

PART ONE.

ONE-DIMENSIONAL, MULTIGROUP DIFFUSION THEORY MODULE FOR K-EFFECTIVE AND SOURCE CALCULATIONS, DIF1D (NUC002)

Analysis Done By - G. K. Leaf and A. S. Kennedy

Programmed By - A. S. Kennedy and D. E. Neal

Documented By - D. E. Neal and G. K. Leaf

1. DIF1D (NUC002) Introduction

a. Description of Capabilities

This module solves a finite-difference approximation to the one-dimensional multigroup diffusion equation. Two basic types of calculations can be made:

(i) homogeneous eigenvalue (k_{eff}) calculation, in which the external boundary conditions are homogeneous and there is no external source; and

(ii) inhomogeneous (source) calculation, in which the external boundary conditions are inhomogeneous and/or there is an external source.

Real and/or adjoint solutions are permitted in the homogeneous case; real or adjoint solutions are permitted in the inhomogeneous case.

The fluxes may be saved during the iterations to permit a restart of the calculation.

b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path

Modules Utilized by Each Path	Paths Utilizing This Module			
	STP001 1-D Diffusion Theory	STP003 2-D Diffusion Theory	STP004 Burnup Diffusion Theory	STP006 1-D Diffusion Theory Perturbation
DELDATA (CSI007) Delayed Neutron Data Preparation				X
INHOMG (NUI001) Multigroup Cross Section Homog. Specifications	X	X	X	X
HOMOG (NUC001) Multigroup Cross Section Homog.	X	X	X	X
HHOMOG (NUC007) Material-to-Composition Cross Section Homogenization			X	
GNIP (NUI002) Neutronics Input Processor	X	X	X	X
DARC1D: DIF1D (NUC002) 1-D Diffusion Theory k_{eff} and Source Calculations	X	X	X	X

b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path (Cont'd)

	Modules Utilized by Each Path	Paths Utilizing This Module			
		STP001 1-D Diffusion Theory	STP003 2 -D Diffusion Theory	STP004 Burnup Diffusion Theory	STP006 1-D Diffusion Theory Perturbation
DARC1D	D1DSCH (NUC004) 1-D Diffusion Theory Criticality Search	X	X	X	X
DARC2D	DIF2D (NUC005) 2-D Diffusion Theory keff and Source Calculations			X	X
DARC2D	D2DSCH (NUC004) 2-D Diffusion Theory Criticality Search		X	X	
	MODXSISO (NUC006) Run-time Microscopic Cross Section Modification	X	X		
	PARC1D (AJC003) 1-D Diffusion Theory Perturbation				X

b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path (Cont'd.)

Modules Utilized by Each Path	Paths Utilizing This Module			
	STP001 1-D Diffusion Theory	STP003 2 -D Diffusion Theory	STP004 Burnup Diffusion Theory	STP006 1-D Diffusion Theory Perturbation
OUTMAN1D (NUE001) 1-D Neutronics Output Manipulation	X	X		X
INVENT1D (AJC001) 1-D Neutron Inventory	X	X		X
INVENT2D (AJC002) 2-D Neutron Inventory		X		
CPGEOM (AJC006) Geometry Data Set (GEOM) Copier				X
REBUS: INFUEL (FCI001) Fuel Cycle Specifications			X	
REBUS: FUELCY (FCC001) Fuel Cycle Analysis			X	

c. Data Sets Utilized by This Module

Upper-interface Data Sets	Records Read
A.DIF1D, One-dimensional Diffusion Theory Module-dependent BCD Input	All
BC, Boundary Condition Specifications	All
ES.D1D, One-dimensional Distributed External Source	All
FA.D1, One-dimensional Adjoint Group Fluxes	All
or	
FR.D1, One-dimensional Real Group Fluxes	All
GEOM, Geometry Data	All except Region Volumes
SP.CICN, Module-independent Data	All
XS.C.MIN, Macroscopic Composition Cross Sections	All
XS.ISO2, File 1, Problem-dependent Microscopic Group Cross-sections, File 1	File Size, Group Structure
or	
XS.ISO, File 1, Microscopic Group Cross Sections, File 1	File Size, Group Structure

Lower-interface Data Sets	Records Written
FA.D1, One-dimensional Adjoint Group Fluxes	All
and/or	
FR.D1, One-dimensional Real Group Fluxes	All
FR.PN, Power-normalized Real Regional Group Fluxes	All
FSA.D1, One-dimensional Adjoint Fission Source	All
and/or	
FSR.D1, One-dimensional Real Fission Source	All

2. DIF1D (NUC002) Description of Problem and Method of Solution

This module solves a finite-difference approximation to the one-dimensional, multigroup diffusion equation.

a. Introduction

Two basic types of calculations can be made, as follows:

(i) homogeneous (eigenvalue -k_{eff} calculation), in which the external boundary conditions are homogeneous and there is no external source; or

(ii) inhomogeneous (source calculation), in which the external boundary conditions are inhomogeneous and/or there is an external source.

Real and/or adjoint solutions are permitted in the homogeneous case; real or adjoint solutions are premitted in the inhomogeneous case.

The module has the following capabilities:

(i) linear external boundary conditions;

(ii) two types of conditions for internal boundaries, logarithmic for regions containing a black composition, or continuity of flux and normal current for all other cases;

(iii) four geometry types, spherical, cylindrical, rectangular slab, or cylindrical slab;

(iv) unrestricted scattering between energy groups; and

(v) composition-dependent χ-matrices or χ-vectors.

b. Derivation of the Finite-difference Equations

The method used to derive the finite-difference approximation to the diffusion equation in this module is the same as that used by Hassitt in CRAM[1] and later by Sangren in CANDID.

Let a reactor of length or outer radius R be given, depending on the geometry type, such that the boundaries of the regions making up the reactor are perpendicular to the transverse or radial axis. The multi-group diffusion system can then be written in the form

$$-\mathrm{div}\,(D^g(\bar{r})\,\nabla\phi_g(\bar{r})) + (\sigma_R)^g(\bar{r})\,\phi_g(\bar{r}) = \sum_{g'\neq g} \sigma^{g'g}(\bar{r})\,\phi_{g'}(\bar{r})$$

$$+ \frac{1}{k}\sum_{g'=1}^{G} \chi^{g'g}(\bar{r})\,(\nu\cdot\sigma_f)^{g'}(\bar{r})\,\phi_{g'}(\bar{r}), \text{ for } g = 1,2,\ldots,G. \qquad (b.1)$$

Here, $\phi_g(\bar{r})$ denotes the neutron flux in the gth energy group at the spatial point $\bar{r}$. The macroscopic cross sections $D^g(\bar{r})$, $(\sigma_R)^g(\bar{r})$, $\sigma^{g'g}(\bar{r})$, $\chi^{g'g}(\bar{r})$, and $(\nu\cdot\sigma_f)^{g'}(\bar{r})$ are presumed to be constant within a given region. It should be noted that the diffusion coefficient is taken to be $\dfrac{1}{3(\sigma_{tr})^g(\bar{r})}$, where $(\sigma_{tr})^g(\bar{r})$ is the macroscopic transport cross section.

At the exterior boundary surfaces, we shall assume the boundary conditions are of the form

$$A^g(\bar{r})\nabla\phi^g(\bar{r}) \cdot \hat{n} + B^g(\bar{r})\,\phi_g(\bar{r}) = 0, \text{ for } g = 1,2,\ldots,G, \qquad (b.2)$$

where $\hat{n}$ denotes the unit outward normal to the surface.

The object in a real k_{eff} calculation is to find the largest value of k and the corresponding set of non-negative fluxes $\{\phi_g(\bar{r})\}$ satisfying Eqs. (b.1) and (b.2). In the adjoint case, the solution to the adjoint system associated with these equations is sought.

A source-type calculation can arise in two ways: (i) if an external source term $S^g(\bar{r})$ is added to the right-hand side of Eq. (b.1), and if the boundary conditions of Eq. (b.2) are either left in their homogeneous form or modified by replacing the right-hand side by a term $C^g(\bar{r})$; or (ii) if the term $C^g(\bar{r})$ is inserted in Eq. (b.2) without the addition of an external source term in Eq. (b.1). In the source cases, either a fixed k is specified or k is set equal to 1.0, and the object is to find the non-negative flux satisfying Eqs. (b.1) and (b.2). In the adjoint source case, the solution to the adjoint system associated with these equations is sought.

To approximate the solution of the system by means of finite differences, we lay down a mesh structure on the reactor, such that all region boundaries lie on mesh lines. The abscissas of the vertical

mesh lines can be denoted by $0 = R_0 < R_1 < R_2 < \cdots < R_I = R$, where the region defined by $R_{i-1} < r < R_i$ with center at r_i will be called the ith mesh cell or mesh interval, and the point r_i will be considered to be the mesh point associated with the interval. Then $\Delta R_i = R_i - R_{i-1}$.

Consider the differential Eq. (b.1) in the ith cell; then Eq. (b.1) becomes

$$- \int_{V_i} \mathrm{div}(D_i^g \cdot \nabla\phi_g(\bar{r}))\,dV + (\sigma_R)_i^g \int_{V_i} \phi_g(\bar{r})\,dV$$

$$= \sum_{g' \neq g} \sigma_i^{g'g} \int_{V_i} \phi_g(\bar{r})\,dV + \frac{1}{k} \sum_{g'=1}^{G} \chi_i^{g'g} (\nu \cdot \sigma_f)_i^{g'} \int_{V_i} \phi_{g'}(\bar{r}) \quad . \qquad \text{(b.3)}$$

Here, let $\phi_{i,g} = \phi_g(r_i)$; then we make the approximation

$$\int_{V_i} \phi_g(\bar{r})\,dV \cong \phi_{i,g}\, V_i \quad . \qquad \text{(b.4)}$$

To the term involving the divergence, we apply Green's theorem to obtain

$$\int_{V_i} \mathrm{div}(D_i^g \cdot \nabla\phi_g(\bar{r}))\,dV = A_i\, D_i^g\, \phi'(R_i^-) - A_{i-1}\, D_i^g\, \phi'(R_{i-1}^+) \ , \qquad \text{(b.5)}$$

where A_i is the surface area per unit height or radius of the right face of the ith cell.

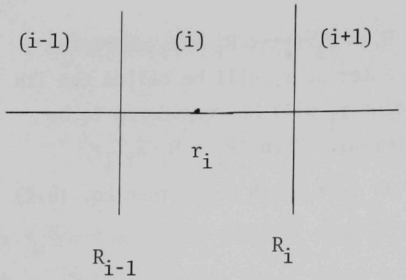
There are three cases to consider in the approximation of the right-hand side of Eq. (b.5):

(i) the ith cell is a "normal" interior cell, i.e., one for which the ordinary conditions of continuity of flux and current can be applied across the boundaries;

(ii) the cell is an exterior cell, such that one of its boundaries is on an external boundary of the reactor;

or (iii) the cell is an "abnormal" interior cell in the sense that it is adjacent to a cell containing a black composition.

The interior mesh cell is shown in the diagram below.



We shall first consider case (i) for each geometry. Note that the energy subscript has been dropped.

For this case, the interface conditions take the form

$$\left.\begin{aligned} &\phi(R_i^-) = \phi(R_i^+) \quad ; \\ &D_i \frac{d\phi}{dr}\bigg|_{r=R_i^-} = D_{i+1} \frac{d\phi}{dr}\bigg|_{r=R_i^+} \end{aligned}\right\} \quad . \tag{b.6}$$

If we expand the flux and its first derivative about the points R_i and R_{i+1}, we obtain

$$\phi(R_i^-) = \phi_i + \frac{1}{2}\Delta R_i\, \phi_i' + \ldots \; ;$$

$$\phi(R_i^+) = \phi_{i+1} - \frac{1}{2}\Delta R_{i+1}\, \phi'_{i+1} + \ldots \; ;$$

$$\phi'(R_i^-) = \phi_i' + 2(\frac{1}{2}\Delta R_i)\, \phi_i'' + \ldots \; ;$$

$$\phi'(R_i^+) = \phi_{i+1}' - 2(\frac{1}{2}\Delta R_{i+1})\, \phi_{i+1}'' + \ldots \; .$$

Solving the first set for ϕ_i and ϕ_{i+1} and using the second set to solve for ϕ_i' and ϕ_{i+1}', we obtain

$$\phi_i = \phi(R_i^-) - \frac{1}{2}\Delta R_i\, \phi'(R_i^-) + \text{higher-order terms};$$

$$\phi_{i+1} = \phi(R_i^+) + \frac{1}{2}\Delta R_{i+1}\, \phi'(R_i^+) + \text{higher-order terms}.$$

Using the interface conditions $\phi(R_i^-) = \phi(R_i^+)$ and $D_i\phi'(R_i^-) = D_{i+1}\,\phi'(R_i^+)$, we multiply the first equation by $\frac{1}{2}\,\Delta R_{i+1}\,D_i$ and the second by $\frac{1}{2}\,\Delta R_i\,D_{i+1}$, then add to obtain, after neglecting terms of order higher than the first,

$$\phi(R_i) \cong \frac{\frac{1}{2}\,\Delta R_i\,D_{i+1}\,\phi_{i+1} + \frac{1}{2}\,\Delta R_{i+1}\,D_i\,\phi_i}{\frac{1}{2}\,\Delta R_i\,D_{i+1} + \frac{1}{2}\,\Delta R_{i+1}\,D_i} \quad . \tag{b.7}$$

Putting this expression for $\phi(R_i)$ into the equation for ϕ_i and solving the resulting equation for $D_i\,\phi'(R_i^-)$, we obtain

$$D_i\,\phi'(R_i^-) \cong \frac{D_i\,D_{i+1}}{\frac{1}{2}\,\Delta R_i\,D_{i+1} + \frac{1}{2}\,\Delta R_{i+1}\,D_i}\,(\phi_{i+1} - \phi_i) \quad . \tag{b.8}$$

Hence, for the boundary R_i,

$$A_i\,D_i\,\phi'(R_i^-) \cong \frac{D_i\,D_{i+1}}{\frac{1}{2}\,\Delta R_i\,D_{i+1} + \frac{1}{2}\,\Delta R_{i+1}\,D_i}\,(\phi_{i+1} - \phi_i)\,A_i \quad , \tag{b.9}$$

where A_i is the surface area per unit height or radius, depending on the geometry type. The other cell boundary is handled in a similar manner.

For the case of spherical geometry, we assume that the reactor configuration is spherically symmetric, such that the flux $\phi(\rho,\theta,\phi) = \phi(\rho)$. From this,

$$\left.\begin{aligned} &\mathrm{div}\,(D\cdot\nabla\phi) = \frac{1}{\rho^2}\frac{d}{d\rho}\left(\rho^2\,D\,\frac{d\phi}{d\rho}\right) \text{ and} \\ &dV = 4\pi\,\rho^2\,d\rho \quad . \end{aligned}\right\} \tag{b.10}$$

For a "normal" interior cell, then, integrating the diffusion equation over the interval, we obtain

$$-4\pi[R_i^2\,D_i\phi'(R_i^-) - R_{i-1}^2\,D_i\phi'(R_{i-1}^+)] + \sigma_{R_i}\,V_i\,\phi_i = S_i\,V_i \quad , \tag{b.11}$$

where

$$V_i = \frac{4\pi}{3} (R_i^3 - R_{i-1}^3) ,$$

$$\phi_i = \phi(r_i), \; r_i = \frac{R_i + R_{i-1}}{2} ,$$

and S_i represents the fission and scattering source terms.

Here we have made the approximation

$$4\pi \int_{R_{i-1}} \phi(\rho) \, \rho^2 \, d\rho \cong \phi(r_i) \, V_i .$$

(Alternatively, we could apply the mean-value theorem, and identify the mean value ϕ with the value of the flux at the center.)

Applying the conditions of continuity of flux and current across interfaces leads to

$$- (a_i \, \phi_{i+1} + c_i \, \phi_{i-1}) + e_i \, \phi_i = S_i \, V_i , \tag{b.12}$$

where

$$\left.\begin{aligned} a_i &= \frac{D_i \, D_{i+1} \, A_i}{\frac{1}{2} \Delta R_i \, D_{i+1} + \frac{1}{2} \Delta R_{i+1} \, D_i} ; \\ c_i &= \frac{D_i \, D_{i-1} \, A_{i-1}}{\frac{1}{2} \Delta R_i \, D_{i-1} + \frac{1}{2} \Delta R_{i-1} \, D_i} = a_{i-1} ; \\ \text{and} \quad e_i &= a_i + c_i + (\overline{\sigma_R})_i \, V_i . \end{aligned}\right\} \tag{b.13}$$

Here, $\Delta R_i = R_i - R_{i-1}$,
and, for spherical geometry, A_i is simply the surface area of a sphere of radius R_i, $A_i = 4\pi R_i^2$. The removal term $(\overline{\sigma_R})_i \equiv (\sigma_R)_i$.

For cylindrical geometry, we have

$$\left.\begin{aligned} &\mathrm{div}\ (D\cdot\nabla\phi) = \frac{1}{\rho}\left(\rho D \frac{d\phi}{d\rho}\right) \quad \text{and} \\ &dV = 2\pi\ \rho H_r\ d\rho \quad , \end{aligned}\right\} \tag{b.14}$$

where the flux is assumed to be independent of the θ-direction and to have a cosine distribution in the z-direction, $\phi(\rho,\theta,z) = \phi(\rho)\cos\frac{\pi z}{2H_r}$, and where H_r is the extrapolated half-height of the reactor in region r.

Integrating the diffusion equation over an interval, as in the spherical case, we find that the finite difference equations are defined as in Eq. (b.13), where the volume per unit height in the z-direction is $V_i = \pi(R_i^2 - R_{i-1}^2)$. Hence, the surface area per unit height is $A_i = 2\pi\ R_i$.

The removal term $(\overline{\sigma_R})_i$ used in computing e_i is, for finite cylindrical geometry,

$$(\overline{\sigma_R})_i = [(\sigma_R)_i + D_i\ B_r^2] \quad , \tag{b.15}$$

where B_r^2 is the buckling in region r,

$$B_r^2 = \left(\frac{\pi}{2H_r}\right)^2 \quad .$$

For infinite cylindrical geometry, again $(\overline{\sigma_R})_i \equiv (\sigma_R)_i$.

For rectangular slab geometry, we have

$$\left.\begin{aligned} &\mathrm{div}(D\cdot\nabla\phi) = \frac{d}{dx}\left(D\frac{d\phi}{dx}\right) \\ &\text{and}\quad dV = dx\ , \end{aligned}\right\} \tag{b.16}$$

where the flux is assumed to have a cosine distribution in the y- and z-directions,

$$\phi(x,y,z) = \phi(x)\cos\frac{\pi y}{2H_{r,1}}\cos\frac{\pi z}{2H_{r,2}}\ ,$$

where $H_{r,1}$ and $H_{r,2}$ are the extrapolated half-heights of the reactor in the y- and z-directions, respectively.

Again, the finite-difference coefficients are found to be defined as in Eq. (b.13), where the volume per unit height in the y- and z-directions is simply $V_i = X_i - X_{i-1} = \Delta X_i$, and where ΔR_i is replaced by ΔX_i. Hence, the surface area per unit heights is $A_i = 1$. The removal term $(\overline{\sigma_R})_i$ now includes the buckling terms in the transverse and axial directions:

$$(\overline{\sigma_R})_i = [(\sigma_R)_i + D_i(B^2_{r,1} + B^2_{r,2})] \quad , \tag{b.17}$$

where $B^2_{r,1} = (\frac{\pi}{2H_{r,1}})^2$, $B^2_{r,2} = (\frac{\pi}{2H_{r,2}})^2$.

Finally, for the case of cylindrical slab geometry, we have

$$\left.\begin{aligned} &\text{div } (D\cdot\nabla\phi) = \frac{d}{dz}\left(D\,\frac{d\phi}{dz}\right) \\ &\text{and} \quad dV = \pi dz, \end{aligned}\right\} \tag{b.18}$$

where the flux is assumed to be independent of the θ-direction and to have a Bessel function distribution in the ρ-direction, $\phi(\rho,z,\theta) = \phi(z)\, J_0\,(\frac{\alpha\rho}{R_r})$. Here R_r is the extrapolated radius of the reactor and α is the first zero of the zero-th order Bessel function of the first kind, J_0.

Once again, the finite-difference coefficients are defined as in Eq. (b.13), where the volume per unit radius is $V_i = \pi(Z_i - Z_{i-1}) = \pi\Delta Z_i$ and the surface area per unit radius is thus π, and where ΔZ_i replaces ΔR_i. The removal term again includes the buckling contribution:

$$(\overline{\sigma_R})_i = [(\sigma_r)_i + D_i\, B^2_r] \ , \tag{b.19}$$

where $B^2_r = (\frac{\alpha}{R_r})^2$.

We shall next consider case (ii), that of an exterior cell. At the right-hand reactor boundary, a boundary condition of the form

$$A\phi'(R_I) + B\phi(R_I) = C \tag{b.20}$$

is applied, where A, B, and C are group-dependent. If we again expand the flux and its derivatives in a Taylor series, we have

$$\phi(R_i) = \phi_i + \frac{1}{2}\Delta R_i\ \phi_i' + (\frac{1}{2}\ R_i)^2\ \phi_i'' + \ldots;$$

$$\phi'(R_i) = \phi_i' + 2(\frac{1}{2}\Delta R_i)\ \phi_i'' + \ldots \quad .$$

Using Eq. (b.20), we obtain

$$C = B\phi_i + (A + \frac{1}{2}\Delta R_i\ B)\ \phi_i' + [2A(\frac{1}{2}\Delta R_i) + B(\frac{1}{2}\Delta R_i)^2]\ \phi_i' + \ldots,$$

where i = I in this case.

Thus, eliminating ϕ_i' between this equation and the expansion for $\phi'(R_i)$, we have

$$(C - B\phi_i) - (A + \frac{1}{2}\Delta R_i\ B)\ \phi'(R_i) = -\frac{1}{2}(\Delta R_i)^2\ B\phi_i'' + \ldots \quad .$$

Neglecting terms of order higher than the first, we obtain

$$\phi'(R_i) \cong \frac{C - B\ \phi_i}{A + \frac{1}{2}\Delta R_i\ B} \quad .$$

Finally, then,

$$A_i\ D_i\ \phi'(R_i^-) \cong \frac{D_i\ (C - B\ \phi_i)\ A_i}{A + \frac{1}{2}\Delta R_i\ B} \quad , \tag{b.21}$$

where A_i is again the surface area associated with the geometry type.

Here we see how inhomogeneous terms

$$\frac{D_i C A_i}{A + \frac{1}{2}\Delta R_i\ B}$$

arise in the difference equation when inhomogeneous boundary conditions are applied.

For spherical coordinates, the left boundary condition is always $\phi'(0) = 0$; thus, $c_1 = 0$. The right boundary conditions are of the form (b.20). Applying this condition in the last interval leads to

$$\left.\begin{aligned} & a_I = 0 \\ \text{and} \quad & e_I = \frac{D_I\, B\, A_I}{A + \frac{1}{2}\Delta R_I\, B} + c_I + (\overline{\sigma_R})_I\, V_I \end{aligned}\right\} \qquad \text{(b.22)}$$

where V_I, ΔR_I, A_I, and $(\overline{\sigma_R})_I$ are defined as in case (i). In addition, if $C \neq 0$, an external source term having the form

$$S_{ext} = \frac{D_I\, C\, A_I}{A + \frac{1}{2}\Delta R_I\, B} \qquad \text{(b.23)}$$

is created. Hence, if $C \neq 0$, the problem takes the form of a source calculation rather than a k_{eff} calculation.

In cylindrical geometry, the left-hand boundary condition is again $\phi'(0) = 0$, which leads to $c_1 = 0$, and the right-hand boundary condition is again of the form (b.20); hence, eq. (b.22) holds, where V_I, ΔR_I, and $(\overline{\sigma_R})_I$ are defined as for case (i). The external source term, if $C \neq 0$, again has the form (b.23).

In rectangular slab geometry, both the left and right boundaries can have general linear boundary conditions of the form (b.20). Hence,

$$
\left.
\begin{aligned}
& c_1 = 0 \\
& \text{and } e_1 = a_1 + \frac{D_1 B' A_1}{A' + \frac{1}{2}\Delta X_1 B'} + (\overline{\sigma_R})_1 V_1 \quad ; \\
& a_I = 0 \\
& \text{and } e_I = \frac{D_I B A_I}{A + \frac{1}{2}\Delta X_I B} + C_I + (\overline{\sigma_R})_I V_I \quad ; \\
& \text{if} \quad C' \neq 0, \\
& \qquad S_{ext,1} = \frac{D_1 C' A_1}{A' + \frac{1}{2}\Delta X_1 B'} \quad ; \\
& \text{if} \quad C \neq 0, \\
& \qquad S_{ext,I} = \frac{D_I C A_I}{A + \frac{1}{2}\Delta X_I B} \quad .
\end{aligned}
\right\} \qquad \text{(b.24)}
$$

Here, V_i, X_i, A_i, and $(\overline{\sigma_R})_i$ are defined as for case (i).

Similarly, for cylindrical slab geometry both the left and right boundaries can have the general linear boundary conditions. Thus, the finite-difference coefficients c_1, e_1, a_I, and e_I, as well as the external source terms $S_{ext,1}$ and $S_{ext,I}$, are computed as in Eq. (b.24), except that ΔX_i is replaced by ΔZ_i. The quantities V_i, A_i, and $(\overline{\sigma_R})_i$ are again defined as for case (i).

Finally, we shall consider case (iii), in which cell i is adjacent to a cell containing a black composition. Suppose that the (i + 1)th cell is "black" with respect to the flux in a given energy group. Then the region interior to this cell is exterior to the domain of the differential equation for that group; moreover, on the boundary a condition of the form

$$D_i \, \phi'(R_i^-) + \Lambda_{r,\ell} \, \phi(R_i) = 0 \qquad \text{(b.25)}$$

is applied. Here, Λ is an input constant, r is the number of the reactor region to which the interval $[R_i, R_{i+1}]$ belongs, and ℓ indicates that the condition is on the left. Hence, in the ith cell, for spherical geometry, we have

$$\left.\begin{aligned} a_i &= 0 \\ \text{and} \quad e_i &= \frac{D_i \Lambda_{r,\ell} A_i}{D_i + \frac{1}{2} \Delta R_i \Lambda_{r,\ell}} + c_i + (\overline{\sigma_R})_i V_i \quad , \end{aligned}\right\} \qquad \text{(b.26)}$$

where V_i, A_i, R_i and $(\overline{\sigma_R})_i$ are defined as for case (i).

The same equations apply for the cylindrical, rectangular slab, and cylindrical slab geometries, with V_i, A_i, ΔR_i (or ΔX_i or ΔZ_i), and $(\overline{\sigma_R})_i$ defined according to geometry type.

Note that all remarks apply only to the energy group for which the cell appears "black"; the same cell may be normal for other energy groups. Since there is no diffusion in the black cell, we set $D_i = 0$; this yields

$$\left.\begin{aligned} a_{i+1} &= c_{i+1} = 0 \quad ; \\ e_{i+1} &= (\overline{\sigma_R})_{i+1} V_{i+1} \quad . \end{aligned}\right\} \qquad \text{(b.27)}$$

If, on the other hand, we suppose that the interval $[R_{i-1}, R_i]$ is the outermost cell of a region black to neutrons in a given group, then, for spherical geometry,

$$\left.\begin{aligned} c_i &= 0 \\ \text{and} \quad & \\ e_i &= a_i + \frac{D_i \Lambda_{r,r} A_i}{D_i + \frac{1}{2} \Lambda_{r,r} \Delta R_i} + (\overline{\sigma_R})_i V_i \quad . \end{aligned}\right\} \qquad \text{(b.28)}$$

The definitions for the other geometry types are analogous.

Note that introducing black regions has not destroyed the symmetry of the coefficient matrix, i.e., $a_{i,g} = c_{i+1,g}$. It has simply permitted the possibility that some of the coefficients $a_{i,g}$ are zero. We shall, however, continue to treat the matrix as nonsymmetric, since more general interface conditions would lead to that situation.

c. Solution of the Finite-difference Equations

In section b, we derived the finite-difference equations associated with the one-dimensional, multigroup diffusion equation. Thus, starting with the system

$$-\mathrm{div}(D^g \nabla \phi_g) + (\sigma_R)^g \phi_g = \sum_{g' \neq g} \sigma^{g'g} \phi_{g'} + \frac{1}{k} \sum_{g'=1}^{G} \chi^{g'g} (\nu\sigma_f)^{g'} \phi_{g'}$$

$$\text{for } g = 1,2,\ldots,G,$$

with external boundary conditions of the form

$$A_g \frac{d\phi_g}{d_n} + B_g \phi_g = 0,$$

we obtained the homogeneous finite-difference system

$$-a_{i,g}\, \phi_{i+1,g} + e_{i,g}\, \phi_{i,g} - c_{i,g}\, \phi_{i-1,g} = \sum_{g' \neq g} \sigma_i^{g'g} \phi_{i,g'} V_i + \frac{1}{k} \sum_{g'=1}^{G} \chi_i^{g'g} (\nu \cdot \sigma_f)_i^{g'} \phi_{i,g'} V_i \qquad \text{(c.1)}$$

$$\text{for } g = 1,2,\ldots,G.$$

Here, we seek the largest positive value of k for which the system has a positive solution $\{\phi_{i,g}\}$.

If the external boundary conditions were inhomogeneous, or if an external source were present, the finite-difference equations would take the form

$$-a_{i,g}\, \phi_{i+1,g} + e_{i,g}\, \phi_{i,g} - c_{i,g}\, \phi_{i-1,g} = \sum_{g' \neq g} \sigma_i^{g'g} \phi_{i,g'} V_i + \sum_{g'=1}^{G} \chi_i^{g'g} (\nu \cdot \sigma_f)_i^{g'} \phi_{i,g'} V_i + S_{i,g} \qquad \text{(c.2)}$$

$$\text{for } g = 1,2,\ldots,G.$$

Here, $S_{i,g}$ represents the source term arising from the boundary conditions and/or the external source. The basic problem is to solve the set of linear equations for the flux $\{\phi_{i,g}\}$.

Two types of problems, real or adjoint, are performed for either k_{eff} or source calculations. We shall discuss the real solution to the k_{eff} calculation first.

We first define the following matrices:

$$A_g = \begin{bmatrix} e_{1g} & -a_{1g} & & & \bigcirc \\ -c_{2g} & e_{2g} & -a_{2g} & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & -a_{I-1,g} \\ \bigcirc & & & -c_{Ig} & e_{Ig} \end{bmatrix} \quad \text{for } g = 1,2,\ldots,G; \tag{c.3}$$

$$B^{g'g} = \text{diag}\left[V_i \sigma_i^{g'g}\right] ; \tag{c.4}$$

$$F^{g'g} = \begin{Bmatrix} \text{diag}\,[\chi_i^{g'g}\,(\nu\cdot\sigma_f)_i^{g'}\,V_i] & \text{for the } \chi\text{-matrix case} \\ \text{diag}\,[\chi_i^{g}\,(\nu\cdot\sigma_f)_i^{g'}\,V_i] & \text{for the } \chi\text{-vector case} \end{Bmatrix} \tag{c.5}$$

With these matrices defined, the basic finite-difference system given in Eq. (c.2) can be written in the form

$$A_g\phi_g = \sum_{g'\neq g} B^{g'g}\phi_{g'} + \frac{1}{k}\sum_{g'=1}^{G} F^{g'g}\,\phi_{g'} \qquad g = 1,2,\ldots,G, \tag{c.6}$$

where

$$\phi_g = (\phi_{1g}, \phi_{2g}, \ldots, \phi_{IG})' .$$

With the logarithmic condition applied at some interfaces, the matrices A_g are symmetric and tridiagonal. Some of the off-diagonal elements are

zero; i.e., $a_{i,g} = c_{i+1,g} = 0$ for some i and g. However, we shall assume that all the elements e_{ig} are positive. Thus each matrix A_g is nonsingular with a nonnegative inverse.

We next define the following matrices:

$$E = \begin{bmatrix} A_1 & -B^{21} & \cdots & -B^{G1} \\ -B^{12} & A_2 & \cdots & -B^{G2} \\ \vdots & & \ddots & \vdots \\ -B^{1G} & \cdots & \cdots & A_G \end{bmatrix} , \qquad (c.7)$$

and

$$F = \begin{bmatrix} F^{11} & F^{21} & \cdots & F^{G1} \\ F^{12} & F^{22} & \cdots & F^{G2} \\ \vdots & & \ddots & \vdots \\ F^{1G} & \cdots & \cdots & F^{GG} \end{bmatrix} . \qquad (c.8)$$

Then the system defined by Eq. (c.6) takes the form

$$E\phi = \frac{1}{k} F\phi, \qquad (c.9)$$

where $\phi = (\phi_1, \phi_2, \ldots, \phi_G)'$.

Now, since

$$(\sigma_R)^g = (\sigma_c)^g + (\sigma_f)^g + \Sigma\sigma^{gg'} ,$$

we have

$$e_{i,g} > \sum_{g' \neq g} \sigma_i^{gg'} V_i .$$

Thus E^{-1} exists and is non-negative. The basic problem, then, is to find the largest value of k and the corresponding non-negative flux that satisfy Eq. (c.9).

The basic method of solution used in this problem is that of power iteration on the matrix $E^{-1}F$. Thus starting with an arbitrary positive flux $\phi^{(0)}$ and a positive scalar $k^{(0)}$, we form iteratively

$$E\phi^{(n+1)} = \frac{1}{k^{(n)}} F\phi^{(n)}, k^{(n+1)} = k^{(n)} \frac{\left(\phi^{(n+1)}, \phi^{(n+1)}\right)}{\left(\phi^{(n+1)}, \phi^{(n)}\right)}, \quad n = 0, 1, 2 \ldots . \tag{c.10}$$

To find $\phi^{(n+1)}$, we must solve a matrix equation. The solution can be effected directly if there is no up-scattering present, since E is block lower triangular in this case.

If up-scattering is present, the solution is effected iteratively by means of block Gauss-Seidel applied to the blocks exhibited in Eqs. (c.7) and (c.8). The iterations by which we find $\phi^{(n+1)}$ are called inner iterations. Thus, if we have $\phi^{(n)}$ and $k^{(n)}$, and we also have the vectors $x_1^{(\ell)}, x_2^{(\ell)}, \ldots, x_{g-1}^{(\ell)}$, then we find $x_g^{(\ell)}$ by solving

$$A_g x_g^{(\ell)} = \sum_{g'<g} B^{g'g} x_{g'}^{(\ell)} + \sum_{g'>g} B^{g'g} x_{g'}^{(\ell-1)} + \frac{1}{k^{(n)}} \sum_{g'=1}^{G} F^{g'g} \phi_{g'}^{(n)}, \tag{c.11}$$

where we take $x_g^{(0)} \equiv \phi_g^{(n)}$ for all g. Since each matrix A_g^{-1} exists and is nonnegative and each matrix $B^{g'g}$ is also nonnegative, the sequence of vectors $x_g^{(\ell)}$ will converge to the solution $\phi_g^{(n+1)}$ of Eq. (c.10) for each $g = 1, 2, \ldots, G$.[2] As noted before, if there is no up-scattering, i.e., if $B^{g'g} = 0$ for $g'>g$, then no iteration is necessary. The above inner iteration is continued until

$$\sum_{g=1}^{G} \sum_{i=1}^{I} \left| x_{i,g}^{(\ell)} - x_{i,g}^{(\ell-1)} \right| < \varepsilon_I \text{ or } \ell > \ell_{max}, \tag{c.12}$$

at which point we set $\phi_{i,g}^{(n+1)} = x_{i,g}^{(\ell)}$.

In practice, the iteration scheme defined by Eq. (c.11) is slightly modified to reduce unnecessary steps. We define $\bar{g}$ to be the smallest value of g into which there is any up-scattering, taking into account all compositions. Then we set

$$\phi_g^{(n+1)} = x_g^{(1)}$$

for $1 \leq g < \bar{g}$. Then, for $\bar{g} \leq g \leq G$, we have the iteration procedure

$$A_g x_g^{(\ell)} = \sum_{\bar{g} \leq g' < g} B^{g'g} x_{g'}^{(\ell)} + \sum_{G > g' > g} B^{g'g} x_{g'}^{(\ell-1)} + \sum_{g'=1}^{\bar{g}-1} B^{g'g} \phi_{g'}^{(n+1)} \tag{c.13}$$

$$+ \frac{1}{k^{(n)}} \sum_{g'=1}^{G} F^{g'g} \phi_g^{(n)}, \quad \text{for } g = \bar{g}, \ldots, G.$$

Each matrix A_g is an I x I tridiagonal matrix to which we apply a slightly modified Choleski algorithm[3] to solve each matrix equation defined by Eq. (b.42) or (b.44). In discussing this algorithm, we shall drop the group index g. Thus our problem is to find the solution x to the linear system

$$Ax = f, \tag{c.14}$$

where A has the form

$$A = \begin{bmatrix} e_1 & -a_1 & & & \bigcirc \\ -c_2 & e_2 & -a_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -a_{I-1} \\ \bigcirc & & & -c_I & e_I \end{bmatrix}$$

with $e_i > 0$, $a_i \geq 0$, and $c_i \geq 0$. The matrix A is factored into the product of a lower and an upper triangular matrix A = LU. Letting $\omega = Ux$, we can write the system Ax = f as

$$\begin{aligned} L\omega &= f, \\ Ux &= \omega. \end{aligned} \tag{c.15}$$

The matrix U has the form

$$U = \begin{bmatrix} 1 & -\gamma_1 & & & \bigcirc \\ & 1 & -\gamma_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -\gamma_{I-1} \\ \bigcirc & & & & 1 \end{bmatrix}, \tag{c.16}$$

and the elements $\{\gamma_i\}_{i=1}^{I-1}$ are stored. The sequence $\{\omega_i\}_{i=1}^{I}$ is found each time a new f is given. The sequences γ_i and ω_i can be generated in the following way:

$$\gamma_1 = a_1/e_1,\ \gamma_i = \frac{a_i}{e_i - c_i\gamma_{i-1}},\quad \text{for } i = 2, \ldots, I-1. \tag{c.17}$$

The sequence $\{\omega_i\}$ can then be found from

$$\omega_1 = f_1/e_1, \omega_i = \frac{f_i + c_i\omega_{i-1}}{e_i - c_i\gamma_{i-1}},\quad i = 2, 3, \ldots, I. \tag{c.18}$$

Since the sequence γ_i is stored rather than e_i, we would rather generate ω_i from

$$\omega_1 = \frac{\gamma_1}{a_1} f_1,\ \omega_i = \frac{\gamma_i}{a_i}[f_i + c_i\omega_{i-1}],\quad i = 2, 3 \ldots, I, \tag{c.19}$$

with $a_I = 1$ and

$$\gamma_I = \frac{1}{e_I - c_I\gamma_{I-1}}.$$

However, if i_0 is a black cell, then $a_{i_0-1} = 0 = c_{i_0} = a_{i_0}$. Thus $\gamma_{i_0-1} = \gamma_{i_0} = 0$ and Eq. (b.50) will yield an indeterminant form. Moreover, the solution of $Ux = \omega$, that is, the backward sweep, is given by

$$x_I = \omega_I,\ x_i = \omega_i + \gamma_i x_{i+1},\quad \text{for } i = I-1, I-2, \ldots, 1. \tag{c.20}$$

This algorithm is valid for black cells, i.e., for $\gamma_{i_0-1} = \gamma_{i_0} = 0$. Thus, we shall use Eq. (c.19) if $a_i \neq 0$, and Eq. (c.18) if $a_i = 0$. This means we have to store $\{e_i : a_i = 0\}$.

The outer iterations are terminated when the following criteria are satisfied:

$$\frac{\sum_{g=1}^{G} \sum_{i=1}^{I} |\phi_{i,g}^{(n+1)} - \phi_{i,g}^{(n)}|}{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n)}} < \varepsilon_\phi \ , \tag{c.21}$$

$$|k^{(n+1)} - k^{(n)}| < \varepsilon_k \ , \tag{c.22}$$

and $(\overline{\lambda}^{(n+1)} - \underline{\lambda}^{(n+1)}) < \varepsilon_\lambda$, (c.23)

where

$$\overline{\lambda}^{(n+1)} = \max_{i,g} \left(\frac{\phi_{i,g}^{(n+1)}}{\phi_{i,g}^{(n)}} \right), \quad \underline{\lambda}^{(n+1)} = \min_{i,g} \left(\frac{\phi_{i,g}^{(n+1)}}{\phi_{i,g}^{(n)}} \right), \tag{c.24}$$

and the extrema are taken over those elements $\phi_{i,g}^{(n)}$ such that $\phi_{i,g}^{(n)} > \phi_{ng\ell}$, where $\phi_{ng\ell}$ is some input factor times the quantity $\sqrt{\dfrac{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n)}}{I \cdot G}}$.

Thus, if the ratio of the flux $\phi_{i,g}$ at any point i and group g to the root mean square flux is less than the input factor, then $\phi_{i,g}$ is not used in calculating $\overline{\lambda}$ and $\underline{\lambda}$.

The inner iterations are not accelerated because the gain would not warrant the additional coding complexity and storage requirements. The outer iterations are accelerated by means of three-term Chebyshev extrapolation.[2]

As we have seen, the outer iterations consist of the power iterates $\phi^{(n+1)}$ defined by

$$\phi^{(n+1)} = \frac{1}{k^{(n)}} T\phi^{(n)}, \tag{c.25}$$

where $T = E^{-1}F$. (This is true if no up-scattering is present; the matrix E^{-1} must be modified if up-scattering is present.) If we make the standard assumption that the eigenvalues of T are real, we can apply the usual three-term extrapolation procedure. Let the eigenvalues of T be denoted by $\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n$. Let $\sigma = \lambda_2/\lambda_1$, and suppose the ratios λ_j/λ_1 lie in the interval $[a,\sigma]$. The iterates then have the form

$$\phi^{(n)} = \alpha_1 \frac{\lambda_1^n}{\prod_{j=0}^{n-1} k^{(j)}} u_1 + \alpha_2 \frac{\lambda_2^n}{\prod_{j=0}^{n-1} k^{(j)}} u_2 + \ldots, \tag{c.26}$$

where u_j is the eigenvector associated with λ_j. For the sake of simplicity, all eigenvalues are assumed simple. If we assume that the estimates $k^{(n)}$ have converged, then Eq. (c.26) takes the form

$$\phi^{(n)} = \alpha_1' u_1 + \alpha_2' \sigma^n u_2 + \ldots , \tag{c.27}$$

where σ is the dominance ratio.

If we apply a polynominal Q_p in T/λ_1 to this iterate, we obtain

$$Q_p(T/\lambda_1)\phi^{(n)} = \alpha_1' Q_p(1)\, u_1 + \alpha_2' Q_p(\sigma) u_2 + \ldots \quad . \tag{c.28}$$

To preserve convergence, we would choose $Q_p(1)$ = constant, taken to be one, and to accelerate convergence, choose $Q_p(x)$ to be minimal in modulus over $[a,\sigma]$ among all polynominals of degree p. As shown in Ref. 2, Q_p then has the form

$$Q_p(x;\sigma;a) = \frac{C_p\left[\dfrac{2x - \sigma - a}{\sigma - a}\right]}{C_p\left[\dfrac{2 - \sigma - a}{\sigma - a}\right]}, \tag{c.29}$$

where $C_p(y)$ is the Chebyshev polynomial of degree p defined by

$$C_p(y) = \cosh p \cosh^{-1} y \text{ for } y > 1. \tag{c.30}$$

Since the polynomials C_p satisfy a three-term recursive relation, the polynomials Q_p satisfy the relation

$$Q_p(x) = Q_{p-1}(x) + \alpha_p(x - 1)\, Q_{p-1} + \beta_p(Q_{p-1} - Q_{p-2}), \tag{c.31}$$

where α_p and β_p are defined by

$$\left.\begin{aligned} &\alpha_1 = \frac{2}{2 - \sigma - a}, \\ &\beta_1 = 0\ ; \\ &\alpha_p = \frac{4}{\sigma - a}\ \frac{\cosh\,(p - 1)\gamma}{\cosh p\gamma}, \\ &1 + \beta_p = \left[1 - \tfrac{1}{2}(\sigma + a)\right]\alpha_p, \\ &\text{for } p \geq 2, \end{aligned}\right\} \tag{c.32}$$

and where $\cosh \gamma = \dfrac{2 - \sigma - a}{\sigma - a}$.

Thus, to apply an extrapolation of order p starting with the iterate $\phi^{(n)}$, we proceed as follows.

We wish to find $\hat{\phi}^{(n+p)} = Q_p\,(T/\lambda_1;\sigma;a)\,\phi^{(n)}$. Suppose we have found $\hat{\phi}^{(n+q-1)}$ for $1 \le q \le p$. Then we define

$$\left.\begin{aligned}
&\text{a)}\quad \phi^{(n+q)} = \frac{T}{k^{(n+q-1)}}\,\hat{\phi}^{(n+q-1)};\\
&\text{b)}\quad k^{(n+q)} = k^{(n+q-1)}\,\frac{\left(\phi^{(n+q)},\ \phi^{(n+q)}\right)}{\left(\phi^{(n+q)},\ \hat{\phi}^{(n+q-1)}\right)};\\
&\text{c)}\quad \overline{\lambda}^{(n+q)} = \max_{i,g}\left(\frac{\phi_{i,g}^{(n+q)}}{\hat{\phi}_{i,g}^{(n+q-1)}}\right);\ \underline{\lambda}^{(n+q)} = \min_{i,g}\left(\frac{\phi_{i,g}^{(n+q)}}{\hat{\phi}_{i,g}^{(n+q-1)}}\right);\\
&\text{d)}\quad \varepsilon_{\phi}^{(n+q)} = \frac{\sum_{g=1}^{G}\sum_{i=1}^{I}\left|\phi_{i,g}^{(n+q)} - \hat{\phi}_{i,g}^{(n+q-1)}\right|}{\sum_{g=1}^{G}\sum_{i=1}^{I}\phi_{i,g}^{(n+q)}}
\end{aligned}\right\}\qquad \text{(c.33)}$$

Then,

$$\hat{\phi}^{(n+q)} = \hat{\phi}^{(n+q-1)} + \alpha_q\left(\phi^{(n+q)} - \hat{\phi}^{(n+q-1)}\right) + \beta_q\left(\hat{\phi}^{(n+q-1)} - \hat{\phi}^{(n+q-2)}\right). \qquad \text{(c.34)}$$

With $\hat{\phi}^{(n)} \equiv \phi^{(n)}$, the above relations hold for $q \ge 1$. Since, in practice, $k^{(n)} \ne \lambda_1$, we are not performing the stationary process described by Eq. (c.28). In addition to the above quantities, we form

$$E_{n,q} = \frac{\varepsilon_{\phi}^{(n+q+1)}}{\varepsilon_{\phi}^{(n+1)}} \quad \text{for } q \ge 2. \qquad \text{(c.35)}$$

For a successful application of the extrapolation procedure, two conditions must be met:

i) $k^{(n)} = \lambda_1$,

and ii) an accurate estimate of σ must be known.

Neither condition is met until the problem has converged; however, the sequence $k^{(n)}$ converges at a faster rate than do the fluxes $\phi^{(n)}$. Thus, condition (i) is met quite early in the iteration procedure.

An accurate estimate $\sigma = \lambda_2/\lambda_1$ is also impossible until the later stages of convergence. Rather than continually trying to estimate σ by some direct means, we have adopted the method, used in the PDQ programs, of estimating σ from the results of previous extrapolations; i.e., we use the updating method.[4] First an initial estimate of σ is computed under the assumption that $k^{(n)} \cong \lambda_1$. From Eq. (c.26),

$$\frac{\varepsilon_\phi^{(n+1)}}{\varepsilon_\phi^{(n)}} \cong \sigma \, . \tag{c.36}$$

Equation (c.36) provides an initial estimate; later in the iteration procedure, it provides estimates when circumstances dictate its use.

Now suppose that we have an estimate $\bar{\sigma}$ for σ and that we have performed an extrapolation cycle of order p on the flux $\phi^{(n)}$. In addition, suppose that this cycle was followed by one unextrapolated iteration so that we have the flux $\phi^{(n+p+1)}$ at our disposal. From Eq. (c.28), we see that the quantity $E_{n,p}$ defined by Eq. (c.35) has the following approximate value:

$$E_{n,p} = \frac{\varepsilon_\phi^{(n+p+1)}}{\varepsilon_\phi^{(n+1)}} \cong Q_p(\sigma;\bar{\sigma};a) \, . \tag{c.37}$$

Now, if our estimate $\bar{\sigma}$ is equal to the theoretical value σ, then our acceleration is optimal for this method and the error reduction ratio is given by

$$Q_p(\bar{\sigma}:\bar{\sigma}:a) = \frac{1}{C_p\left(\dfrac{2 - \bar{\sigma} - a}{\bar{\sigma} - a}\right)} \, , \tag{c.38}$$

where $C_p(x)$ is the Chebyshev polynomial of degree p.

Thus the first thing we do is to compare the actual error-reduction ratio, $E_{n,p}$, with the optimal one; i.e., we make the following test:

$$E_{n,p} \leq \frac{1}{C_p\left(\dfrac{2 - \bar{\sigma} - a}{\bar{\sigma} - a}\right)} . \tag{c.39}$$

If this test is satisfied, we keep $\bar{\sigma}$. If this test is not satisfied and $E_{n,p} < 1$, then σ is too low and we re-estimate $\bar{\sigma}$. This conclusion rests heavily on the assumption that $k^{(n)} \cong \lambda_1$ and that the error reduction ratio $E_{n,p}$ is an accurate estimate of the theoretical error reduction factor involved when $\bar{\sigma}$ is an incorrect estimate for the true σ. Thus, if these assumptions are satisfied, we obtain a new estimate for σ by solving the following equation for σ':

$$E_{n,p} = Q_p(\sigma';\bar{\sigma};a). \tag{c.40}$$

The new estimate σ' is then given by

$$\sigma' = \frac{1}{2}\left\{(\bar{\sigma} + a) + (\bar{\sigma} - a) \cosh \frac{1}{p}\,\Omega\right\} , \tag{c.41}$$

where $\Omega = \cosh^{-1} [E_{n,p} \cosh p\gamma]$, and γ is defined by

$$\gamma = \cosh^{-1}\left(\frac{2 - \bar{\sigma} - a}{\bar{\sigma} - a}\right) .$$

It has been shown in Ref. 4 that $1 < \sigma' < \bar{\sigma}$.

In order to discuss how the extrapolation is performed, we must define the following integer input parameters:

(i) N_{start} - the number of unextrapolated iterations to be performed at the start before testing whether or not to extrapolate;

(ii) N_{test} - the number of extrapolated iterations to be performed in an extrapolation sequence before deciding whether the extrapolations are effective;

(iii) N_{ext} - the maximum number of extrapolated iterations permitted in an extrapolation sequence;

(iv) N_{unext} - the number of unextrapolated iterations to be performed following one extrapolation sequence before deciding whether to begin another.

We can now describe the procedure used in applying Chebyshev extrapolation to the outer iterations.

At the start of the iterations, N_{start} unextrapolated iterations are performed. Then the following two tests are made:

$$\text{(i)}\ \frac{\bar{\lambda}^{(n+1)} - \lambda^{(n+1)}}{\bar{\lambda}^{(n)} - \underline{\lambda}^{(n)}} > .5 \quad ; \tag{c.42}$$

and

$$\text{(ii)}\ \bar{\sigma}^{(n+1)} > .5 \ .$$

Each of the quantities on the left is a measure of the rate of convergence. If neither of these tests is satisfied, another unextrapolated iteration is performed and the test is repeated.

If, on the other hand, either test is satisfied, we test $\bar{\sigma}^{(n+1)} > 1$. If this test is satisfied the iterations are diverging, and extrapolation is not attempted. If this test is satisfied on twelve successive iterations, execution is terminated. However, if this test is not satisfied, an extrapolation sequence is begun with $\bar{\sigma}^{(n+1)}$ as the estimate for σ.

Suppose the extrapolation sequence has begun with the nth iteration. N_{test} extrapolated iterations will be performed, and the effectiveness of the iterations will then be tested. This test will occur after every effective extrapolated iteration until N_{ext} extrapolated iterations have been performed. At this point, at least N_{unext} unextrapolated iterations will be performed before deciding whether to embark on another extrapolation sequence.

We can now discuss the effectiveness test more fully. Again, suppose the iteration count is n+p, such that p extrapolated iterations have been performed, where $p \geq N_{test}$. Moreover, let $\bar{\sigma}$ denote the estimate for σ. The following test is then made:

$$E_{n,p-1} \leq \bar{\sigma}^{p-1} \ . \tag{c.43}$$

Recall that $E_{n,p-1}$ is an indicator of the error reduction in this cycle of length (p-1) which started at the nth iteration. Hence, the above test is an attempt to ascertain whether this cycle was at least as effective as a cycle of unextrapolated iterations having the same length. If this test is failed, the extrapolation sequence is ended and N_{unext} unextrapolated iterations are performed. A new estimate for σ is obtained using

$\overline{\sigma}$ as defined in Eq. (c.34) over the set of unextrapolated iterations. A decision is then made as to whether to try another extrapolation cycle.

However, if the test (c.43) is satisfied, a further test is then made on the convergence of the iterates $k^{(n)}$. If these have not converged, we will be satisfied with the current estimate for σ, provided that the sequence of length p was at least as effective as p+1 unextrapolated iterations would have been. If it was not as effective, the extrapolation sequence is ended and N_{unext} unextrapolated iterations are performed. A new estimate of σ is obtained using $\overline{\sigma}$ over this set.

If the iterates $k^{(n)}$ are converged, we test the error reduction factor $E_{n,p-1}$ against the theoretical error reduction factor as in (c.39). If this test is satisfied, the estimate for σ is retained and extrapolation is continued, unless, of course, N_{ext} extrapolated iterations have been performed in the sequence, in which case N_{unext} unextrapolated iterations will be performed and a new cycle will be started using the same estimate for σ. If the error reduction factor does not satisfy the test (c.39), a new estimate σ' for the dominance ratio σ will be sought according to Eq. (c.41). This new estimate will be used in the next extrapolation sequence, which will be begun after N_{unext} unextrapolated iterations have been performed.

It should be noted that estimates for σ, however they arise, are never used if they exceed 0.99995. If they do exceed this number, then it is this number that is used for σ in the extrapolation sequence.

d. <u>The Solution of the Adjoint k_{eff} Calculation</u>

In the adjoint k_{eff} calculation, we wish to find the solution ϕ^* and the corresponding k_{eff} for the equation adjoint to Eq. (c.9),

$$E^*\phi^* = \frac{1}{k} F^* \phi^*, \tag{d.1}$$

where E^* and F^* denote the adjoint or transposed matrices of E and F, respectively. They are defined as follows.

$$E^* = \begin{bmatrix} A_1^* & -B^{12} & \cdots & -B^{1G} \\ -B^{21} & A_2^* & \cdots & -B^{2G} \\ \vdots & \vdots & \ddots & \vdots \\ -B^{G1} & -B^{G2} & \cdots & A_G^* \end{bmatrix} \tag{d.2}$$

If only black compositions were present, $a_{i,g} = c_{i+1,g}$, so that the matrices A_g would be symmetric in that case.

$$F^* = \begin{bmatrix} F^{11} & F^{12} & \cdots & F^{1G} \\ F^{21} & F^{22} & F^{23} \cdots & F^{2G} \\ \vdots & \vdots & \ddots & \vdots \\ F^{G1} & F^{G2} & \cdots & F^{GG} \end{bmatrix} \tag{d.3}$$

Since the matrix E^* is heavily upper triangular and the iteration algorithms are designed for a lower-triangular matrix, we reorder the adjoint flux during the equation by defining

$$\psi_{i,g} = \phi^*_{i,G-g+1} \tag{d.4}$$

When the indices of the matrix elements are reordered in this manner, the calculation can proceed exactly as in the real k_{eff} calculation. At the end of the calculation, the flux vector is reordered to obtain the adjoint flux in its standard ordering.

e. <u>The Solution of the Source Calculation</u>

Source calculations arise from the presence of an inhomogeneous source density distribution, from the presence of inhomogeneous boundary conditions, or from both. A source calculation can be performed with either the real or the adjoint matrices arising from the finite approximation to the multi-group diffusion equations; for the adjoint case, the matrices and the resulting fluxes are reordered by group as described in section d. above.

Fixed external source can be input either by mesh interval and group or by mesh interval alone, to be distributed over groups in each interval using a fission density weighting. For the case of a real problem with a χ-vector, the weighting is accomplished as $S_{i,g} = \chi_c^g S_i$, where c is the fissionable composition assigned to mesh interval i. For a real problem with a χ-matrix, the distributed source is $S_{i,g} = \left(\frac{1}{G}\sum_{g'=1}^{G} \chi_c^{g'g}\right) S_i$. For an adjoint problem,

$$S_{i,g} = \left[\frac{(\nu\cdot\sigma_f)_c^g}{\sum_{g'=1}^{G} (\nu\cdot\sigma_f)_c^{g'}}\right] S_i \quad .$$

For the fixed external source, we wish to find the corresponding flux; thus, in the notation of Eqs. (c.7)-(c.9), we must solve the equation

$$(E - \mu F)\phi = S \quad , \tag{e.1}$$

where μ is a specified constant which ensures that the reactor in the absence of any source is subcritical. Since matrix F can be redefined, it is sufficient to consider the case in which $\mu = 1$.

Since, by our assumption, the largest eigenvalue of $E^{-1}F$ is less than one, $(E-F)^{-1} = (1-E^{-1}F)^{-1} E^{-1}$ exists. Moreover, it is non-negative, since both E^{-1} and $E^{-1}F$ are non-negative. In addition, the off-diagonal elements are nonpositive; hence we can employ any iterative method derived from a regular splitting[2] of E-F. Let A = E-F; then a decomposition A = M-N is called a regular splitting of A if matrices M and N satisfy the condition $N \geq 0$, M^{-1} exists, and $M^{-1} \geq 0$. The iterative process for solving a system Ax = f generated by a regular splitting A = M-N is defined by $Mx^{(n+1)} = Nx^{(n)} + f$. Note that the splitting A = E-F is a regular splitting of A and from the above discussion $A^{-1} \geq 0$.

The following result is valid for regular splittings (Ref. 2, Theorem 3.13, pg 89). Let $\rho(T)$ denote the spectral radius of any matrix T. Let A = M-N be a regular splitting of A and suppose $A^{-1} \geq 0$; then

$$\rho(M^{-1}N) = \frac{\rho(A^{-1}N)}{1 + \rho(A^{-1}N)} \quad .$$

From this result we see that if $A = M_i - N_i = M_2 - N_2$ are two regular splittings with $N_1 \geq N_2$, then $\rho(M_1^{-1}N_1) \geq \rho(M_2^{-1}N_2)$. Hence the asymptotic rate of convergence of the iterative process generated by the splitting M_2-N_2 is not less than that generated by the splitting M_1-N_1. To this end, the following splitting is used in the source calculation: $E-F = (E-F_L) - F_U$.

$$E-F_L = \begin{bmatrix} A_1 & & & \bigcirc \\ -(B^{12}+F^{12})A_2 & \cdot & & \\ \cdot & & \cdot & \\ \cdot & & & \cdot \\ \cdot & & & \\ -(B^{1G}+F^{1G}) & \cdots & \cdots & A_G \end{bmatrix}$$

The matrix F_U is the remaining upper triangular portion of the matrix F. Note that F_U is not strictly upper triangular. Hence the basic iteration scheme is defined by

$$A_g\phi_g^{(n+1)} = \sum_{g'<g} (B^{g'g} + F^{g'g})\phi_{g'}^{(n+1)} + \sum_{g'\geq g} (B^{g'g} + F^{g'g})\phi_{g'}^{(n)} + S_g. \tag{e.3}$$

The tridiagonal matrix A_g is inverted by the standard Choleski method used in the k-effective calculation.

Successive block over-relaxation is not used to accelerate the source calculations for the following reason. The block Jacobi matrix $D^{-1}(L + U)$ of this splitting is primitive[2] for $G > 2$, since $F^{g'g} > 0$ for $g' \neq g$. Thus the optimal over-relaxation factor is not given by the relation

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - r^2}}, \tag{e.4}$$

which is valid for consistently ordered two cyclic Jacobi matrices,[2] where r is the spectral radius of the Jacobi matrix. Without some such relationship or procedure for estimating the optimal factor, the method of successive over-relaxation loses a great deal of its effectiveness.

For this reason, an alternative acceleration procedure is provided for source calculations. Recall that in terms of the block Gauss-Seidel splitting $E - F = (D - L) - U$, the basic iterative procedure can be written in the form

$$\phi^{(n+1)} = R\phi^{(n)} + f, \tag{e.5}$$

where $R = (D - L)^{-1} U$ and $f = (D - L)^{-1}S$. If it is assumed that the matrix R has only real eigenvalues, then Chebyshev acceleration can be applied to the above procedure.[2] Suppose that the eigenvalues lie in the interval $[a,\lambda]$, where $|a| \leq \lambda < 1$. Suppose that beginning with $\phi^{(n)}$ we wish to accelerate the sequence of iterates. If $\hat{\phi}$ denotes the extrapolated flux, the procedure can be represented in the form

$$\hat{\phi}^{(n+p)} = \hat{\phi}^{(n+p-1)} + \alpha_p \left(\phi^{(n+p)} - \hat{\phi}^{(n+p-1)}\right) + \beta_p\left(\hat{\phi}^{(n+p-1)} - \hat{\phi}^{(n+p-2)}\right). \tag{e.6}$$

The coefficients α_p and β_p are defined just as they were in Eq. (c.32), except that now the estimate λ for the largest eigenvalue in modulus of R replaces the estimate for the dominance ratio σ of the matrix T. Note that the unextrapolated flux $\phi^{(n+p)}$ is defined by

$$\phi^{(n+p)} = R\hat{\phi}^{(n+p-1)} + f \quad \text{for } p \geq 1, \tag{e.7}$$

with $\hat{\phi}^{(n)} \equiv \phi^{(n)}$. Again, just as in the power iteration case, the estimation of acceleration parameters is the most important aspect of the procedure. In this case the relevant parameter is the estimate λ of the largest eigenvalue of R. The other parameter, a, is an input quantity. We use the same technique that is used in the power iteration case for estimating and updating the estimate of the largest eigenvalue of R.

f. Output Edits

The edits which may be provided by this module are divided into three categories: input, iteration history, and output.

(i) Input

This section first gives the general specifications of the problem from data sets A.DIF1D and SP.CICN, as follows:

the calculation type (K-EFF or SOURCE) and problem type (REAL, ADJOINT, or BOTH);

the $\mu = 1/k_{fixed}$ factor for a source calculation;

the kinetics option type* (no kinetics option, delayed data for a source calculation, or delayed data for an alpha calculation);

the maximum number of outer iterations, NMAX;

the extrapolation type (no extrapolation, Chebyshev extrapolation, or successive over-relaxation);

the number of unextrapolated iterations to be done before attempting to extrapolate, NSTART;

the minimum number of iterations to be done in an extrapolation sequence, NTEST;

the maximum number of iterations to be done in an extrapolation sequence, NEXT, for Chebyshev extrapolation (not applicable to SOR);

the maximum number of unextrapolated iterations to be done after one extrapolation sequence before beginning another, NUNEXT;

the interval shift factor for Chebyshev extrapolations, a;

the convergence criteria ε_k (for a k-effective calculation), ε_ϕ, and ε_λ;

the ratio of flux to root mean square flux below which fluxes will be neglected in computing $\bar{\lambda}$ and $\underline{\lambda}$, FLXMON;

the number of iterations after which the flux data set will be written for the restart option, NRST, and the increment to be used in this number thereafter, NRSTI;

the maximum number of upscattering iterations, LMAX, and the upscattering iteration convergence criterion ε_S, if upscattering iterations are defined;

the power normalization factor for burnup (watts), P_0, if one is defined;

---

*The two kinetics options are not available, although some provision has been made for them in the module, because the input data needed has not been defined. Hence, these options are not discussed here.

and the amount of storage allocation for POINTR arrays, in number of R*8 words in core and number in bulk storage. (After the first time a problem is run, these values can be determined from the printed output; their sum should be the maximum of the amounts of storage needed for the input, calculation, and output sections).

Following this general information, an edit of the geometry data from data set GEOM is given. The geometry type (CYLINDER, SPHERE, RECTANGULAR SLAB, or CYLINDRICAL SLAB) is printed, along with a table including the locations of mesh lines, the length and volume of each mesh interval, and the names of the region and composition corresponding to each interval. A second table, by regions, gives the name of each region; the name of the composition assigned to it; the buckling, actual half height, and extrapolation distance in each finite direction defined; and the physical region volume.

The next edit displays the boundary and interface condition types and constants from data set BC by mesh line location and number. A negative mesh line number indicates the condition is defined on the left of that line; a positive number indicates that it is defined on the right.

The fourth section consists of a display of the group-dependent macroscopic cross-section data for each composition defined. The fission spectrum χ, if present as a matrix, appears in a separate table; the scattering cross section from g' into g appears in the final table.

If external source data has been defined, it is then displayed by mesh interval and group.

Following the input section, the amount of POINTR storage used in core and in bulk, in R*8 words, is printed.

(ii) Iteration history

First the amount of POINTR storage required for the calculation section, in core and in bulk, is printed.

Two lines of output are then printed for each outer iteration: these differ slightly depending on the calculation type. For a k_{eff} calculation, the following measures of convergence are printed on the first line:

ITER NO, the outer iteration number, n;

KEFF, $k_{eff}^{(n)}$;

DIF KEFF, $k_{eff}^{(n)} - k_{eff}^{(n-1)}$;

LAM UPPER, $\overline{\lambda}^{(n)} = \max_{i,g}\left(\dfrac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$, where $\phi_{i,g}^{(n-1)} >$ NGL FLUX;

LAM LOWER, $\underline{\lambda}^{(n)} = \min_{i,g}\left(\dfrac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$, where $\phi_{i,g}^{(n-1)} >$ NGL FLUX;

DIF LAM, $\overline{\lambda}^{(n)} - \underline{\lambda}^{(n)}$;

DIF FLUX, $\dfrac{\sum_{g=1}^{G}\sum_{i=1}^{I}\left|\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)}\right|}{\sum_{g=1}^{G}\sum_{i=1}^{I}\left|\phi_{i,g}^{(n)}\right|}$;

and NGL FLUX, $\delta\sqrt{\dfrac{\sum_{g=1}^{G}\sum_{i=1}^{I}(\phi_{i,g}^{(n)})^2}{I\cdot G}}$, where δ is an A.DIF1D input quantity, FLXMON.

On the second line, the following acceleration parameters are given:

P, the Chebyshev extrapolation iteration counter (when $P < 0$, it is a count of the number of unextrapolated iterations to be done before attempting an extrapolation sequence; when $P > 0$, it is the degree P of the Chebyshev polynomial);

SIGMA, the estimate for the dominance ratio, σ;

ENPM1, the theoretical error-reduction factor $E_{p-1}^{(n)}$;

SIGMA BAR, the estimate of the convergence rate, $\overline{\sigma}$;

LAM RAT, $\dfrac{\overline{\lambda}^{(n)} - \underline{\lambda}^{(n)}}{\overline{\lambda}^{(n-1)} - \underline{\lambda}^{(n-1)}}$;

ALPHA, the Chebyshev acceleration parameter α;

and BETA, the Chebyshev acceleration parameter β.

For a source calculation, the first line contains the following:

ITER, the outer iteration number, n;

GROUP, the number of the group g in which DIF FLUX occurred;

$$\text{LAM UPPER, } \overline{\lambda}^{(n)} = \max_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right), \text{ where } \phi_{i,g}^{(n-1)} > \text{NGL FLUX;}$$

$$\text{LAM LOWER, } \underline{\lambda}^{(n)} = \min_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right), \text{ where } \phi_{i,g}^{(n-1)} > \text{NGL FLUX;}$$

$$\text{DIF LAM, } \overline{\lambda}^{(n)} - \underline{\lambda}^{(n)} \text{ ;}$$

$$\text{DIF FLUX, } \max_{g} \left[\frac{G \cdot \sum_{i=1}^{I} \left(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)} \right)}{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n-1)}} \right] \text{ ;}$$

and NGL FLUX, $\delta \sqrt{\dfrac{\sum_{i=1}^{I} \sum_{g=1}^{G} \left(\phi_{i,g}^{(n)} \right)^2}{I \cdot G}}$, where δ is an A.DIF1D input quantity, FLXMON.

On the second line, the same acceleration parameters as for a k_{eff} calculation are given for a problem using no acceleration or Chebyshev accleration. For a problem using successive over-relaxation to accelerate, the parameters given on the second line are

P, the SOR extrapolation iteration counter;

ENPM1, the theoretical error-reduction factor, $E_{p-1}^{(n)}$;

SIGMA BAR, the estimate of the convergence rate, $\overline{\sigma}$;

$$\text{LAM RATIO, } \frac{\overline{\lambda}_{i,g}^{(n)} - \underline{\lambda}_{i,g}^{(n)}}{\overline{\lambda}^{(n-1)} - \underline{\lambda}^{(n-1)}} \text{ ;}$$

and EXTRAP PARAM, the SOR acceleration parameter, ω.

(iii) Output

First the amount of POINTR storage required for the output section, in R*8 words in core and in bulk, is printed.

This module provides four basic categories of output inventories, all based on macroscopic cross sections. These major categories are flux integrals, fission integrals, power calculations, and balances.

All inventories are based on weighted integrals of the flux as a function of three spatial variables, depending on the geometry type, and one energy variable, with the integrals extending over the physical dimensions of the reactor.

In conjunction with the derivation of the finite difference coefficients, the following shapes were assumed for the flux in each of the four geometries:

a) sphere $\phi(\rho,\theta,\phi) = \phi(\rho)$;

b) cylinder $\phi(\rho,z,\theta) = \phi(\rho) \cos \frac{\pi z}{2H_r}$ for a finite reactor

or $\phi(\rho,z,\theta) = \phi(\rho)$ for an infinite reactor,

c) rectangular slab $\phi(x,y,z) = \phi(x) \cos \frac{\pi y}{2H_{r,1}} \cos \frac{\pi z}{2H_{r,2}}$ for a finite rectangular parallelepiped

or $\phi(x,y,z) = \phi(x) \cos \frac{\pi y}{2H_{r,1}}$ for a finite column

or $\phi(x,y,z) = \phi(x)$ for an infinite reactor;

d) cylindrical slab $\phi(\rho,z,\theta) = \phi(z)\, J_0\left(\frac{\alpha\rho}{R_r}\right)$ for a finite reactor

or $\phi(\rho,z,\theta) = \phi(z)$ for an infinite reactor.

Consider any macroscopic cross section $\sigma^g(r)$, where r denotes a particular region in the reactor. Using cylindrical geometry as an example, a typical inventory $I_{r,g}$ might involve the numerical approximation to the following integral:

$$\int_{\rho\varepsilon r} \int_0^{2\pi} \int_{-(H_L)_r}^{(H_L)_r} \sigma^g(r)\ \phi_g(\rho)\ \cos\frac{\pi z}{2H_r}\ dz\ d\theta\rho d\rho$$

$$= \int_{\rho\varepsilon r} \sigma^g(r)\ \phi_g(\rho) \left[\int_{-(H_L)_r}^{(H_L)_r} \cos\frac{\pi z}{2H_r}\ dz \right] 2\pi\rho\ d\rho$$

$$= \int_{\rho\varepsilon r} \sigma^g(r)\ \phi_g(\rho)\ W_r\ 2\pi\rho\ d\rho$$

$$\cong W_r\ \sigma^g(r) \sum_{i\varepsilon r} \phi_g(\rho_i)\ V_i \quad ,$$

where W_r is the integration weighting factor and V_i is the volume element corresponding to cylindrical geometry. Also for cylindrical geometry, the physical volume V_r of region r is defined as the approximation of the region integral, as follows:

$$V_r = \int_{\rho\varepsilon r} \int_0^{2\pi} \int_{-(H_L)_r}^{(H_L)_r} d\theta\ dz\ \rho\ d\rho$$

$$= \int_{\rho\varepsilon r} 2(H_L)_r\ (2\pi\rho d\rho)$$

$$\cong 2(H_L)_r \sum_{i\varepsilon r} V_i \quad .$$

For each geometry type, then, the following table gives the values used for the volume elements V_i, the integration weighting factors W_r, and the physical volumes V_r.

Geometry	V_i	W_r	V_r
Sphere	$\frac{4\pi}{3}(R_{i+1}^3 - R_i^3)$	1	$\sum_{i\varepsilon r} V_i$
Cylinder			
finite	$\pi(R_{i+1}^2 - R_i^2)$	$\frac{4\,H_r}{\pi}\sin\frac{\pi(H_L)_r}{2\,H_r}$	$2(H_L)_r \sum_{i\varepsilon r} V_i$
infinite	$\pi(R_{i+1}^2 - R_i^2)$	1	$\sum_{i\varepsilon r} V_i$
Rectangular Slab			
rectangular parallelepiped	$\Delta X_i = X_{i+1} - X_i$	$\left(\frac{4\,H_{r,1}}{\pi}\sin\frac{\pi(H_L)_{r,1}}{2\,H_{r,1}}\right)\left(\frac{4\,H_{r,2}}{\pi}\sin\frac{\pi(H_L)_{r,2}}{2\,H_{r,2}}\right)$	$(2(H_L)_{r,1})(2(H_L)_{r,2})\sum_{i\varepsilon r} V_i$
rectangular column	$\Delta X_i = X_{i+1} - X_i$	$\frac{4\,H_{r,1}}{\pi}\sin\frac{\pi(H_L)_{r,1}}{2\,H_{r,1}}$	$2\,(H_L)_{r,1}\sum_{i\varepsilon r} V_i$
infinite slab	$\Delta X_i = X_{i+1} - X_i$	1	$\sum_{i\varepsilon r} V_i$
Cylindrical Slab			
finite	$\pi\Delta Z_i = \pi(Z_{i+1}-Z_i)$	$\frac{(R_L)_r\,R_r}{\alpha}J_1\left(\frac{\alpha(R_L)_r}{R_r}\right)$	$(R_L)^2\sum_{i\varepsilon r} V_i$
infinite	$\pi\Delta Z_i = \pi(Z_{i+1}-Z_i)$	1	$\sum_{i\varepsilon r} V_i$

Here, $(H_L)_r$, $(H_L)_{r,1}$, and $(H_L)_{r,2}$ denote the physical half-heights of region r in cylindrical and rectangular slab geometries, respectively, and $(R_L)_r$ denotes the physical radius of region r in cylindrical slab geometry. H_r, $H_{r,1}$, and $H_{r,2}$ denote extrapolated half-heights, and R_r denotes the extrapolated radius, correspondingly.

We can now display the calculations made in each of the four inventory categories. Flux integrals computed are of the following types:

a) real or adjoint regional flux integrals by group, $\Phi_{r,g} = W_r \sum_{i \varepsilon r} \phi_{i,g} V_i$;

b) real or adjoint regional flux averages by group, $\overline{\Phi}_{r,g} = \frac{\Phi_{r,g}}{V_r}$;

c) real power-normalized regional flux averages by group, if input power factor P_0 in watts is present, $\overline{\Phi}^P_{r,g} = \overline{\Phi}_{r,g} \cdot 3.1 \times 10^{10} \cdot P_0$

For a k-effective calculation, all of these quantities as well as the scalar flux $\phi_{i,g}$ by interval and group are normalized to one fission in the reactor. The normalization factor N by which all are divided is

$$N = \sum_{g=1}^{G} \sum_{r=1}^{R} (\sigma_f)^g_{c(r)} \Phi_{r,g} ,$$

where c(r) is the composition assigned to region r. For a source calculation, no normalization of this type is performed.

Two fission integrals by interval and group are computed. They are

a) fission density:

i) for real χ-matrix $FD_{i,g} = W_r \sum_{g'=1}^{G} \chi^{g'g}_{c(r)} (\nu \cdot \sigma_f)^{g'}_{c(r)} \phi_{i,g'}$;

ii) for real χ-vector $FD_{i,g} = W_r \chi^g_{c(r)} \sum_{g'=1}^{G} (\nu \cdot \sigma_f)^{g'}_{c(r)} \phi_{i,g'}$;

iii) for adjoint χ-matrix $FD_{i,g} = W_r (\nu \cdot \sigma_f)^g_{c(r)} \sum_{g'=1}^{G} \chi^{g'g}_{c(r)} \phi_{i,g'}$;

iv) for adjoint χ-vector $FD_{i,g} = W_r (\nu \cdot \sigma_f)^g_{c(r)} \sum_{g'=1}^{G} \chi^{g'}_{c(r)} \phi_{i,g'}$.

(For a source calculation, $FD_{i,g}$ is multiplied by the scaling factor $\mu = 1/k_{fixed}$.)

and

b) fission source: $FS_{i,g} = FD_{i,g} \cdot V_i$.

On option, the following power calculations are performed:

a) average power by intervals $\overline{P}_i = \sum_{g=1}^{G} W_{r(i)} \phi_{i,g} \cdot (\sigma_f)^g_{c(i)}$;

b) maximum average power over intervals $\max_i [\overline{P}_i]$;

c) minimum average power over intervals $\min_i [\overline{P}_i]$;

d) power by regions $P_r = \sum_{i \varepsilon r} \overline{P}_i V_i$;

e) maximum power over regions $\max_r [P_r]$;

f) minimum power over regions $\min_r [P_r]$;

g) average power by regions $\overline{P}_r = \frac{P_r}{V_r}$;

h) total power $P_T = \sum_{r=1}^{R} P_r$.

The balance calculations by region and group consist of a leakage term, including $D \cdot B^2$; a removal term, including $\frac{\alpha}{v}$, where α is the inverse reactor time period and v is the neutron speed by group; a scattering source term; a fission source term; and, for source calculations, an external source term, as well as the total balance term. The current across region interfaces and the average fission source by region and group are also computed. In addition, each of the balance terms is displayed summed over regions by group, over groups by region, and over the reactor.

The leakage term is defined as

$$L_{r,g} = W_r \sum_{i \varepsilon r} L_{i,g} \quad ,$$

where the total leakage in a given mesh cell, $L_{i,g}$, is defined as

$$L_{i,g} = L_{i,gR} + L_{i,gL} + D^g_{c(i)} (B^2_{r(i),1} + B^2_{r(i),2}) \phi_{i,g} V_i \quad ,$$

if buckling is defined, and simply

$$L_{i,g} = L_{i,gL} + L_{i,gR}$$

if it is not.

The leakages on the right and left of the cell, $L_{i,gR}$ and $L_{i,gL}$, are defined as follows for the condition of continuity of flux and current:

$$L_{i,gR} = - \frac{D^g_{c(i)} \; D^g_{c(i+1)} \; (\phi_{i+1,g} - \phi_{i,g}) \; A_{iR}}{\frac{1}{2} \Delta X_i \; D^g_{c(i+1)} + \frac{1}{2} \Delta X_{i+1} \; D^g_{c(i)}}$$

and $$L_{i,gL} = \frac{D^g_{c(i)} \; D^g_{c(i-1)} \; (\phi_{i,g} - \phi_{i-1,g}) \; A_{iL}}{\frac{1}{2} \Delta X_i \; D^g_{c(i-1)} + \frac{1}{2} \Delta X_{i-1} \; D^g_{c(i)}} \; .$$

For a boundary condition of the form $A\phi' + B\phi = C$, these quantities become

$$L_{i,gR} = \frac{D^g_{c(i)} (C - B\phi_{i,g}) \; A_{iR}}{A + \frac{1}{2} \Delta X_i \; B} \; .$$

and $$L_{i,gL} = \frac{D^g_{c(i)} \; (B\phi_{i,g} - C) \; A_{iL}}{A + \frac{1}{2} \Delta X_i \; B} \; .$$

Here A_{i_R} and A_{i_L} are the area terms to the left and right of mesh line i, and are geometry-dependent, as defined in section b.

The current at a region boundary r (where there are R + 1 region boundaries, including the exterior reactor boundaries), is calculated similarly. If region boundary r falls on mesh line i, the current given is that on the right-hand side of mesh line i:

$$J_{i,g_R} = - \frac{D^g_{c(i)} \; D^g_{c(i+1)} \; (\phi_{i+1,g} - \phi_{i,g})}{\frac{1}{2} \Delta X_i \; D^g_{c(i+1)} + \frac{1}{2} \Delta X_{i+1} \; D^g_{c(i)}}$$

for the condition of continuity of flux and current, or

$$J_{i,g_R} = - \frac{D^g_{c(i)} \; (C - B\phi_{i,g})}{A + \frac{1}{2} \Delta X_i \; B}$$

for a boundary condition of the form $A\phi' + B\phi = C$.

The only exception is the left reactor boundary, on the mesh line $i = 1$. In this case, the negative of the current on the left-hand side of the mesh line is used:

$$- J_{1,g_L} = - \frac{D^g_{c(1)} \; (B\phi_{1,g} - C)}{A + \frac{1}{2} \Delta X_1 \; B} \quad .$$

The removal contribution to the balance is simply

$$R_{r,g} = (\sigma_R)^g_{c(r)} + \frac{\alpha}{v_g} \quad \Phi_{r,g} \quad ,$$

where α is the inverse reactor time period, if present, and v_g is the neutron speed corresponding to group g.

The scattering source contribution to the balance is defined as

$$SS_{r,g} = \sum_{g'=1}^{G} \sigma^{g'g}_{c(r)} \; \Phi_{r,g'} \quad .$$

The fission source contribution by region and group is based on the fission source term by interval and group computed previously;

$$FS_{r,g} = \sum_{i \varepsilon r} FS_{i,g} \quad .$$

From this, the average fission source by region and group is obtained as

$$\overline{FS}_{r,g} = \frac{FS_{r,g}}{V_r} \quad .$$

The average fission source is displayed only by region and group; it is not summed over regions, over groups, or over the reactor.

The external source contribution, present only for a source calculation, is defined as

$$ES_{r,g} = W_r \sum_{i\varepsilon r} (S_{ext})_{i,g} V_i \quad ,$$

where S_{ext} is the external source by interval and group.

The balance term $B_{r,g}$ is then calculated simply as

$$B_{r,g} = - L_{r,g} - R_{r,g} + SS_{r,g} + \frac{1}{k_{eff}} FS_{r,g}$$

for a k_{eff} calculation, or

$$B_{r,g} = - L_{r,g} - R_{r,g} + SS_{r,g} + FS_{r,g} + ES_{r,g}$$

for a source calculation.

As mentioned, balance totals by region, by group, and over the reactor are then displayed.

g. References

1. A. Hassitt, *CRAM - A One and Two Dimensional Multigroup Diffusion Theory Reactor Code*, TRG Report 229(r) (March 8, 1962).

2. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Inc., New Jersey (1962).

3. A. Ralston and H. S. Wilf, Eds., *Mathematical Methods for Digital Computers*, John Wiley & Sons, Inc. (1960).

4. G. G. Bilodeau et al., *PDQ - An IBM-704 Code to Solve the Two-Dimensional Few-group Neutron Diffusion Equations*, WAPD-TM-70 (1957).

5. K. Ott, *Quasistatic Treatment of Spatial Phenomena in Reactor Dynamics*, Nucl. Sci. Eng., Vol. 26, No. 4, p. 563 (1966) Technical Note.

6. B. J. Toppel, ed., *The Argonne Reactor Computation (ARC) System*, ANL-7332 (November, 1967).

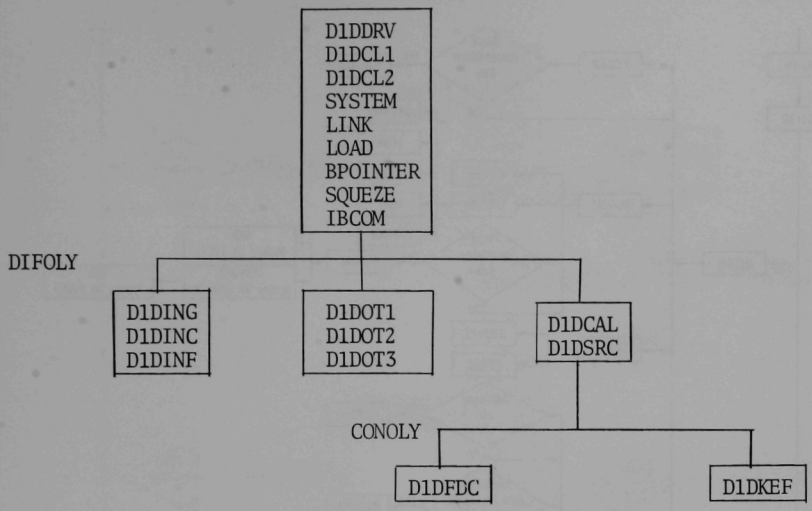
3. DIF1D (NUC002) Description of Module Flow

This section defines the overlay structure of this module and the flow of logic through the subprograms which form the module.

Section a. consists of an overlay structure diagram, showing the ARC System segments and DIF1D module segments required by this module and their positions in the overlay heirarchy. This is followed by a list of the subprograms located in each DIF1D module segment, in alphabetical order.

Section b. illustrates the flow of logic through the subprograms of this module, and section c. describes this flow in terms of the module's functions. Calls to PAGHED, the subroutine which prints the page heading, and ERRMSG, the subroutine which prints error messages, are not indicated, nor are calls to subroutines or entry points which are part of the ARC System utility routines, such as POINTR and SNIFF.

a. DIF1D (NUC002) Overlay Structure Diagram



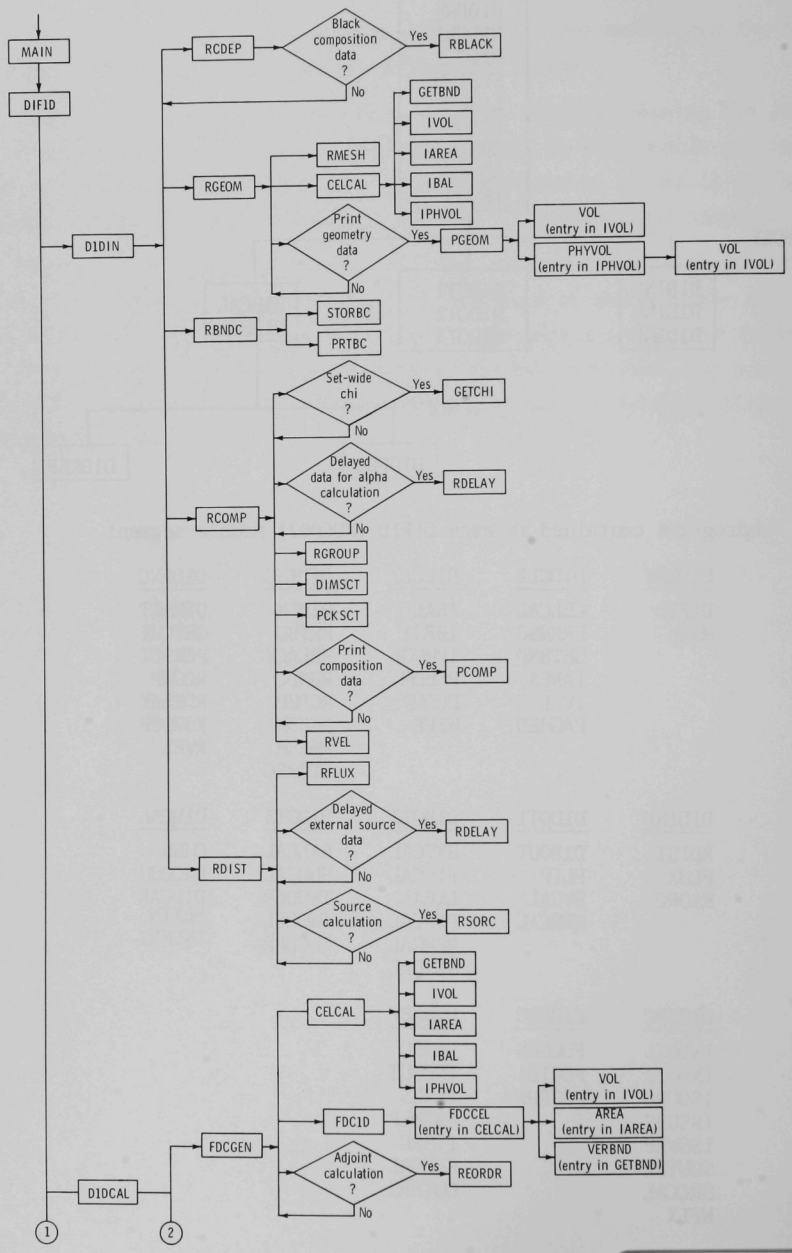
Subprograms contained in each DIF1D (NUC002) module segment

D1DDRV	D1DCL1	D1DCL2	D1DING	D1DINC
DIF1D	CELCAL	IBAL	D1DIN	DIMSCT
Main	ERRMSG	IBFJ1	PRTBC	GETCHI
	GETBND	IPHVOL	RBLACK	PCKSCT
	IAREA	PGEOM	RBNDC	RCOMP
	IVOL	PCOMP	RCDEP	RDELAY
	PAGHED	WIPE	RGEOM	RGROUP
			RMESH	RVEL
			STORBC	

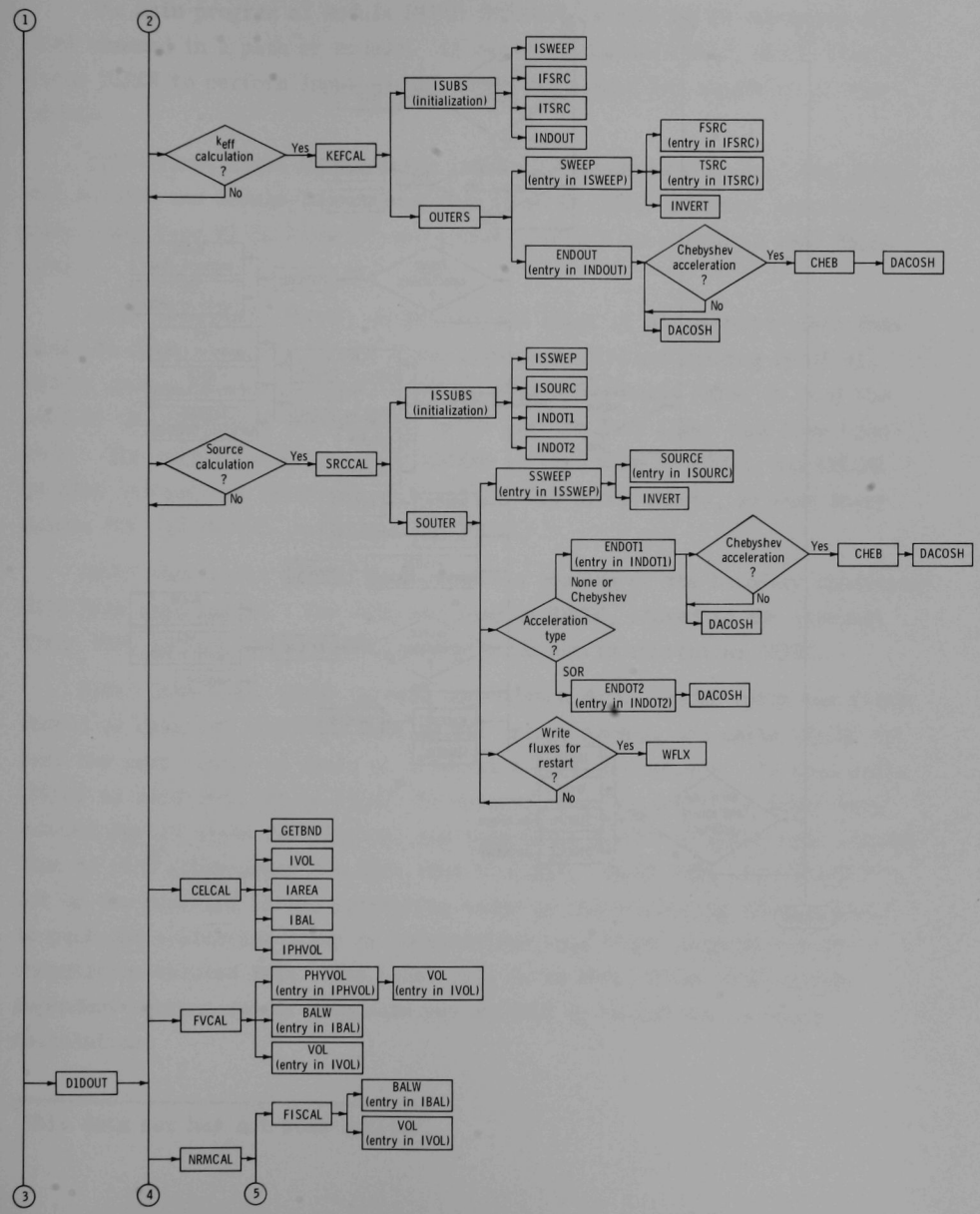
D1DINF	D1DOT1	D1DOT2	D1DOT3	D1DCAL
RDIST	D1DOUT	EXTCAL	BALCAL	CHEB
FLUX	FLIP	FISCAL	SUMCAL	DACOSH
RSORC	FVCAL	LKCAL	TWODCR	D1DCAL
	NRMCAL	PWRCAL	TWODFL	FLXIN
		REMCAL	TWODPR	INVERT
		SCTCAL		

D1DSRC	D1DFDC	D1DKEF
INDOT1	FDCGEN	IFSRC
INDOT2	FDC1D	INDOUT
ISOURC	REORDR	ISUBS
ISSUBS		ISWEEP
ISSWEP		ITSRC
SOUTER		KEFCAL
SRCCAL		OUTERS
WFLX		

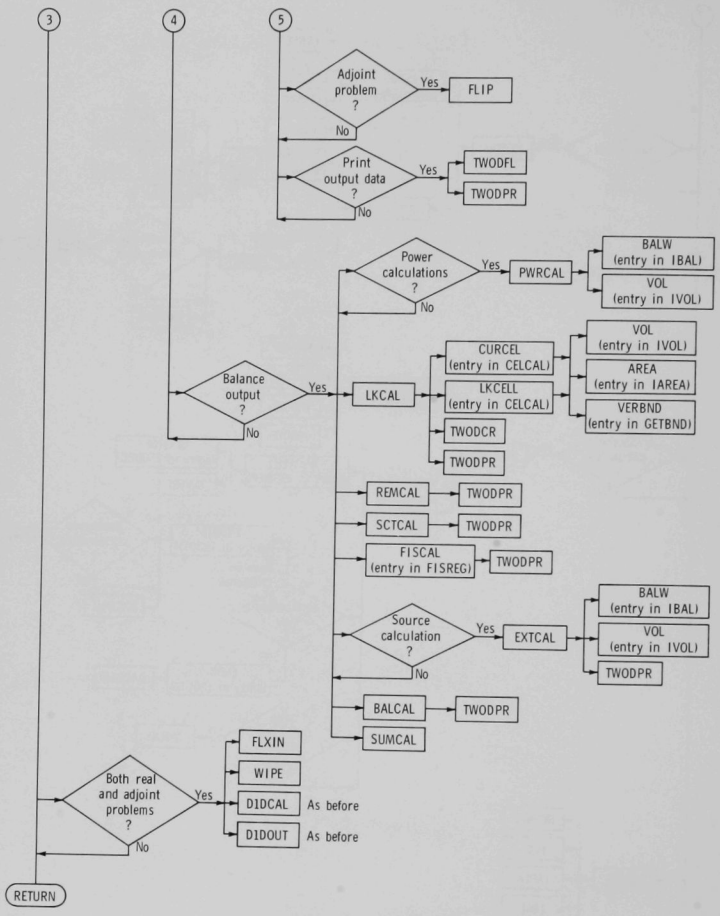
b. Diagrammatic Subprogram Logic Flow



b. Diagrammatic Subprogram Logic Flow (Contd.)



b. Diagrammatic Subprogram Logic Flow (Contd.)



c. Descriptive Subprogram Logic Flow

The main program of module DIF1D (NUC002) is entered by execution of a LINK command in a path or module. It calls subroutine DIF1D, which first calls D1DIN to perform input of all data sets needed for execution of the module.

D1DIN first calls RCDEP, which inputs module-dependent data from data set A.DIF1D and module-independent data from SP.CICN. If black composition data (card type 07 in A.DIF1D) are present, RBLACK is called to read these data.

D1DIN then calls RGEOM, which controls input of all geometry data from data set GEOM. Reading of the first record of GEOM and setting up of all arrays needed is accomplished in RGEOM, which then calls RMESH to read the rest of the GEOM data and PGEOM to print out the data after they have been read. The initialization call to CELCAL is also made by RGEOM, and CELCAL in turn initializes functions IVOL and IPHVOL, among others, so that entry points VOL and PHYVOL can be used by PGEOM.

D1DIN then calls RBNDC, which controls reading of the boundary condition data from data set BC. The data are read in RBNDC, stored in the internal array BNDC in subroutine STORBC, and printed out in subroutine PRTBC.

D1DIN then calls RCOMP to read composition data. RCOMP reads the first record of data set XS.C.MIN, sets up all arrays needed, and calls GETCHI to read the next record if there is a set-wide fission spectrum. It then calls RDELAY to read data set XS.DELAY* if delayed neutrons are to be taken into account for an alpha calculation, and then calls RGROUP once for each composition to read group-dependent data from XS.C.MIN. RCOMP then calls DIMSCT to set up the position of the scattering terms in the scattering array, PCKSCT to pack the scattering array by compositions, and PCOMP to print out the composition-related data. Its final call is to RVEL, which reads group-dependent neutron speeds from data set XS.ISO2 or XS.ISO for an alpha calculation.

---

*This data set has not been defined.

D1DIN then calls RDIST, which controls reading of distributed data. RDIST sets up all arrays needed for these data and calls RFLUX to read flux input from either data set FR.D1 or data set FA.D1, depending on the problem type. If the problem is a source calculation and if the external source is due to delayed neutrons, RDIST calls RDELAY to read data set XS.DELAY.* For any source calculation, RDIST then calls RSORC to read the external fixed source data from ES.DID and to distribute the source data using fission spectrum weighting if this option is specified.

Following the call to RDIST, D1DIN prints the amount of POINTR array space required for the input section and packs the container arrays by calling PURGE and PURGEB. It then returns control to DIF1D.

DIF1D then calls to RDIST, which controls the calculation of the solution to the problem. D1DCAL first calls FDCGEN, which controls generation of the finite-difference coefficients.

FDCGEN sets up the needed arrays and calls CELCAL to initialize routines to be used. It then calls FDC1D to control the calculation of the coefficients. FDC1D calls entry point FDCCEL of CELCAL for each interval and group to calculate the coefficients and the external inhomogeneous boundary terms, if any. FDC1D then completes calculation of these and related terms and reorders these arrays by groups if an adjoint problem is to be done. Then, if this is an adjoint problem, FDCGEN calls REORDR to re-order other group-dependent arrays before returning control to D1DCAL.

D1DCAL then determines if the problem is a k-effective calculation or a source calculation: if the former, it calls KEFCAL, and if the latter, SRCCAL.

KEFCAL controls k-effective calculation. It first sets up necessary arrays and prints the space required in POINTR arrays for the calculation section. It then calls ISUBS to initialize subroutines and then transfers control to OUTERS. OUTERS controls the number of power iterations done and writes FR.D1 or FA.D1 after a specified number of passes if this restart option has been specified. For every iteration, it calls SWEEP (entry in ISWEEP) to perform the mesh sweep and ENDOUT (entry in INDOUT) to perform Chebyshev acceleration if desired, to compute monitoring information, and to test for convergence.

---

*This data set has not been defined.

SWEEP calls FSRC (entry in IFSRC) to compute the fission source for every point and group. Then, for each group, it calls TSRC (entry in ITSRC) to compute the total source including scattering for each channel and INVERT to perform a Choleski inversion of the resulting tri-diagonal matrix. After this has been completed for all groups, SWEEP performs upscattering iterations, if any, in the same manner, calling TSRC and INVERT.

ENDOUT calculates and prints monitoring information and performs Chebyshev acceleration, if it is called for, by calling CHEB to obtain the Chebyshev coefficients and then by modifying the flux appropriately. It also tests for convergence and returns a flag to OUTERS, which returns control to D1DCAL and then to subroutine DIF1D if the problem has converged or has exceeded the maximum number of outer iterations and continues to another iteration if not.

SRCCAL, which controls source calculations, works in a similar fashion. It sets up necessary arrays, prints the space required in POINTR arrays for the calculation section, and initializes subroutines by calling ISSUBS. It then calls SOUTER to control the source iterations. SOUTER also writes FR.D1 or FA.D1 after a specified number of passes if this option has been requested, calling WFLX to write the second record. For each iteration, SOUTER calls SSWEEP (entry in ISSWEP) to perform the mesh sweep by groups, and either ENDOT1 (entry in INDOT1) or ENDOT2 (entry in INDOT2), depending on the type of acceleration chosen, to perform acceleration if required, to compute monitoring information, and to test for convergence.

For each group SSWEEP calls SOURCE (entry in ISOURC) to compute the total source, including the scattering source, the fission source, the fixed external source, and the external source due to inhomogeneous interface conditions, for all mesh points in that group. It then calls INVERT to perform a Choleski inversion of the resulting tri-diagonal matrix.

ENDOT1 is used by problems for which no acceleration or Chebyshev acceleration has been specified; ENDOT2 is used by problems requiring successive over-relaxation. ENDOT1 calls CHEB to obtain the Chebyshev coefficients, if needed; ENDOT2 computes the extrapolation parameter internally, using the function DACOSH. Either routine performs acceleration if desirable, computes monitoring information, and tests for convergence. If the problem has converged,

or if the maximum number of source iterations has been exceeded, control is returned to SOUTER and thence to D1DCAL and subroutine DIF1D; otherwise, SOUTER continues to another iteration.

Subroutine DIF1D then calls D1DOUT to control output of data sets and printed output, if desired. D1DOUT purges the POINTR container arrays, sets up necessary arrays, and prints the space required for the output section. It calls CELCAL to initialize various routines used. It then calls FVCAL to compute regional flux integrals and then NRMCAL. NRMCAL computes regional flux averages and regional flux integrals power-normalized for burnup, if desired, normalizes fluxes and flux-related terms for a k-effective calculation, calls FISCAL to calculate fission density and fission source integrals by mesh intervals and group, and then calls FLIP to re-order these quantities by group if the problem is adjoint. NRMCAL then outputs data sets: FR.D1 or FA.D1 for fluxes, FR.PN for a real problem for which power-normalization of fluxes for burnup is specified, and FSR.D1 or FSA.D1 if fission source integral output is requested. Then, if printing of these quantities has been specified, NRMCAL prints all flux-related and fission-related quantities thus far computed, utilizing printing routines TWODPR and TWODFL.

D1DOUT then returns control to subroutine DIF1D if further output is not desired. If power-related terms are desired, D1DOUT calls PWRCAL to compute and print them. D1DOUT then calls LKCAL, which computes the leakage contribution to the balance equation (using entry LKCELL of CELCAL) and the current across region interfaces (using entry CURCEL of CELCAL), and prints them by calling TWODPR and TWODCR.

D1DOUT then calls REMCAL to compute the removal contribution to the balance equation and print it using TWODPR; SCTCAL to compute the scattering source contribution to the balance equation and print it using TWODPR; FISREG (entry in FISCAL) to compute the fission source contribution to the balance calculation and print it using TWODPR; and BALCAL to calculate the balance term and print it using TWODPR. Finally, SUMCAL is called to sum all balance-related terms over regions, over groups, and over the reactor, and to print these quantities. Following this, control is returned to subroutine DIF1D.

If the problem being solved was either real or adjoint, this completes the program. If the problem type was specified as both, however, WIPE is called to remove output arrays from the POINTR containers and the containers are purged. FLXIN is called to read a user-supplied flux guess for the adjoint part of the calculation. If no adjoint flux guess is provided, the latest real flux generation is used as the initial flux guess. D1DCAL and D1DOUT are called again to perform the second (adjoint) calculation. Following this, control is returned by subroutine DIF1D to the main program, which calls FREE to free storage and then returns control to the program which linked to the module.

4. DIF1D (NUC002) Subprogram Descriptions

This section contains descriptions of all subprograms which are part of the DIF1D module. The descriptions appear in alphabetical order by subprogram name.

Each subprogram description begins with a listing of the calling sequence of the subprogram, and of its entry points, if any. Subsection 1 is a list of the arguments of the subprogram and its entry points, in order of appearance, with the type of each argument, its dimensions in the subprogram if it is an array, and its definition or use. The four types of variables used are I*2, denoting an integer halfword (16 bits); I*4, denoting an integer fullword (32 bits); R*4, denoting a floating-point fullword; and R*8, denoting a floating-point doubleword (64 bits). Variables used as array dimensions are defined either as arguments or as members of common blocks.

When the character "*" appears as an argument, it indicates an optional return to a statement number specified in the calling subprogram. See the IBM FORTRAN manuals for further explanation.

Subsection 2 contains the names of all common blocks used by the subprogram. Each is followed by a list of the variables actually used from the common block within the subprogram, together with their types and definitions. The dimensions, if any, follow the variable name in parentheses.

Exceptions to this are the /BLKSTR/ and /SINGLE/ common blocks. For each, its presence is noted; for the /SINGLE/ block, the locations of variables used are listed, together with their local names and definitions.

Subsection 3 contains a listing of important variables which are local to the subprogram, in a format like that described for subsection 2.

Subsection 4 describes the functions performed by the subprogram in terms of the overall purpose of the module.

Subsection 5 lists all subprograms called by the subprogram or any of its entry points. Subsection 6 lists the subprograms which call the subprogram or

any of its entry points. Many of these subprograms are shared by the D1DSCH module; when they are called from different subprograms in the two modules, this fact is noted.

Subsection 7 describes error messages which can be generated by the subprogram, together with the action taken.

More detailed explanations of the contents of common blocks, and of the variable-dimensioned arrays stored in the large container array /BLKSTR/, can be found in sections 5 and 6.

Subroutine BALCAL (BLNC, LEAK, RMVL, SCAT, FISS, EXTS, RMAX)

1. Arguments

Name	Type	Dimension	Definition
BLNC	R*8	RMAX*GMAX	Balance by region and group
LEAK	R*8	RMAX*GMAX	Leakage contribution to balance, by region and group
RMVL	R*8	RMAX*GMAX	Removal contribution to balance, by region and group
SCAT	R*8	RMAX*GMAX	Scattering source contribution to balance, by region and group
FISS	R*8	RMAX*GMAX	Fission source contribution to balance, by region and group
EXTS	R*8	RMAX*GMAX	External source contribution to balance, by region and group
RMAX	I*4	--	Number of regions defined

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(2)	KEFF	k_{eff} for k-effective calculation	
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
INT(37)	GMAX	Number of groups defined	

3. Local Variables

Name	Type	Definition	
TITLE(9)	R*8	Title for balance for TWODPR	
ROWT	R*8	'REGION' for row title for TWODPR	
COLT	R*8	'GROUP' for column title for TWODPR	
SORC	R*8	'SOURCE' for calculation type	
ADJ	R*8	'ADJOINT' for problem type	
FIXK	R*8	$1./k_{eff}$ for k-effective calculation 1. for source calculation	
JPONE	I*4	Problem type flag for TWODPR	0 - real 2 - adjoint

4. Functions and tasks performed by this subroutine

This subroutine computes the balance term BLNC by region and group and prints the results by calling TWODPR.

The balance term is computed as

BLNC(R,G) = -LEAK(R,G) - RMVL(R,G) + SCAT(R,G) + FIXK*FISS(R,G)

for a k-effective calculation, where FIXK is $1./k_{eff}$.

For a source calculation, the balance term is

BLNC(R,G) = -LEAK(R,G) - RMVL(R,G) + SCAT(R,G) + FIXK*FISS(R,G) + EXTS(R,G), where FIXK is 1., reflecting the fact that the scaling factor $\mu = 1./k_{fixed}$ already appears in the fission source term.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
TWODPR

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine CELCAL (REGMSH, CMPMSH, X, DX, BSQ, BNDC, DC, REM, FLUX, GGEOM, RMAX, BCMAX, CMAX, IMAX)

Entry Points

```
Entry FDCCEL (AIG, CIG, EIG, BCIGL, BCIGR, I, G, IERR)
Entry LKCELL (FLUX, LIG, I, G, IERR)
Entry FLXCEL (FLUX, FIGL, FIGR, I, G, IERR)
Entry DERCEL (FLUX, DIGL, DIGR, I, G, IERR)
Entry CURCEL (FLUX, JIGL, JIGR, I, G, IERR)
Entry LKFACE (FLUX, LIGL, LIGR, I, G, IERR)
```

1. Arguments

Name	Type	Dimension	Definition
REGMSH	I*2	IMAX	Region-to-mesh correspondence
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence
X	R*8	IMAX+1	Mesh line locations
DX	R*8	IMAX	Mesh interval lengths
BSQ	R*8	RMAX*2	Buckling B^2 by region and direction
REGHHT	R*8	RMAX*2	Region half-height $(H_L)_r$ by direction
BNDC	I*4 R*4	8*BCMAX	Boundary and interface condition descriptions and coefficients, where the first four quantities for each boundary are integers and the second four are floating-point quantities
DC	R*8	CMAX*GMAX	Diffusion coefficients D_c^g
REM	R*8	CMAX*GMAX	Removal cross section σ_R
FLUX	R*8	IMAX*GMAX	Fluxes by mesh interval and group (FLX(1,1,NTH) in BLK array)
GGEOM	R*8	--	Geometry type 'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'
RMAX	I*4	--	Number of regions defined

Name	Type	Dimension	Definition
BCMAX	I*4	--	Number of boundary conditions defined (NBCDEF)
CMAX	I*4	--	Number of compositions defined
IMAX	I*4	--	Number of mesh intervals defined
I	I*4	--	Mesh interval index
G	I*4	--	Group index
IERR	I*4	--	BNDC boundary condition error flag 0 - boundary condition 1 or 2 specified 1 - boundary condition 3 (general internal interface conditions) specified
AIG	R*8	--	Finite difference coefficient on right of cell i,g
CIG	R*8	--	Finite difference coefficient on left of cell i,g
EIG	R*8	--	Finite difference coefficient in center of cell i,g
BCIGL	R*8	--	Source due to inhomogeneous boundary condition on right of cell i,g
LIG	R*8	--	Total leakage for cell i,g
FIGL	R*8	--	Flux on left cell boundary for cell i,g
DIGL	R*8	--	Derivative of flux on left boundary for cell i,g
DIGR	R*8	--	Derivative of flux on right boundary for cell i,g
JIGL	R*8	--	Current across right cell boundary for cell i,g
LIGL	R*8	--	Leakage across left cell boundary for cell i,g
LIGR	R*8	--	Leakage across right cell boundary for cell i,g

2. Common Blocks

/GNAME/

Variables used from this common block

Name	Type	Definition
GEOM	R*8	Geometry type (from GGEOM)

3. Local Variables

Name	Type	Definition
REGNO	I*4	Region number corresponding to interval I (from REGMSH)
CMPNO	I*4	Composition number corresponding to interval I, I + 1, or I - 1 (from CMPMSH)
IDUM	I*4	Dummy variable for initializing functions
JUMP	I*4	Flag indicating entry point used 1 - FDCCEL 2 - LKCELL 3 - others
VOLI	R*8	Volume per unit height of cell I
AREAR	R*8	Area of cell I on right
AREAL	R*8	Area of cell I on left
DCIG	R*8	Diffusion coefficient corresponding to I (from DC)
DCIP1G	R*8	Diffusion coefficient corresponding to I + 1 (from DC)
DCIM1G	R*8	Diffusion coefficient corresponding to I - 1 (from DC)
MIP1	I*4	-(I+1), used to get right cell boundary from VERBND
A	R*8	First coefficient of boundary condition on specified mesh line, if one is defined (from VERBND)

Name	Type	Definition
B	R*8	Second coefficient of boundary condition on specified mesh line, if one is defined (from VERBND)
C	R*8	Third coefficient of boundary condition on specified mesh line, if one is defined (from VERBND)
D	R*8	Fourth coefficient of boundary condition on specified mesh line, if one is defined (from VERBND)
ITYPER	I*4	Boundary condition type on right cell boundary (from VERBND) 0 - continuity of flux and current 1 - $A\phi' + B\phi = C$, where A = C1, B = C2, and C = C3 2 - $A\phi' + B\phi = C$, where A = C1*DCIG, B = C2, and C = C3*DCIG 3 - $\phi_{i+1} = A\phi_i + B\phi'_i$, $\phi'_{i+1} = C\phi_i + D\phi'_i$ where A = C1, B = C2, C = C3, and D = C4
ITYPEL	I*4	Boundary condition type on left cell boundary (from VERBND)
DENOM	R*8	Denominator of expression (calculated as below)
DUM1	R*8	1./DENOM if DENOM = 0.; otherwise 0
DUM2	R*8	DUM1* appropriate diffusion coefficient (see below)

4. Functions and tasks performed by this subroutine

CELCAL is the initialization entry; the program's functions are performed by calling the appropriate entry point after initialization.

This subroutine calculates various quantities associated with the operator -div(D·∇ϕ) in one dimension, for a specified cell I and energy group G. The calculations are based on linear approximations; i.e., three-point formulas are used.

The geometries for which calculations can be performed are rectangular slab, cylindrical slab, cylinder, and sphere. Volumes and areas associated with these geometry types for each cell are computed in the separate subprograms IVOL and IAREA.

When the condition of continuity of flux and current across an interface is applied, the finite difference coefficients are

$$a_{i,g} = \frac{D^g_{c(i)}\, D^g_{c(i+1)}\, A_{i_R}}{\frac{1}{2}\,\Delta X_i\, D^g_{c(i+1)} + \frac{1}{2}\,\Delta X_{i+1}\, D^g_{c(i)}}$$

$$c_{i,g} = \frac{D^g_{c(i)}\, D^g_{c(i-1)}\, A_{i_L}}{\frac{1}{2}\,\Delta X_i\, D^g_{c(i-1)} + \frac{1}{2}\,\Delta X_{i-1}\, D^g_{c(i)}}$$

and

$$e_{i,g} = a_{i,g} + c_{i,g} + \left[(\sigma_R)^g_{c(i)} + D^g_{c(i)}\left(B^2_{r(i),1} + B^2_{r(i),2}\right)\right]V_i \,,$$

where buckling exists, or

$$e_{i,g} = a_{i,g} + c_{i,g} + (\sigma_R)^g_{c(i)}\, V_i \,,$$

where buckling does not exist. In these equations, A_{i_R} and A_{i_L} are the areas associated with cell i on the right and left, respectively, and V_i is the volume of cell i, for the geometry specified.

Further, for the same interface condition, the flux on the right boundary of cell i is

$$\phi_{i,g_R} = \frac{\frac{1}{2}\,\Delta X_i\, D^g_{c(i+1)}\,\phi_{i+1,g} + \frac{1}{2}\,\Delta X_{i+1}\, D^g_{c(i)}\,\phi_{i,g}}{\frac{1}{2}\,\Delta X_i\, D^g_{c(i+1)} + \frac{1}{2}\,\Delta X_{i+1}\, D^g_{c(i)}} ,$$

and the flux on the left is analogous. Hence,

$$\phi'_{i,g_R} = \frac{D^g_{c(i+1)}\,(\phi_{i+1,g} - \phi_{i,g})}{\frac{1}{2}\,\Delta X_i\, D^g_{c(i+1)} + \frac{1}{2}\,\Delta X_{i+1}\, D^g_{c(i)}} ,$$

and

$$\phi'_{i,g_L} = \frac{D^g_{c(i-1)}\,(\phi_{i,g} - \phi_{i-1,g})}{\frac{1}{2}\,\Delta X_i\, D^g_{c(i-1)} + \frac{1}{2}\,\Delta X_{i-1}\, D^g_{c(i)}} .$$

For an interface condition of the type $C_1\phi + C_2\phi = C_3$ (where $A = C_1$, $B = C_2$, and $C = C_3$), or of the type $C_1 \cdot D^g_{c(i)}\phi' + C_2\phi = C_3 \cdot D^g_{c(i)}$, $\left(\text{where } A = C_1 \cdot D^g_{c(i)},\ B = C_2, \text{ and } C = C_3 \cdot D^g_{c(i)}\right)$, the finite difference coefficients are

$$a_{i,g} = 0 .$$

$$c_{i,g} = \frac{D^g_{c(i)}\, D^g_{c(i-1)}\, A_{i_L}}{A + \frac{1}{2}\,\Delta X_i B}$$

if the condition appears on the right cell boundary, or

$$a_{i,g} = \frac{D^g_{c(i)}\, D^g_{c(i+1)}\, A_{i_R}}{A + \frac{1}{2}\,\Delta X_i B}$$

$$c_{i,g} = 0.$$

if the condition occurs on the left. The calculation of $e_{i,g}$ remains the same as above in either case.

The flux then becomes, on the right or left,

$$\phi_{i,g_{(R \text{ or } L)}} = \frac{\frac{1}{2}\Delta X_i C + A\phi_{i,g}}{A + \frac{1}{2}\Delta X_i B} .$$

The derivatives of the flux are then

$$\phi'_{i,g_R} = \frac{C - B\phi_{i,g}}{A + \frac{1}{2}\Delta X_i B}$$

$$\phi'_{i,g_L} = \frac{B\phi_{i,g} - C}{A + \frac{1}{2}\Delta X_i B}$$

Additionally, if the boundary condition is inhomogeneous (C = 0.), boundary terms of the form

$$S_{ext_R} = \frac{D^g_{c(i)} CA_{i_R}}{A + \frac{1}{2}\Delta X_i B}$$

$$S_{ext_L} = \frac{D^g_{c(i)} CA_{i_L}}{A + \frac{1}{2}\Delta X_i B}$$

are created.

For any interface condition mentioned, the current across the cell boundaries is simply

$$J_{i,g_R} = -D^g_{c(i)} \phi'_{i,g_R}$$

$$J_{i,g_L} = D^g_{c(i)} \phi'_{i,g_L}$$

and the corresponding leakages are

$$L_{i,g_R} = J_{i,g_R} A_{i_R}$$

$$L_{i,g_L} = J_{i,g_L} A_{i_R}$$

If total cell leakage is requested, it consists of

$$L_{i,g} = L_{i,g_R} + L_{i,g_L} + D^g_{c(i)} \left(B^2_{r(i)_1} + B^2_{r(i)_2}\right) \phi_{i,g} V_i ,$$

where buckling term or terms are defined, and simply

$$L_{i,g} = L_{i,g_R} + L_{i,g_L}$$

where buckling is not defined.

The denominator term common to the calculations performed in this subroutine is computed as follows:

```
DENOM  = .5*(DX(I)*DCIP1G + DX(I+1)*DCIG) on right if ITYPER = 0
       = .5*(DX(I)*DCIM1G + DX(I-1)*DCIG) on left if ITYPEL = 0
       = (A + .5*DX(I)*B)                 if ITYPER or ITYPEL = 1 or 2

 DUM1  = 1./DENOM                         if DENOM ≠ 0
       = 0.                               if DENOM = 0

 DUM2  = DCIP1G*DUM1                      on right if ITYPER = 0
       = DCIM1G*DUM1                      on left if ITYPEL = 0
       = DCIG*DUM1                        if ITYPER or ITYPEL = 1 or 2
```

The left, right, and central finite difference coefficients are then computed as follows, in the order given, if JUMP = 1:

first,

```
  AIG = DCIG*DUM2*AREAR and EIG = AIG              if ITYPER = 0,
```

or

```
  AIG = 0.              and EIG = DUM2*B*AREAR     if ITYPER = 1 or 2;
```

then,

```
CIG = DCIG*DUM2*AREAL and EIG = EIG + CIG          if ITYPEL = 0,
```

or

```
CIG = 0.              and EIG = EIG + DUM2*B*AREAL  if ITYPEL = 1 or 2;
```

and finally,

```
EIG = EIG + (DCIG*(BSQ(REGNO,1) + BSQ(REGNO,2) + REM(CMPNO,G))*VOLI.
```

The homogeneous boundary terms are calculated as follows, also if JUMP = 1:

```
BCIGR   = 0.                                       if ITYPER = 0,
        = DUM2*C*AREAR                             if ITYPER = 1 or 2;

BCIGL   = 0.                                       if ITYPEL = 0,
        = DUM2*C*AREAL                             if ITYPEL = 1 or 2.
```

The flux on the right and left cell boundaries, if JUMP = 2 or JUMP = 3, is calculated as follows:

```
FIGR = .5*(DCIG*FLUX(I,G)*DX(I+1) + DCIPIG*
          FLUX(I+1,G)*DX(I))*DUM1                  if ITYPER = 0,
```

or

```
FIGR = .5*(DX(I)*C + A*FLUX(I,G))*DUM1             if ITYPER = 1 or 2;
FIGL = .5*(DCIG*FLUX(I,G)*DX(I-1) + DCIMIG*
         FLUX(I-1,g)*DX(I))*DUM1                   if ITYPEL = 0,
```

or

```
FIGL = (.5*DX(I)*C + A*FLUX(I,G))*DUM1             if ITYPEL = 1 or 2.
```

The derivative of the flux on the cell boundaries is then calculated as follows:

```
DIGR = DUM2*(FLUX(I+1,G) - FLUX(I,G))              if ITYPER = 0,
```

or

```
DIGR = (C-B*FLUX(I,G))*DUM1                        if ITYPER = 1 or 2;
DIGL = DUM2*(FLUX(I,G) - FLUX(I-1,G))              if ITYPEL = 0,
```

or

```
DIGL = (B*FLUX(I,G) -C)*DUM1                       if ITYPEL = 1 or 2.
```

The currents across the right and left cell boundaries respectively are then simply

```
JIGR = -DCIG*DIGR
```

and

```
JIGL = DCIG*DIGL,
```

and the leakage across the left and right cell boundaries is

```
LIGR = JIGR*AREAR
```

and

```
LIGL = JIGL*AREAL.
```

The total cell leakage, if JUMP = 2, is then computed as follows:

```
LIG = LIGL + LIGR + DCIG*(BSQ(REGNO,1) + BSQ(REGNO,2))*VOLI*FLUX(IG,)
```

5. <u>Subprograms called by this subroutine</u>

```
CELCAL calls IPHVOL
             IVOL
             IBAL
             IAREA
             GETBND

All entry points call VOL(entry point in IVOL)
                      AREA (entry point in IAERA)
                      VERBND (entry point in GETBND)
                      ERRMSG
```

6. <u>Subprograms calling this subroutine</u>

```
CELCAL is called by RGEOM
                    FDCGEN
                    D1DOUT
FDCCEL is called by FDCID
LKCELL is called by LKCAL
FLXCEL is called by - None
DERCEL is called by - None
CURCEL is called by LKCAL
LKFACE is called by - None
```

7. Error messages generated by this subroutine

Message: *****GENERAL INTERNAL INTERFACE CONDITIONS SPECIFIED FOR I = iiii, G = iiii

ERROR NUMBER $\left\{\begin{matrix}1\\2\end{matrix}\right\}$ IN SUBROUTINE CELCAL

Significance: An interface condition (ITYPER or ITYPEL) not equal to 0, 1, or 2 was returned to CELCAL from VERBND (entry in GETBND). This is assumed to indicate that an interface condition of 3, indicating general internal interface conditions, was specified in data set BC, and hence by implication that the internal interface condition type 2 was specified in A.NIP (card type 10, cols. 19-24), for this I and G. An error number 1 indicates that the error was encountered on the right of cell i,g; an error number 2, that it was encountered on the left.

Action taken: The IERR flag is set to 1 and control is returned to the calling program. No calculations are performed.

Subroutine CHEB (ALPHA, BETA, P, SIGMA, A)

1. Arguments

Name	Type	Dimension	Definition
ALPHA	R*8	--	First Chebyshev coefficient α
BETA	R*8	--	Second Chebyshev coefficient β
P	I*4	--	Degree of Chebyshev polynomial p
SIGMA	R*8	--	Estimate of rate of convergence of flux
A	R*8	--	Interval shift factor for Chebyshev acceleration

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
COSHGM	R*8	The hyperbolic cosine of γ, as defined
GAMMA	R*8	The term γ, found by taking the arc-hyperbolic cosine of COSHGM

4. Functions and tasks performed by this subroutine

This subroutine computes the Chebyshev coefficients α and β for the Chebyshev polynomial

$$Q_p(x) = Q_{p-1}(x) + \alpha_p(x - 1)\, Q_{p-1}(x) + \beta_p(Q_{p-1}(x) - Q_{p-2}(x)),$$

used in Chebyshev extrapolation. For a given order p, estimated rate of convergence σ, and interval shift factor a, the coefficients are computed as follows:

$$\alpha_1 = \frac{2}{2 - \sigma - a},$$

$$\beta_1 = 0,$$

for $p = 1$.

$$\alpha_p = \frac{4}{\sigma - a} \cdot \frac{\cosh (p - 1)\gamma}{\cosh p\gamma},$$

$$\beta_p = [1 - \tfrac{1}{2}(\sigma - a)]\, \alpha_p - 1,$$

for $p \geq 2$,

where

$$\cosh \gamma = \frac{2 - \sigma - a}{\sigma - a} .$$

5. Subprograms called by this subroutine

DACOSH

6. Subprograms calling this subroutine

ENDOUT (entry in INDOUT)
ENDOT1 (entry in INDOT1)

7. Error messages generated by this subroutine - None

REAL Function DACOSH (X)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	--	Number for which the arc-hyperbolic cosine is to be calculated

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this function

This function, which is typed REAL, calculates the arc-hyperbolic cosine of a number X, according to the equation

$$\cosh^{-1} (X) = \ln \left(X + \sqrt{X^2 - 1} \right) .$$

5. Subprograms called by this function - None

6. Subprograms calling this function

CHEB
ENDOUT (entry in INDOUT)
ENDOT1 (entry in INDOT1)

7. Error messages generated by this function - None

Subroutine DIF1D(IERR)

1. Arguments

Name	Type	Dimension	Definition	
IERR	I*4	--	Error flag returned to path level	0 - execution of NUC002 1 - execution of NUC002 terminated abnormally

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100),INT(100)	Single-variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT' 'BOTH'
INT(19)	MAXBLK	Size of POINTR container array in bulk storage	
INT(32)	IMAX	Number of mesh intervals defined	
INT(37)	GMAX	Number of groups defined	
INT(94)	KFLAG	Dummy search flag, set to zero	

3. Local Variables

Name	Type	Definition
REAL	R*8	'REAL' for problem type
ADJ	R*8	'ADJOINT' for problem type
BOTH		'BOTH' for problem type
IPROB	I*4	Flag to preserve problem type 0 - 'REAL' or 'ADJOINT' 1 - 'BOTH'
TIME	R*4	Time remaining at beginning of module; total time spent in module in minutes
DUMMY	I*4	Dummy argument to TLEFT
LAST	I*4	Last location used in POINTR core container array
LASTB	I*4	Last location used in POINTR bulk container array

4. Functions and tasks performed by this subroutine

This subroutine is the actual driver for the ARC System one-dimensional diffusion theory k-effective and source calucation module, DIF1D (NUC002). It first calls D1DIN to perform the input for the module. It then uses IPROB to save the problem type if it is 'BOTH', and resets PROB so that to the calculation and output sections the type appears only as 'REAL' or 'ADJOINT'. It then calls D1DCAL and D1DOUT to perform the calculations and output for the module.

For a 'BOTH' problem, DIF1D calls WIPE to wipe out unneeed arrays from the POINTR container BLK, and PURGE and PURGEB to pack the container array in main and in bulk storage. If the problem type is 'BOTH', FLXIN is called to read a user-supplied flux guess for the adjoint part of the calculation. If no adjoint flux guess is provided, the latest real flux generation is used as the initial guess for the adjoint calculation. It then calls D1DCAL and D1DOUT again to do the adjoint problem.

5. Subprograms called by this subroutine

D1DIN
D1DCAL
D1DOUT
FLXIN
TLEFT (IBM routine)
WIPE
PURGE (entry in POINTR)
PURGEB (entry in POINTR)

6. Subprograms calling this subroutine

Main program for module NUC002

7. Error messages generated by this subroutine

Message: *****EXECUTION OF MODULE NUC002 HAS BEEN TERMINATED

Significance: An error has been encountered in the execution of the module which has made successful execution of the module impossible.

Action taken: The error flag IERR is set to 1 and control is returned to the main program for NUC002.

Subroutine DIMSCT (SCTLIM, SCTLOC, SCTDIM, CMAX,*)

1. Arguments

Name	Type	Dimension	Definition
SCTLIM	I*2	2*CMAX	Maximum number of groups of downscattering and maximum number of groups of upscattering for each composition
SCTLOC	I*2	CMAX*GMAX	Location of last scattering element for composition preceding given composition in SCTXS (first CMAX locations only)
SCTDIM	I*4	--	First dimension of scattering array SCTXS
CMAX	I*4	--	Number of compositions defined
*	--	--	Error return for error in POINTR dynamic storage allocation

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy variable used for checking for POINTR errors

4. Functions and tasks performed by this subroutine

This subroutine utilizes the scattering band widths for each composition (set up in array SCTLIM in subroutine RGROUP) to set up, in the first CMAX locations of the array SCTLOC, the position in an array SCTXS of the final element of the scattering array $\sigma_{c-1}^{g'g}$ corresponding to the composition preceding a given composition c(CMPNO).

The position corresponding to the last scattering element for the composition preceding CMPNO = 1 is 0; for each succeeding CMPNO, the position is SCTLOC(CMPNO) = SCTLOC(CMPNO - 1) + SCTLIM(1,CMPNO - 1) + SCTLIM(2,CMPNO - 1) + 1, where SCTLIM(1,CMPNO - 1) + SCTLIM(2,CMPNO - 1) + 1 is the band width corresponding to composition CMPNO - 1.

This routine also computes the first dimension SCTDIM of the scattering array SCTXS; this dimension is computed as the position of the final element of composition CMAX - 1, SCTLOC(CMAX), plus the band width of the last composition, SCTLIM(1,CMAX) + SCTLIM(2,CMAX) + 1. In other words, the first dimension of SCTXS is the sum over all compositions of the band widths associated with each composition; the second dimension is GMAX. Hence, in the scattering array, each composition and group is allocated only the maximum band width associated with the composition.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)

6. Subprograms calling this subroutine

RCOMP

7. Error messages generated by this subroutine - None

Subroutine D1DCAL (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program if error is encountered in any subprogram called

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single-variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'

3. Local Variables

Name	Type	Definition
SORC	R*8	'SOURCE' for calculation type

4. Functions and tasks performed by this subroutine

This subroutine is the driver for the one-dimensional diffusion theory k-effective and source calculation iterations. It calls FDCGEN to generate the finite-difference coefficients and related terms, and then calls either KEFCAL or SRCCAL to control the iterations.

5. Subprograms called by this subroutine

FDCGEN
KEFCAL
SRCCAL

6. Subprograms calling this subroutine

DIF1D (when executing module NUC002)
D1DSCH (when executing module NUC004)

7. Error messages generated by this subroutine - None

Subroutine D1DIN (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program if error is encountered in any subprogram called

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(19)	MAXBLK	Size of POINTR container array in bulk storage

3. Local Variables

Name	Type	Definition
LAST	I*4	Last location used in POINTR core container array
LASTB	I*4	Last location used in POINTR bulk container array

4. Functions and tasks performed by this subroutine

This subroutine is the driver for the input section of the one-dimensional diffusion theory k-effective and source calculation module, DIF1D (NUC002), and the search calculation module D1DSCH (NUC004). It controls input of all data sets needed by these modules, except SP.CRIT for a search calculation.

After the input is done, D1DIN determines and prints the last locations used in main and bulk POINTR container arrays and purges each of the containers of arrays which have been wiped out.

5. Subprograms called by this subroutine

RCDEP	ILAST (entry in POINTR)
RGEOM	ILASTB (entry in POINTR)
RBNDC	PURGE (entry in POINTR)
RCOMP	PURGEB (entry in POINTR)
RDIST	

6. Subprograms calling this subroutine

DIF1D (when executing module NUC002)
D1DSCH (when executing module NUC004)

7. Error messages generated by this subroutine - None

Subroutine D1DOUT (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program if error is encountered in any subprogram called

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single-variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
FLT(31)	GEOM	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'
INT(7)	KOUT	Output flag	<2 - no printed output 2 - balance output 3 - balance plus power output
INT(19)	MAXBLK	Size of POINTR container array in bulk storage	
INT(31)	RMAX	Number of regions defined	
INT(32)	IMAX	Number of mesh intervals defined	
INT(34)	BCMAX	Number of non-continuity boundary conditions defined	
INT(35)	CMAX	Number of compositions defined	
INT(36)	FSCMAX	Number of fissionable compositions defined	
INT(37)	GMAX	Number of groups defined	
INT(40)	CHIDIM	First dimension of fission spectrum array CHI	
INT(41)	SCTDIM	First dimension of scattering array SCTXS	

3. Local Variables

Name	Type	Definition
SORC	R*8	'SOURCE' for calculation type
ADJ	R*8	'ADJOINT' for problem type
RMAXP1	I*4	RMAX + 1 for defining size of current-related arrays

4. Functions and tasks performed by this subroutine

This subroutine is the driver for the output section of the one-dimensional diffusion theory k-effective and source calculation module and the one-dimensional diffusion theory search module. It first wipes out arrays not needed and calls PURGE and PURGEB to pack the POINTR container array. It then sets up storage for all arrays needed and calls CELCAL to initialize various subroutines and functions needed.

It then calls FVCAL to compute regional flux integrals and NRMCAL to normalize fluxes for a k-effective calculation or search problem, to write desired data sets, and to print the quantities written on data sets, if desired (KOUT $\geq$2).

Following this, if power calculations are desired for a real problem (KOUT = 3), the subroutine calls PWRCAL: then, if KOUT $\geq$2, it calls subroutines to calculate and print the rest of the output balance quantities: LKCAL for leakage, REMCAL for removal, SCTCAL for scattering source, FISREG for fission source, EXTCAL for external source for a source calculation, BALCAL for the balance terms, and SUMCAL for the sums over regions, over groups, and over the entire reactor.

5. Subprograms called by this subroutine

WIPOUT (entry in POINTR)	PWRCAL
PURGE (entry in POINTR)	LKCAL
PURGEB (entry in POINTR)	REMCAL
PUTPNT (entry in POINTR)	SCTCAL
IGET (entry in POINTR)	FISREG (entry in FISCAL)
CELCAL	EXTCAL
FVCAL	BALCAL
NRMCAL	SUMCAL

6. Subprograms calling this subroutine

DIF1D (when executing module NUC002)
D1DSCH (when executing module NUC004)

7. Error messages generated by this subroutine - None

Subroutine ERRMSG (NAM1, NUM)

1. Arguments

Name	Type	Dimension	Definition
NAM1	R*4	2	Name of subprogram calling this subroutine (A8)
NUM	I*4	--	Number assigned to error in subprogram calling this subroutine

2. Common Blocks

/NAMBLK/

Variables used from this common block

Name	Local Name	Definition
NAME	NAM2(1)	Name of subprogram calling this subroutine (A8) (NAM2 is R*4 and is dimensioned 2)

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine prints the name of the subprogram calling it and the error number assigned to the error which occurred, in the format

ERROR NUMBER iiiii IN SUBROUTINE aaaaaaaa

and returns control to the calling subroutine.

The name is input as an R*4 array of dimension 2 in NAM1 and is set equal to the R*4 array NAM2 which is equivalenced to the R*8 member NAME of common block /NAMBLK/, to allow the name to be kept in a common block.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FDCCEL (entry in CELCAL) LKCELL (entry in CELCAL) FLXCEL (entry in CELCAL) DERCEL (entry in CELCAL) CURCEL (entry in CELCAL) LKFACE (entry in CELCAL) PCOMP RCDEP RGEOM RBNDC PRTBC RCOMP RVEL RDIST RSORC RFLUX OUTERS ENDOUT (entry in INDOUT) SOUTER ENDOT1 (entry in INDOT1) FLXIN	(when executing module NUC002)
D1DSCH SRCHIN RDDDX RDBSQ DELBSQ DELX FDCCEL (entry in CELCAL) LKCELL (entry in CELCAL) FLXCEL (entry in CELCAL) DERCEL (entry in CELCAL) CURCEL (entry in CELCAL) LKFACE (entry in CELCAL) PCOMP RCDEP RGEOM RBNDC PRTBC RCOMP RVEL RDIST RFLUX OUTERS ENDOUT (entry in INDOUT) WSRCH RCORD1 RCORD2 RCORD3 RCORD4 FLXIN	(when executing module NUC004)

7. Error messages generated by this subroutine - None

Subroutine EXTCAL (EXTS, FIXSRC, REGMSH, IMAX, RMAX)

1. Arguments

Name	Type	Dimension	Definition
EXTS	R*8	RMAX*GMAX	External source contribution to balance for source calculation
FIXSRC	R*8	IMAX*GMAX	Fixed external source $S_{i,g}$ by mesh interval and group
REGMSH	I*2	IMAX	Region-to-mesh correspondence
IMAX	I*4	--	Number of intervals defined
RMAX	I*4	--	Number of regions defined

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
INT(37)	GMAX	Number of groups defined	

3. Local Variables

Name	Type	Definition	
TITLE	R*8	Title for external source for TWODPR	
ROWT	R*8	'REGION' for row title for TWODPR	
COLT	R*8	'GROUP' for column title for TWODPR	
ADJ	R*8	'ADJOINT' for problem type	
JPONE	I*4	Problem type flag for TWODPR	0 - real 2 - adjoint

4. Functions and tasks performed by this subroutine

This subroutine computes the external source contribution to the balance equation by region and group for a source problem, and prints the results by calling TWODPR. The external source term $ES_{r,g}$ is defined as

$$ES_{r,g} = W_r \sum_{i \varepsilon r} S_{i,g} \cdot V_i,$$

where $S_{i,g}$ is the external source, V_i the mesh interval volume per unit height from function VOL, and W_r the balance integration weighting factor from function BALW.

5. Subprograms called by this subroutine

BALW (entry in IBAL)
VOL (entry in IVOL)
IGET (entry in POINTR)
TWODPR

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine FDCGEN (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program if error is encountered in any subprogram called

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single-variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
FLT(31)	GEOM	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'
INT(31)	RMAX	Number of regions defined	
INT(32)	IMAX	Number of mesh intervals defined	
INT(34)	BCMAX	Number of non-continuity boundary conditions defined (NBCDEF)	
INT(35)	CMAX	Number of compositions defined	
INT(36)	FSCMAX	Number of fissionable compositions defined	
INT(37)	GMAX	Number of groups defined	
INT(40)	CHIDIM	First dimension of fission spectrum array CHI	
INT(41)	SCTDIM	First dimension of scattering array SCTXS	

3. Local Variables

Name	Type	Definition
ADJ	R*8	'ADJOINT' for problem type
SORC	R*8	'SOURCE' for calculation type

4. Functions and tasks performed by this subroutine

This subroutine controls the generation of the finite-difference coefficients $a_{i,g}$, $c_{i,g}$, and $e_{i,g}$ by mesh interval and group for the one-dimensional diffusion theory modules, along with calculation of the related term $r_{i,g}$ required in the Choleski inversion of the tri-diagonal finite difference coefficient matrix and calculation of the external source due to inhomogeneous interface conditions, if any.

The subroutine allocates storage for the required arrays, initializes CELCAL (an entry point of which will be used in the generation of the coefficients), and calls FDC1D to calculate the described data. Then, for an adjoint problem, all group-related arrays are re-ordered by group so that the calculations can be performed in the same manner as the calculations for a real problem.

5. Subprograms called by this subroutine

REDEF (entry in POINTR)
IGET (entry in POINTR)
CELCAL
FDC1D
REORDR
DUMP (entry in POINTR)

6. Subprograms calling this subroutine

D1DCAL

7. Error messages generated by this subroutine - None

Subroutine FDC1D (REGMSH, CMPMSH, DX, BSQ, DC, REM, FSCMP, NUFIS, CHI, EXTBC, A, C, E, GAMMA, R, VEL, CMAX, FSCMAX, CHIDIM, RMAX, IMAX,*)

1. Arguments

Name	Type	Dimension	Definition
REGMSH	I*2	IMAX	Region-to-mesh correspondence
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence

Name	Type	Dimension	Definition
DX	R*8	IMAX	Mesh interval lengths
BSQ	R*8	RMAX*2	Buckling B^2 by region and direction
DC	R*8	CMAX*GMAX	Diffusion coefficients D_c^g
REM	R*8	CMAX*GMAX	Removal cross section σ_R
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
NUFIS	I*4	CMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi_c^{g'g}$
EXTBC	R*8	IMAX*GMAX for source calculation 0 for k_{eff} calculation	Source due to inhomogeneous interface conditions
A	R*8	IMAX*GMAX	Finite difference coefficient on right of cell for all groups
C	R*8	IMAX*GMAX	Finite difference coefficient on left of cell for all groups
E	R*8	IMAX	Central finite difference coefficient for cell for given group
GAMMA	R*8	IMAX	Recurrence coefficients for Choleski inversion for all cells for given group
R	R*8	IMAX*GMAX	Reciprocals of diagonal elements of lower triangular matrix L of tri-diagonal matrix A = LU for Choleski inversion
VEL	R*8	GMAX	Neutron speeds v_g by group
CMAX	I*4	--	Number of compositions defined
FSCMAX	I*4	--	Number of fissionable compositions defined

Name	Type	Dimension	Definition
CHIDIM	I*4	--	First dimension of fission spectrum array CHI
RMAX	I*4	--	Number of regions defined
IMAX	I*4	--	Number of mesh intervals defined
*	--	--	Error return for error in POINTR dynamic storage allocation

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(2)	KEFF	k_{eff} for k-effective calculation	
FLT(3)	ALPHA	Inverse reactor time period α	
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
FLT(6)	MU	$1/k_{fixed}$ factor for sourc calculation	
FLT(31)	GEOM	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'
FLT(32)	XL	Left reactor boundary (X(1))	
INT(37)	GMAX	Number of groups defined	
INT(39)	MTRCHI	Fission spectrum array type	0 - χ-vector 1 - χ-matrix

3. Local Variables

Name	Type	Definition
SORC	R*8	'SOURCE' for calculation type
ADJ	R*8	'ADJOINT' for problem type
DUMMY	R*8	Dummy variable for checking for POINTR errors
BCIGL	R*8	Source due to inhomogeneous interface conditions on left of cell I
BCIGR	R*8	Source due to inhomogeneous interface conditions on right of cell I
IERR	I*4	Error flag returned from FDCCEL (entry in CELCAL)
REGNO	I*4	Region number corresponding to given mesh interval I
CMPNO	I*4	Composition number corresponding to given mesh interval I
NGMAX	I*4	GMAX/2, used in re-ordering arrays for adjoint problem
SAVE	R*8	Dummy variable used in re-ordering arrays for adjoint problem

4. Functions and tasks performed by this subroutine

This subroutine calculates the finite difference coefficients $a_{i,g}$, $c_{i,g}$, and $e_{i,g}$ for each mesh interval and group, as well as the recursion coefficients $\gamma_{i,g}$ and the related terms $r_{i,g}$ used in Choleski inversion of the tri-diagonal finite difference coefficient matrix A, and the external source due to inhomogeneous interface conditions.

FDCCEL, an entry in CELCAL, is called to obtain the coefficients $a_{i,g}$, $c_{i,g}$, and $e_{i,g}$ and the source due to inhomogeneous interface conditions.

FDC1D then sets $a_{1,g}$ to 1. for all groups g, and modifies the central finite difference coefficient $e_{i,g}$ by adding an

$$\frac{\alpha}{v_g} V_i$$

term. The coefficients $a_{i,g}$ and $c_{i,g}$ are stored for all intervals and groups; the coefficients $e_{i,g}$ are stored for all intervals for one group only at a time.

The recurrence coefficients $\gamma_{i,g}$ are then computed from the finite difference coefficients for one group.

$$\gamma_{1,g} = \frac{a_{1,g}}{e_{1,g}} ;$$

for each successive i, i = 2, ..., I,

$$\gamma_{i,g} = \frac{a_{i,g}}{e_{i,g} - c_{i,g}\,\gamma_{i-1,g}} .$$

From the $\gamma_{i,g}$'s, for each group, the terms $r_{i,g}$ are computed and stored for all mesh points and groups:

$$r_{i,g} = \frac{\gamma_{i,g}}{a_{i,g}}$$

or, where $a_{i,g} = 0$,

$$r_{i,g} = \frac{1}{e_{i,g}} - c_{i,g}\,\gamma_{i-1,g} .$$

For an adjoint problem, this subroutine then re-orders the arrays A, C, and R by groups, so that the same subprograms that are used for a real problem can be used for an adjoint problem.

5. <u>Subprograms called by this subroutine</u>

IPTERR (entry in POINTR)
FDCCEL (entry in CELCAL)
VOL (entry in IVOL)
DUMP (entry in POINTR)

6. Subprograms calling this subroutine

FDCGEN

7. Error messages generated by this subroutine - None

Subroutine FISCAL (FISDEN, FISSOR, FISS, FISAVG, FSCMP, NUFIS, CHI, FLX, REGVOL, REGMSH, CMPREG, CHIDIM, FSCMAX, IMAX, RMAX, GMAX, CMAX, FSOPT)

Entry Points

Entry FISREG

1. Arguments

Name	Type	Dimension	Definition
FISDEN	R*8	IMAX*GMAX	Fission density by mesh interval and group
FISSOR	R*8	IMAX*GMAX	Fission source integrals by mesh interval and group
FISS	R*8	RMAX*GMAX	Fission source integrals by region and group
FISAVG	R*8	RMAX*GMAX	Average fission source by region and group
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
NUFIS	R*8	CMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8		Fission spectrum $\chi_c^{g'g}$
FLUX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
REGVOL	R*8	RMAX	Physical region volumes V_r (from function PHYVOL)

Name	Type	Dimension	Definition
REGMSH	I*2	IMAX	Region-to-mesh correspondence
CMPREG	I*2	RMAX	Composition-to-region correspondence
CHIDIM	I*4	--	First dimension of fission spectrum array CHI
FSCMAX	I*4	--	Number of fissionable compositions defined
IMAX	I*4	--	Number of mesh intervals defined
RMAX	I*4	--	Number of regions defined
GMAX	I*4	--	Number of groups defined
CMAX	I*4	--	Number of compositions defined
FSOPT	I*4	--	Fission source option: 0 - do not write FSR.D1 or FSA.D1; 1 - write FSR.D1 or FSA.D1

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
FLT(6)	MU	$1/k_{fixed}$ factor for source calculation	
INT(39)	MTRCHI	Fission spectrum array type	0 - χ-vector 1 - χ-matrix
INT(51)	NTH	Index of nth-generation flux in array FLX	

3. Local Variables

Name	Type	Definition	
TITLE(9)	R*8	Title for fission source by region and group for TWODPR	
ROWT	R*8	'REGION' for row title for TWODPR	
COLT	R*8	'GROUP' for column title for TWODPR	
TITLE2(9)	R*8	Title for average fission source by region and group for TWODPR	
ADJ	R*8	'ADJOINT' for problem type	
SORC	R*8	'SOURCE' for calculation type	
FISLOC	I*4	Location of given element in fission spectrum array CHI for χ-matrix	
JPONE	I*4	Problem type flag for TWODPR	0 - real 2 - adjoint

4. Functions and tasks performed by this subroutine

The FISCAL entry of this subroutine calculates the fission density FISDEN and the fission source integral FISSOR by mesh interval and group. The FISREG entry computes and prints the fission source integral FISS and the average fission source FISAVG by region and group, based on the already-computed fission source by interval and group.

For a real problem, the fission density is defined as

$$FD_{i,g} = W_r \sum_{g'=1}^{G} \chi_c^{g'g} \cdot (\nu \cdot \sigma_f)_c^{g'} \cdot \phi_{i,g'}$$

for a χ-matrix, or

$$FD_{i,g} = W_r \chi_c^{g} \sum_{g'=1}^{G} (\nu \cdot \sigma_f)_c^{g'} \cdot \phi_{i,g'}$$

for a χ-vector.

For an adjoint problem, the fission density is

$$FD_{i,g} = W_r (\nu \cdot \sigma_f)_c^g \sum_{g'=1}^{G} \chi_c^{g'g} \phi_{i,g'}$$

for a χ-matrix, or

$$FD_{i,g} = W_r (\nu \cdot \sigma_f)_c^g \sum_{g'=1}^{G} \chi_c^{g'} \phi_{i,g'}$$

for a χ-vector.

In both cases above, W_r is the balance integration weighting factor for the region r of which the interval i is a part. Additionally, for a source calculation, the fission density is multiplied by the μ factor, $\mu = 1./k_{fixed}$, the scaling factor for the problem.

The fission source integral by interval and group is then easily computed as

$$FS_{i,g} = FD_{i,g} \cdot V_i \quad ,$$

where V_i is the mesh interval volume per unit height, from function VOL.

In FISREG, the fission source integral by region by region and group is computed simply as

$$FS_{r,g} = \sum_{i \varepsilon r} FS_{i,g} \quad ,$$

and the average fission source as

$$\overline{FS}_{r,g} = \frac{FS_{r,g}}{V_r} \quad ,$$

where V_r is the physical region volume. These two quantities are printed by calling TWODPR.

5. Subprograms called by this subroutine

FISCAL calls VOL (entry in IVOL)
BALW (entry in IBAL)

FISREG calls IGET (entry in POINTR)
TWODPR

6. Subprograms calling this subroutine

FISCAL is called by NRMCAL
FISREG is called by D1DOUT

7. Error messages generated by this subroutine - None

Subroutine FLIP (FLX, FLXVOL, REGFLX, FISDEN, FISSOR, IMAX, RMAX, GMAX, FSOPT)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
FLXVOL	R*8	RMAX*GMAX	Regional flux integrals by group
REGFLX	R*8	RMAX*GMAX	Average regional flux integrals by group
FISDEN	R*8	IMAX*GMAX	Fission density by interval and group
FISSOR	R*8	IMAX*GMAX	Fission source integrals by interval and group
IMAX	I*4	--	Number of mesh intervals defined
RMAX	I*4	--	Number of regions defined

Name	Type	Dimension	Definition
GMAX	I*4	--	Number of groups defined
FSOPT	I*4	--	Fission source option 0 - do not write FSR.D1 or FSA.D1 1 - write FSR.D1 or FSA.D1

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(51)	NTH	Index of nth-generation flux in array FLX

3. Local Variables

Name	Type	Definition
NGMAX	I*4	GMAX/2
G	I*4	Index on groups, G = 1, ... NGMAX
GSUB	I*4	Decreasing index on groups, GSUB = GMAX - G + 1
SAVE	R*8	Dummy variable used in re-ordering

4. Functions and tasks performed by this subroutine

This subroutine re-orders by groups the quantities to be put out on data sets or printed after an adjoint problem, so that the group ordering corresponds to that which would be expected. This is necessary because calculations of these quantities were performed using data previously re-ordered to make it possible to do an adjoint problem using the same routines as for a real problem. If further calculations are to be performed, the quantities are again re-ordered, so that they can be used in their form before the first re-ordering.

Quantities re-ordered are the nth generation of the flux FLX, the regional flux integral FLXVOL, the regional flux average REGFLX, the fission density FISDEN, and, if desired, the fission source integral FISSOR.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

NRMCAL

7. Error messages generated by this subroutine - None

Subroutine FLXIN (FLUX, IMAX, GMAX, *)

1. Arguments

Name	Type	Dimension	Definition
FLUX	R*8	IMAX*GMAX*3	Flux $\phi_{i,g}$ by interval and group for initial (nth) generation
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100),INT(100)	Single variable container: FLT contains 100 R*8 words, and INT congains 100 I*4 words

Variables used from this common block

Name	Local name if equivalenced	Definition
INT(51)	NTH	Index of the nth generation of the flux in the POINTR array FLX

3. Local Variables

Name	Type	Definition
FAD1	R*8	'FA.D1' for SNIFF call
SAVE	R*8	Dummy location used in reordering flux by group
KEFF	R*8	k_{eff} defined on FA.D1
G	I*4	Index for reordering flux by groups and group index
GA	I*4	Index for reordering flux by group; GA=GMAX-G+1
GMX1	I*4	Number of groups defined on FA.D1
IMX1	I*4	Number of mesh intervals defined on FA.D1
NGMAX	I*4	GAMX/2 for reordering flux by groups
ITER	I*4	Number of iterations at which flux was written on FA.D1
KIND	I*4	Kind of problem from which flux FA.D1 originated: -1 input guess or incomplete problem >0 completed problem
N1	I*4	Data set reference number for FA.D1
I	I*4	Index for mesh intervals

4. Functions and tasks performed by this subroutine

This subroutine reads the adjoint flux data set FA.D1 and replaces the nth generation flux array in the POINTR container if FA.D1 is a user-supplied data set.

It first reads the initial record of FA.D1 and compares the number of energy groups and mesh intervals defined in this data set with those of the current problem. It then tests to see if this is a user-supplied adjoint flux guess (ITER>0), and if so reads the actual flux input. If the flux guess is from a previous run (KIND>0), the flux is reordered by group. If FA.D1 is not a user-supplied adjoint flux guess, return is made to the

calling program and the last real flux is used as an initial guess for the adjoint part of the calculation.

5. Subprograms called by this subroutine

PAGHED
ERRMSG
SNIFF (ARC System module)

6. Subprograms calling this subroutine

DIF1D
D1DSCH

7. Error messages generated by this subroutine

Message: ERROR NUMBER 1 IN SUBROUTINE FLXIN

Significance: The data set FA.D1 could not be found.

Action taken: Execution is terminated.

Message: *****ERROR IN READING FLUX GUESS******
IMAX=iiii IMX1=iiii
GMAX=iiii GMX1=iiii

ERROR NUMBER 2 IN SUBROUTINE FLXIN

Significance: The number of energy groups (GMX1) and/or the number of mesh intervals (IMX1) defined in the data set FA.D1 do not correspond to the values (GMAX,IMAX) for the current problem.

Action taken: Execution is terminated.

Subroutine FVCAL (FLX, FLXVOL, REGVOL, REGMSH, IMAX, RMAX, GMAX,*)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ fof four or three generations

Name	Type	Dimension	Definition
FLXVOL	R*8	RMAX*GMAX	Regional flux integrals by group
REGVOL	R*8	RMAX	Physical region volumes (from function PHYVOL)
REGMSH	I*2	IMAX	Region-to-mesh correspondence
IMAX	I*4	--	Number of mesh intervals defined
RMAX	I*4	--	Number of regions defined
GMAX	I*4	--	Number of groups defined
*	--	--	Error return (not used)

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT951)	NTH	Index of nth-generation flux in array FL

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine first sets up the physical region volume REGVOL, defined as it is in the function PHYVOL. The routine then computes the regional flux integrals by group, FLXVOL:

$$\Phi_{r,g} = W_r \sum_{i \in r} \phi_{i,g}^{(n)} \cdot V_i \quad ,$$

where W_r is the integration weight, from function BALW, and V_i is the volume per unit height of mesh interval i, from function VOL.

5. Subprograms called by this subroutine

PHYVOL (entry in IPHVOL)
BALW (entry in IBAL)
VOL (entry in IVOL)

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine GETBND (BNDC, JBNDC, NBCDEF)

Entry Points

Entry VERBND (IMESH, NGRP, DIFCOE, C1, C2, C3, C4, ITYPE)

1. Arguments

Name	Type	Dimension	Definition
BNDC	I*4	8*NBCDEF	Name used to reference floating-point portion (last four numbers of each eight) of boundary condition array
JBNDC	I*4	8*NBCDEF	Name used to reference integer portion (first four numbers of each eight) of boundary condition array
NBCDEF	I*4	--	Number of non-continuity boundary conditions defined
IMESH	I*4	--	Mesh line number (negative for left side of line, positive for right side)
NGRP	I*4	--	Group number
DIFCOE	R*8	--	Diffusion coefficient corresponding to IMESH and NGRP (from DC)
C1	R*8	--	First boundary condition coefficient
C2	R*8	--	Second boundary condition coefficient
C3	R*8	--	Third boundary condition coefficient
C4	R*8	--	Fourth boundary condition coefficient

Name	Type	Dimension	Definition
ITYPE	I*4	--	Boundary condition type 0 - continuity of flux and current 1 - $C_1\phi' + C_2\phi = C_3$ where C_1 = BNDC(5, NBC) C_2 = BNDC(6, NBC) C_3 = BNDC(7, NBC) 2 - $C_1\phi' + C_2\phi = C_3$ where C_1 = BNDC(5, NBC) *DIFCOE C_2 = BNDC(6, NBC) C_3 = BNDC(7, NBC) *DIFCOE 3 - $\phi_{i+1} = C_1\phi_i + C_2\phi_i'$ $\phi_{i+1}' = C_3\phi_i + C_4\phi_i'$ where C_1 = BNDC(5, NBC) C_2 = BNDC(6, NBC) C_3 = BNDC(7, NBC) C_4 = BNDC(8, NBC)

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

GETBND is the initialization entry; VERBND performs the specified function.

This subroutine checks through the mesh lines for which boundary conditions have been defined, comparing JBNDC(1, NBC), where NBC = 1,..., NBCDEF, to IMESH. If a match is found, the group numbers between which the condition applies, JBNDC(3, NBC) and JBNDC(4, NBC), are checked. If JBNDC(3, NBC) $\leq$ NGRP $\leq$ JBNDC(4, NBC) the arguments returned are defined as follows:

```
ITYPE = JBNDC(2, NBC)
C1    = BNDC(5, NBC)
C2    = BNDC(6, NBC)
C3    = BNDC(7, NBC)
C4    = BNDC(8, NBC)
```

Further, if ITYPE = 2,

C1 = C1*D1FCOE

and

C3 = C3*D1FCOE.

If no match is found, ITYPE and all constants (C1, C2, C3, C4) are set to zero.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

```
GETBND is called by CELCAL
VERBND is called by FDCCEL (entry in CELCAL)
                    LKCELL (entry in CELCAL)
                    FLXCEL (entry in CELCAL)
                    DERCEL (entry in CELCAL)
                    CURCEL (entry in CELCAL)
                    LKFACE (entry in CELCAL)
```

7. Error messages generated by this subroutine - None

Subroutine GETCHI (SETCHI, MTRCHI, GMAX, NTAPE)

1. Arguments

Name	Type	Dimension	Definition
SETCHI	R*8	ISCHI*GMAX	Set-wide fission spectrum $\chi^{g'g}$
MTRCHI	I*4	--	Set-wide fission spectrum type (ISCHI) 1-χ-vector GMAX-χ-matrix
GMAX	I*4	--	Number of groups defined
NTAPE	I*4	--	Data set reference number for XS.C.MIN

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine reads the set-wide fission spectrum $\chi^{g'g}$, if one exists, from record type 2 of data set XS.C.MIN.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

RCOMP

7. Error messages generated by this subroutine - None

Subroutine IAREA (X)

Entry Points

Entry AREA (AREAL, AREAR, I)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	IMAX+1	Mesh line locations
AREAL	R*8	--	Area on the left associated with mesh line I
AREAR	R*8	--	Area on the right associated with mesh line I
I	I*4	--	Mesh line index

2. Common Blocks

/GNAME/

Variables used from this common block

Name	Type	Definition	
GEOM	R*8	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'

3. Local Variables

Name	Type	Definition
RSLAB	R*8	'RECTSLAB' for geometry type
CYL	R*8	'CYLINDER' for geometry type
SPH	R*8	'SPHERE' for geometry type
PI	R*8	3.14159265

4. Functions and tasks performed by this subroutine

This subroutine calculates the area per unit height associated with a specified mesh line I on the left (A_{i_L}) and on the right (A_{i_R}). The initialization call is made to IAREA, and calls for the area values are made to AREA.

The areas are computed as follows:

$A_{i_L} = 4\pi X_i^2$
$A_{i_R} = 4\pi X_{i+1}^2$ for a sphere,

$A_{i_L} = 2\pi X_i$
$A_{i_R} = 2\pi X_{i+1}$ for a cylinder,

$A_{i_L} = A_{i_R} = 1.0$ for a rectangular slab,

and $A_{i_L} = A_{i_R} = \pi$ for a cylindrical slab.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

IAREA is called by CELCAL

AREA is called by FDCCEL (entry in CELCAL)
LKCELL (entry in CELCAL)
FLXCEL (entry in CELCAL)
DERCEL (entry in CELCAL)
CURCEL (entry in CELCAL)
LKFACE (entry in CELCAL)

7. Error messages generated by this subroutine - None

Function IBAL (BSQ, REGHHT, REGMSH, RMAX)

Entry Points

Entry BALW (REGNO)

1. Arguments

Name	Type	Dimension	Definition
BSQ	R*8	RMAX*2	Buckling by B^2 region and direction
REGHHT	R*8	RMAX*2	Actual region half-height $(H_L)_r$ by direction

Name	Type	Dimension	Definition
REGMSH	I*2	IMAX	Region-to-mesh correspondence
RMAX	I*4	--	Number of regions defined
REGNO	I*4	--	Region number index

2. Common Blocks

/GNAME/

Variables used from this common block

Name	Type	Definition	
GEOM	R*8	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'

3. Local Variables

Name	Type	Definition
CSLAB	R*8	'CYSLAB' for geometry type
CYL	R*8	'CYLINDER' for geometry type
SPH	R*8	'SPHERE' for geometry type
PI	R*8	3.14159265
DUMMY	R*8	Dummy argument passed to initialize J_1 function
BUCK	R*8	$B_r = \sqrt{B_r^2}$
WMOD1	R*8	$W_{r,1}$ for rectangular slab
WMOD2	R*8	$W_{r,2}$ for rectangular slab

4. Functions and tasks performed by this function

This function calculates the integration weighting factor W_r for balance calculations for a specified region r, denoted by REGNO. The initialization call is made to IBAL, and calls for the integration weighting factor are made to BALW.

The weighting factor is computed as follows:

$W_r = 1.$ for a sphere or for an infinite region ($B_r^2 = 0$);

$$W_r = \frac{2}{B_r} \sin(B_r(H_L)_r),$$

where

$$B_r = \frac{\pi}{2H_r} = \sqrt{B_r^2},$$ for a finite cylinder;

$$W_r = \frac{2}{B_{r,1}} \sin(B_{r,1}(H_L)_{r,1}),$$

where

$$B_{r,1} = \frac{\pi}{2H_{r,1}} = \sqrt{B_{r,1}^2},$$ for a rectangular slab with buckling in the transverse direction only;

$$W_r = \frac{2}{B_{r,2}} \sin(B_{r,2}(H_L)_{r,2}),$$

where

$$B_{r,2} = \frac{\pi}{2H_{r,2}} = \sqrt{B_{r,2}^2},$$ for a rectangular slab with buckling in the axial direction only;

$$W_r = \frac{4}{B_{r,1}B_{r,2}} \sin(B_{r,1}(H_L)_{r,1}) \sin(B_{r,2}(H_L)_{r,2}),$$

where

$B_{r,1}$ and $B_{r,2}$ are defined as above, for a rectangular slab with buckling in both directions;

and

$$W_r = \frac{(H_L)_r}{B_r} J_1(B_r(H_L)_r),$$

where

$$B_r = \frac{\alpha}{H_r} = \sqrt{B_r^2}$$

and α is the first zero of J_0, $\alpha \simeq 2.40482556$, for a finite cylindrical slab.

5. Subprograms called by this function

IBAL calls IBFJ1

BALW calls EBFJ1 (entry in IBFJ1)
DUMP (entry in POINTR)

6. Subprograms calling this function

IBAL is called by CELCAL

BALW is called by FVCAL
LKCAL
PWRCAL
FISCAL
EXTCAL

7. Error messages generated by this function - None

Function IBFJ1 (X)

Entry Points

Entry EBFJ1 (X)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	--	Number for which the Bessel function $J_1(X)$ is to be computed

2. Common Blocks - None

3. Local Variables - Not Applicable

4. Functions and tasks performed by this function

This function computes the Bessel function $J_1(X)$ for a given X. The initialization call is to IBFJ1(X), where X is a dummy variable; the Bessel function is actually computed by calling EBFJ1(X).

The function was written by K. Hillstrom and is documented in the Applied Mathematics Division program library under the name BFJ1(X). The

only change has been to set up initialization and calculation entries, and therefore the method used will not be discussed here.

The program is used in this module in computing the integration weighting factor W_r for a cylindrical slab.

5. Subprograms called by this function - None

6. Subprograms calling this function

IBFJ1 is called by IBAL

EBFJ1 is called by BALW (entry in IBAL)

7. Error messages generated by this function - None

Subroutine IFSRC (CMPMSH, FSCMP, NUFIS, CHI, FLUX, FISSRC, CHIDIM, FSCMAX, IMAX)

Entry Points

Entry FSRC (FLUX, FISSRC)

1. Arguments

Name	Type	Dimension	Definition
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
NUFIS	R*8	FSCMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi^{g'g}$
FLUX	R*8	IMAX*GMAX	Flux $\phi_{i,g}$ by mesh interval and group (FLX(1,1,NTH) in BLK array)
FISSRC	R*8	IMAX*GMAX	Fission source by mesh interval and group
CHIDIM	I*4	--	First dimension of fission spectrum array CHI

Name	Type	Dimension	Definition
FSCMAX	I*4	--	Number of fissionable compositions defined
IMAX	I*4	--	Number of mesh intervals defined

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
INT(37)	GMAX	Number of groups defined	
INT(39)	MTRCHI	Fission spectrum array type	0-χ-vector 1-χ-matrix

3. Local Variables

Name	Type	Definition
ADJ	R*8	'ADJOINT' for problem type
CMPNO	I*4	Number of composition corresponding to mesh interval I
NFSCMP	I*4	Number of fissionable composition corresponding to mesh interval I
FISLOC	I*4	First subscript of given element in CHI array for χ-matrix
SUM	R*8	Sum of fission source terms dependent on g' (H)

4. Functions and tasks performed by this subroutine

This subroutine computes the fission source by mesh point and group for a k-effective calculation. IFSRC is the initialization entry; FSRC performs the calculation.

The fission source is defined, for a real problem, as

$$\frac{1}{k_{eff}} \sum_{g'=1}^{G} \chi_i^{g'g} (\nu \cdot \sigma_f)^{g'}_{c(i)} \phi_{ig'} V_i \qquad \text{for a } \chi\text{-matrix}$$

and

$$\frac{1}{k_{eff}} \chi_i^{g} \sum_{g'=1}^{G} (\nu \cdot \sigma_f)^{g'}_{c(i)} \phi_{ig'} V_i \qquad \text{for a } \chi\text{-vector.}$$

For an adjoint problem, the fission source is

$$\frac{1}{k_{eff}} (\nu \cdot \sigma_f)^{g}_{c(i)} \sum_{g'=1}^{G} \chi_i^{gg'} \phi_{ig'} V_i \qquad \text{for a } \chi\text{-matrix,}$$

and

$$\frac{1}{k_{eff}} (\nu \cdot \sigma_f)^{g}_{c(i)} \sum_{g'=1}^{G} \chi^{g'} \phi_{ig'} V_i \qquad \text{for a } \chi\text{-vector .}$$

The entire fission source term is not computed in this subroutine; the $\frac{1}{k_{eff}}$ and V_i terms are multiplied in TSRC (entry in ITSRC).

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

IFSRC is called by ISUBS
FSRC is called by SWEEP (entry in ISWEEP)

7. Error messages generated by this subroutine - None

Subroutine INDOT1 (FLX, IMAX, GMAX)

Entry Points

Entry ENDOT1 (N,IRETRN)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations

Name	Type	Dimension	Definition
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
N	I*4	--	Iteration number
IRETRN	I*4	--	Convergence flag returned: -1-error has occured; 0-normal return; 1-problem has converged

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(11)	FLXEPS	Convergence criterion ε_f on maximum average relative flux difference
FLT(12)	LAMEPS	Convergence criterion ε_λ on difference in flux bounds
FLT(14)	A	Interval shift factor for Chebyshev extrapolation
FLT(15)	FLXMON	Flux monitoring ratio (factor for computing FLXNGL)
FLT(72)	FLXCVG	Sum of (n-1)th-generation fluxes over intervals and groups
FLT(73)	FLXNGL	Number less than which fluxes are to be neglected in computation of flux bounds
INT(13)	NTEST	Number of iterations to be done in an extrapolation sequence before testing for payoff
INT(14)	NEXT	Maximum number of iterations to be done in an extrapolation sequence
INT(15)	NUNEXT	Number of unextrapolated iterations to be done following an extrapolation sequence before starting another

Name	Local Name	Definition
INT(16)	P	Degree of Chebyshev polynomial (iteration counter)
INT(51)	NTH	Index of nth-generation flux in array FLX
INT(52)	NM1	Index of (n-1)th-generation flux in array FLX
INT(53)	NM2	Index of (n-2)th-generation flux in array FLX

3. Local Variables

Name	Type	Definition
NDIV	I*4	Number of successive times iterations have diverged ($\sigma \geq 1$)
FLXDIF	R*8	Maximum relative average flux difference, used in testing for convergence $FLXDIF = \max_g (FLXDF1_g)$
LAMDIF	R*8	Difference between flux bounds, $\overline{\lambda}-\underline{\lambda}$, used in testing for convergence
SIGMA	R*8	Estimated rate of convergence of fluxes σ for Chebyshev extrapolation
ENPM1	R*8	Actual reduction ratio $E_{p-1}^{(n)}$ for Chebyshev extrapolation, the ratio of the flux difference FLXDIF after p-1 extrapolated iterations to the flux difference at the beginning of the extrapolation sequence
ALPHA	R*8	First Chebyshev coefficient α
BETA	R*8	Second Chebyshev coefficient β
NUM	R*8	Sum of squares of fluxes $\sum_{g=1}^{G} \sum_{i=1}^{I} (\phi_{i,g}^{(n)})^2$
NUM2	I*4	Sum of current fluxes for update of FLXCVG $\sum_{g=1}^{G} \sum_{i=1}^{I} (\phi_{i,g}^{(n)})$

Name	Type	Definition
LAMUP	R*8	Upper flux bound, $\bar{\lambda}^{(n)} = \max_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right)$
LAMLOW	R*8	Lower flux bound, $\underline{\lambda}^{(n)} = \min_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right)$
SIGBAR	R*8	Ratio of flux differences FLXDIF for nth generation to that for (n-1)th generation
LAMRAT	R*8	Ratio of difference between flux bounds LAMDIF for nth generation to that for (n-1)th generation
FLXDF1	R*8	Average relative flux difference for given group g $\text{FLXDF1} = \frac{G \cdot \sum_{i=1}^{I} \left(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)} \right)}{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n-1)}}$
FLXNTH	R*8	nth-generation flux for given i and g, before extrapolation
FLXNM1	R*8	(n-1)th-generation flux for given i and g
FLXNM2	R*8	(n-2)th-generation flux for given i and g
RATO	R*8	Ratio of fluxes for given i and g, $\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}$, used in computing flux bounds $\bar{\lambda}$ and $\underline{\lambda}$
PM1	R*8	p-1, for computing CPM1
ARG	R*8	Hyperbolic cosine of γ, $\cosh \gamma = \frac{2-\sigma-a}{\sigma-a}$, for Chebyshev extrapolation
CMP1	R*8	Optimal reduction ratio for Chebyshev extrapolation, $C_{p-1} = \cosh[(p-1)\gamma]$
OMEGA	R*8	Intermediate step in computation of new σ

4. Functions and tasks performed by this subroutine

This subroutine ends each outer iteration by monitoring the iteration, performing Chebyshev acceleration if desired, and checking for convergence. INDOT1 is the initialization entry; ENDOT1 performs the calculations.

ENDOT1 first updates P, the Chebyshev acceleration counter and degree of the polynomial when greater than zero, by adding 1 to it. (P is initially set to -NSTART in SOUTER and is reset to -NUNEXT after an extrapolation sequence, and hence when negative it counts the number of iterations between extrapolation sequences.) If the resulting value of P is negative, no acceleration is contemplated for this iteration; if zero, acceleration could begin on the next iteration if otherwise feasible; and if greater than zero, the iteration is part of an acceleration sequence. If acceleration is to be performed on this iteration (P > 0), CHEB is called to compute the Chebyshev coefficients, α and β.

The information necessary for monitoring and for checking for convergence is then updated. The sum of the squares of the fluxes NUM and the sum of the fluxes NUM2 are computed. The relative average flux difference FLXDF1 is computed as

$$\text{FLXDF1} = \frac{G \cdot \left(\sum_{i=1}^{I} \phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)} \right)}{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n-1)}} ,$$

and the maximum relative average flux difference FLXDIF is then $\max_g [(\text{FLXDF1})_g]$. The number of the group in which FLXDIF occurs is preserved as GMMAX.

If $\phi_{i,g}^{(n-1)}$ is greater than FLXNGL (the lower bound on fluxes used in computing the flux bounds LAMUP and LAMLOW), the rate of convergence RATO for an interval I and group G, $\left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right)$, is computed, and the flux bounds $\bar{\lambda}^{(n)} = \max_{i,g} (\text{RATO}_{i,g})$ and $\underline{\lambda}^{(n)} = \min_{i,g} (\text{RATO}_{i,g})$ are tallied. Then, if acceleration is to be performed, the flux $\phi_{i,g}^{(n)}$ is updated using the equation $\bar{\phi}_{i,g}^{(n)} = \phi_{i,g}^{(n-1)} + \alpha \left(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)} \right) + \beta \left(\phi_{i,g}^{(n-1)} - \phi_{i,g}^{(n-2)} \right)$, where α and β are the Chebyshev coefficients. After these tasks have been performed over all intervals and groups, the flux bounds difference LAMDIF is computed

as $\overline{\lambda}^{(n)} - \underline{\lambda}^{(n)}$. The information generated on this pass is then printed, and a new FLXNGL and FLXCVG to be used on the next pass are generated. The subroutine then checks for convergence, comparing FLXDIF to FLXEPS and LAMDIF to LAMEPS. If the problem has converged, the subroutine sets IRETRN to 1 and returns control to the calling program.

If the problem has not converged, two indicators of the rate of convergence, SIGBAR and LAMRAT, are computed. SIGBAR is the ratio of the flux difference FLXDIF computed on this iteration to that computed on the last iteration, and LAMRAT is the analogous ratio of flux bounds differences LAMDIF. For an iteration directly preceeding one in which an extrapolation sequence could begin (P=0), the subroutine checks to see if this is desirable. If the rate of convergence SIGBAR is satisfactory ($\leq .5$), P is set to -1, thus deferring acceleration for one more pass.

If this is not the case, the subroutine checks to see if previous extrapolation has caused the problem to diverge (SIGBAR ≥ 1). If this has happened, P is also set to -1. If this occurs for twelve iterations in succession, the problem is considered to be insoluble; this probably indicates that the input data is incorrect. Execution is terminated.

If, following these tests, the beginning of an extrapolation sequence on the next pass is still desired, and if the estimated rate of convergence SIGMA has previously been set to zero, SIGMA is set equal to SIGBAR. This completes the calculations preliminary to an extrapolation sequence.

For an iteration which was part of an extrapolation sequence, other monitoring is performed.

The error-reduction ratio ENPM1 $\left(E_{p-1}^{(n)}\right)$, consisting of the ratio of the flux difference at this iteration to that at the beginning of the extrapolation sequence, is updated. If the minimum number of iterations NTEST to an extrapolation sequence has been performed, the error-reduction ratio over the sequence, ENPM1, is compared to the estimated ratio without extrapolation, SIGMA to the (p-1) power. If the estimated ratio without extrapolation is the same as or better than the actual ratio, SIGMA is set to zero (so that it can be re-estimated before the next sequence) and the sequence is terminated.

If the actual ratio is better, the problem becomes one of whether to keep the current SIGMA or to estimate a new value. To determine this, the term CPM1, the reciprocal of the optimal reduction ratio, is computed:

$$C_{(p-1)} = \cosh\left[(p-1) \quad \cosh^{-1}\left(\frac{2-\sigma-a}{\sigma-a}\right)\right],$$

where a is the interval shift factor A and σ is SIGMA. The optimal reduction ratio $\frac{1}{CMPI}$ is then compared to the actual reduction ratio, ENPM1; if the actual ratio is better, SIGMA is kept, and if the optimal ratio is better, SIGMA is re-estimated as

$$= .5\left[(\sigma+a) + (\sigma-a)\cosh\left(\frac{\cosh^{-1}\left(E_{p-1}^{(n)} \cdot C_{p-1}\right)}{p-1}\right)\right].$$

In any case where extrapolation can be continued, then, the number of extrapolated iterations P in this sequence is compared to the maximum number NEXT in a sequence. If this number has been reached, ENPM1 is set to zero, P is set to -NUNEXT, and the sequence is ended. In all cases, if SIGMA is greater than .99995, it is reset to that value, and the accleration monitoring values are printed. This concludes the tasks performed by this subroutine.

5. Subprograms called by this subroutine

INDOT1 call - None
ENDOT1 calls CHEB
DACOSH
ERRMSG

6. Subprograms calling this subroutine

INDOT1 is called by ISSUBS
ENDOT1 is called by SOUTER

7. Error messages generated by this subroutine

Message: *****OUTER ITERATIONS HAVE DIVERGED TWELVE TIMES IN SUCCESSION

ERROR NUMBER 1 IN SUBROUTINE ENDOT1

Significance: Following Chebyshev extrapliation, the iterations have diverged on twelve successive passes.

Action taken: IRETRN is set to -1, and execution is terminated.

Subroutine INDOT2 (FLX, IMAX, GMAX)

Entry Points

Entry ENDOT2 (N, IRETRN)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
N	I*4	--	Iteration number
IRETRN	I*4	--	Convergence flag -1-error has occurred 0-normal return 1-problem has converged

2. Common Blocks

/SINGLE/FLT(100),INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(11)	FLXEPS	Convergence criterion ε_f on difference in flux
FLT(12)	LAMEPS	Convergence criterion ε_λ on flux bounds
FLT(15)	FLXMON	Factor for computing FLXNGL
FLT(72)	FLXCVG	Sum of nth-generation fluxes over intervals and groups
FLT(73)	FLXNGL	Number less than which fluxes are neglected in computation of flux bounds

Name	Local Name	Definition
INT(13)	NTEST	Number of iterations to be done in an extrapolation sequence before testing for payoff
INT(15)	NUNEXT	Number of unextrapolated iterations to be done following an extrapolation sequence before starting another
INT(16)	P	Degree of Chebyshev polynomial (iteration counter)
INT(51)	NTH	Index of nth-generation flux in array FLX
INT(52)	NM1	Position of (n-1)th-generation flux in array FLX
INT(53)	NM2	Position of (n-2)th-generation flux in array FLX

3. Local Variables

Name	Type	Definition
GMMAX	I*4	Number of group in which maximum relative average flux difference FLXDIF occurs
NUM	R*8	Sum of squares of fluxes $\sum_{g=1}^{G} \sum_{i=1}^{I} (\phi_{i,g}^{(n)})^2$
NUM2	R*8	Sum of current fluxes for update of FLXCVG $\sum_{g=1}^{G} \sum_{i=1}^{I} (\phi_{i,g}^{(n)})$
LAMUP	R*8	Upper flux bound, $\overline{\lambda} = \max_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right)$
LAMLOW	R*8	Lower flux bound, $\underline{\lambda}^{(n)} = \max_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right)$
SIGBAR	R*8	Ratio of flux difference FLXDIF for nth generation to that for (n-1)th generation
LAMRAT	R*8	Ratio of difference between flux bounds LAMDIF for nth generation to that for (n-1)th generation

Name	Type	Definition
FLXDF1	R*8	Average relative flux difference for given group g $$\mathrm{FLXDF1}_g = \frac{G \cdot \sum_{i=1}^{I} \left\lvert \phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)} \right\rvert}{\sum_{g=1}^{G} \sum_{i=1}^{I} (\phi_{i,g}^{(n-1)})}$$
FLXDIF	R*8	Maximum relative average flux difference, used in testing for convergence $$\mathrm{FLXDIF} = \max_g (\mathrm{FLXDF1}_g)$$
FLXNTH	R*8	nth-generation flux for given i and g
FLXNM1	R*8	(n-1)th-generation flux for given i and g
OMEGA	R*8	Successive over-relaxation parameter ω
RATØ	R*8	Ratio of fluxes for given i and g, $\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}$, used in computing flux bounds $\overline{\lambda}$ and $\underline{\lambda}$
LAMDIF	R*8	Difference between flux bounds $\overline{\lambda}$ and $\underline{\lambda}$, used in testing for convergence
ENPM1	R*8	Actual reduction ratio $E_{p-1}^{(n)}$ for extrapolation, the ratio of the flux difference FLXDIF after p-1 extrapolated iterations to the flux difference at the beginning of the extrapolation sequence

4. Functions and tasks performed by this subroutine

This subroutine ends each outer iteration by monitoring the iteration, performing successive over-relaxation extrapolation if necessary, and checking for convergence. INDOT2 is the initialization entry; ENDOT2 performs the calculations.

ENDOT2 first updates P, the SOR extrapolation counter, by adding 1 to it. (P is initially set to - NSTART in SOUTER and is reset to - NUNEXT after an extrapolation sequence; hence when negative it counts the number of iterations between extrapolation sequences, and when positive counts the

number of extrapolated iterations.) If the resulting value of P is negative, no acceleration is planned for this iteration; if zero, acceleration could begin on the next iteration if otherwise feasible; and if greater than zero, the iteration is part of an acceleration sequence. If extrapolation is to be performed, the extrapolated flux $\bar{\phi}_{i,g}^{(n)}$ is computed as

$$\bar{\phi}_{i,g}^{(n)} = \phi_{i,g}^{(n)} + \omega(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)}).$$

The extrapolated flux is then used as the nth-generation flux $\phi_{i,g}^{(n)}$.

The information necessary for monitoring the iteration and checking for convergence is then updated. The sum of the squares of the fluxes NUM and the sum of the fluxes NUM2 are calculated. The relative average flux difference for each group, $FLXDF1_g$, is computed as

$$FLXDF1_g = \frac{G \cdot \sum_{i=1}^{I} \left(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)}\right)}{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n-1)}}$$

and the maximum relative average flux difference FLXDIF is then $\max_g [FLXDF1_g]$. The number of the group in which FLXDIF occurs is preserved as GMMAX.

If $\phi_{i,g}^{(n-1)}$ is greater than FLXNGL (the lower bound on fluxes used in computing the flux bounds LAMUP and LAMLOW), the rate of convergence RATO for an interval i and group g, $\left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$, is computed, and the flux bounds LAMUP and LAMLOW, $\bar{\lambda}^{(n)} = \max_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$, $\underline{\lambda}^{(n)} = \min_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$, are computed. The flux bounds difference LAMDIF is then calculated as $(\bar{\lambda}^{(n)} - \underline{\lambda}^{(n)})$. The information generated on this pass is printed.

The subroutine then checks for convergence, comparing FLXDIF to FLXEPS and LAMDIF to LAMEPS. If the problem has converged, the flag IRETRN is set to 1 and control is returned to the calling program.

If it has not converged, a new FLXNGL and FLXCVG to be used on the next pass are generated. Two indicators of the rate of convergence, SIGBAR ($\bar{\sigma}$) and LAMRAT, are computed. SIGBAR is the ratio of the flux difference FLXDIF computed on this iteration to that computed on the last iteration, and LAMRAT is the analogous ratio of the flux bounds differences LAMDIF.

For an iteration immediately preceding one in which an extrapolation sequence could begin (P=0), the subroutine updates the extrapolation parameter OMEGA according to the formula

$$\omega = \frac{2}{(1 + \sqrt{1 - \bar{\sigma}})} .$$

For an iteration which is part of an extrapolation sequence, the error-reduction ratio $E_{p-1}^{(n)}$ is computed. This consists of the ratio of the flux difference FLXDIF after p extrapolated iterations to the flux difference at the first iteration in this extrapolation sequence. If p is less than NTEST, the minimum number of iterations to be done in an extrapolation sequence, no test of the effectiveness the acceleration procedure is made. Otherwise, the following test is made:

$$(E_{p-1}^{(n)})^{\frac{1}{p-1}} < (\omega-1).$$

If this is true, extrapolation will be continued. If it is not true, the extrapolation procedure is considered not to be paying; the value of p is set to -NUNEXT to force unextrapolated iterations, and the value of ω is set to 1. If the problem is close to diverging, i.e., if $\bar{\sigma} \geq .995$, unextrapolated iterations will also be forced.

In any case, the extrapolation monitoring values are then printed, and IRETRN is set to zero. This concludes the tasks of the subroutine.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

INDOT2 is called by ISSUBS
ENDOT2 is called by SOUTER

7. Error messages generated by this subroutine - None

Subroutine INDOUT (FLX, IMAX, GMAX)

Entry Points

Entry ENDOUT (N, IRETRN)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
N	I*4	--	Iteration number
IRETRN	I*4	--	Convergence flag -1-error has occurred 0-normal return 1-problem has converged

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(2)	KEFF	k_{eff}
FLT(11)	FLXEPS	Convergence criterion ε_f on sum of absolute value of flux differences
FLT(12)	LAMEPS	Convergence criterion ε_λ on difference in flux bounds
FLT(13)	EPSKEF	Convergence criterion ε_k on difference in k_{eff}
FLT(14)	AOUTER	Interval shift factor for Chebyshev extrapolation
FLT(15)	FLXMON	Flux monitoring ratio (factor for computing FLXNGL)

Name	Local Name	Definition
FLT(72)	FLXCVG	Sum of nth-generation fluxes over intervals and groups
FLT(73)	FLXNGL	Number less than which fluxes are neglected in computation of flux bounds
INT(13)	NTEST	Number of iterations to be done in an extrapolation sequence before testing for payoff
INT(14)	NEXT	Maximum number of extrapolated iterations in an extrapolation sequence
INT(15)	NUNEXT	Number of unextrapolated iterations to be done between extrapolation sequences
INT(16)	P	Degree of Chebyshev polynomial (iteration counter)
INT(51)	NTH	Index of nth-generation flux in array FLX
INT(52)	NM1	Index of (n-1)th-generation flux in array FLX
INT(53)	NM2	Index of (n-2)th-generation flux in array FLX

3. Local Variables

Name	Type	Definition
NDIV	I*4	Number of successive times iterations have diverged ($\sigma \geq 1$)
FLXDIF	R*8	Flux difference used in testing for convergence $$\text{FLXDIF} = \frac{\sum_{g=1}^{G} \sum_{i=1}^{I} \lvert \phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)} \rvert}{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n)}}$$
LAMDIF	R*8	Difference between flux bounds, $\overline{\lambda} - \underline{\lambda}$, used in testing for convergence
SIGMA	R*8	Estimated rate of convergence of fluxes for Chebyshev extrapolation
ENPM1	R*8	Actual reduction ratio $E_{p-1}^{(n)}$ for Chebyshev extrapolation, the ratio of the flux difference after p-1 extrapolated iterations to the flux difference at the beginning of the extrapolation sequence

Name	Type	Definition
ALPHA	R*8	First Chebyshev coefficient α
BETA	R*8	Second Chebyshev coefficient β
NUM	R*8	Sum of squares of fluxes $\sum_{g=1}^{G} \sum_{i=1}^{I} (\phi_{i,g}^{(n)})^2$
DENOM	R*8	Denominator for computing new k_{eff} $k_{eff}^{(n)} = k_{eff}^{(n-1)} \frac{\sum_{g=1}^{G} \sum_{i=1}^{I} \left(\phi_{i,g}^{(n)}\right)^2}{\sum_{g=1}^{G} \sum_{i=1}^{I} \left(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)}\right)}$ $= k_{eff}^{(n-1)} \cdot \frac{NUM}{DENOM}$
LAMUP	R*8	Upper flux bound, $\bar{\lambda}^{(n)} = \max_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right)$
LAMLOW	R*8	Lower flux bound, $\underline{\lambda}^{(n)} = \min_{i,g} \left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}} \right)$
LAMRAT	R*8	Ratio of difference between flux bounds LAMDIF for nth generation to that for (n-1)th generation
SIGBAR	R*8	Ratio of flux differences FLXDIF for nth generation to that for (n-1)th generation
FLXNTH	R*8	nth-generation flux for given i and g before extrapolation
FLXNM1	R*8	(n-1)th-generation flux for given i and g
FLXNM2	R*8	(n-2)th-generation flux for given i and g

Name	Type	Definition
RATO	R*8	Ratio of fluxes for given i and g, $\left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$, used in computing flux bounds $\overline{\lambda}$ and $\underline{\lambda}$
KEFDUM	R*8	Temporary storage for new k_{eff}, $k_{eff}^{(n)}$
KEFDIF	R*8	Difference in k_{eff} at nth generation, $k_{eff}^{(n)} - k_{eff}^{(n-1)}$, the absolute value of which is used in testing for convergence
PM1	R*8	P-1, for computing CPM1
ARG	R*8	Hyperbolic cosine of γ, $\cosh \gamma = \frac{2-\sigma-a}{\sigma-a}$ for Chebyshev extrapolation
CPM1	R*8	Optimal reduction ratio for Chebyshev extrapolation, $C_{p-1} = \cosh [(p-1)\gamma]$

4. Functions and tasks performed by this subroutine

This subroutine ends each outer iteration by monitoring the iteration, performing Chebyshev acceleration if desired, and checking for convergence. INDOUT is the initialization entry; ENDOUT performs the calculations.

ENDOUT first updates P, the Chebyshev acceleration counter and degree of the polynomial when greater than zero, by adding 1 to it. (P is initially set to -NSTART in OUTERS and is reset to -NUNEXT after an extrapolation sequence, and hence when negative it counts the number of iterations between extrapolation sequences.) If the resulting value of P is negative, no acceleration is contemplated for this iteration; if zero, acceleration could begin on the next iteration if otherwise feasible; and if greater than zero, the iteration is part of an acceleration sequence. If acceleration is to be performed on this iteration (P>0), CHEB is called to compute the Chebyshev coefficients, α and β.

The information necessary for monitoring and for checking for convergence is then updated. The sum of the squares of the fluxes NUM, the sum of the fluxes FLXCVG, and the denominator DENOM used in the k_{eff} calculation are computed. If $\phi_{i,g}^{(n-1)}$ is greather than FLXNGL (the lower bound on fluxes used

in computing the flux bounds LAMUP and LAMLOW), the rate of convergence RATO for an interval I and group G,

$$\left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right),$$

is computed, and the flux bounds

$$\overline{\lambda} = \max_{i,g}\left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right) \text{ and } \underline{\lambda} = \min_{i,g}\left(\frac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$$

are tallied. For all fluxes the sum of the flux difference over intervals and groups,

$$\sum_{g=1}^{G}\sum_{l=1}^{I} |(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)})|,$$

is computed. Then, if acceleration is to be performed, the flux $\phi^{(n)}$ is updated using the equation

$$\overline{\phi}_{i,g}^{(n)} = \phi_{i,g}^{(n-1)} + \alpha\,(\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)}) + \beta\,(\phi_{i,g}^{(n-1)} - \phi_{i,g}^{(n-2)}),$$

where α and β are the Chebyshev coefficients. If the flux becomes negative as a result of this operation, it is assumed to be near zero, and the absolute value of the flux is used.

After these tasks have been performed over all intervals and groups, the new k_{eff} is computed from the fluxes before acceleration,

$$k_{eff}^{(n)} = k_{eff}^{(n-1)} \frac{\sum_{g=1}^{G}\sum_{i=1}^{I} (\phi_{i,g}^{(n)})^2}{\sum_{g=1}^{G}\sum_{i=1}^{I} (\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)})} .$$

The flux bounds difference LAMDIF is $\overline{\lambda}^{(n)} - \underline{\lambda}^{(n)}$, and the flux difference FLXDIF becomes

$$\frac{\sum_{g=1}^{G}\sum_{i=1}^{I} |\phi_{i,g}^{(n)} - \phi_{i,g}^{(n-1)}|}{\sum_{g=1}^{G}\sum_{i=1}^{I} \phi_{i,g}^{(n)}} .$$

The information generated on this pass is then printed, and a new FLXNGL to be used on the next pass is generated. The subroutine then checks for convergence, comparing the absolute value of KEFDIF to EPSKEF, FLXDIF to FLXEPS, and LAMDIF to LAMEPS. If the problem has converged, the subroutine sets IRETRN to 1 and returns control to the calling program.

If the problem has not converged, two indicators of the rate of convergence, SIGBAR and LAMRAT, are computed. SIGBAR is the ratio of the flux difference FLXDIF computed on this iteration to that computed on the last iteration, and LAMRAT is the analogous ratio of flux bounds differences LAMDIF. For an iteration directly preceeding one in which an extrapolation sequence could begin (P=0), the subroutine checks to see if this is desirable. If the rates of convergence SIGBAR and LAMDIF are both satisfactory (< .5), P is set to -1, thus deferring acceleration for one more pass.

If this is not the case, the subroutine checks to see if previous extrapolation has caused the problem to diverge (SIGBAR $\geq$ 1). If this has happened, P is also set to -1. If this occurs for twelve iterations in succession, the problem is considered to be insoluble; this probably indicates that the input data is incorrect. Execution is terminated.

If, following these tests, the beginning of an extrapolation sequence on the next pass is still desired, and if the estimated rate of convergence SIGMA has previously been set to zero, SIGMA is set equal to SIGBAR. This completes the calculations preliminary to an extrapolation sequence.

For an iteration which was part of an extrapolation sequence, other monitoring is performed.

The error-reduction ratio ENPM1 ($E_{p-1}^{(n)}$ is updated, consisting of the ratio of the flux difference at this iteration to that at the beginning of the extrapolation sequence. If the minimum number of iterations NTEST to an extrapolation sequence has been performed, the error-reduction ratio over the sequence, ENPM1, is compared to the estimated ratio without extrapolation, SIGMA to the P-1) power. If the estimated ratio without extrapolation is the same as or better than the actual ratio, SIGMA is set to zero (so that it can be re-estimated before the next sequence) and the sequence is terminated.

If the actual ratio is better, the subroutine compares the difference in k_{eff}, KEFDIF, to the convergence criterion again, since satisfaction of this criterion determines for the purposes of the routine that k_{eff} is known well enough to extrapolate successfully. If this criterion is not met, but if the error-reduction ratio ENPM1 over the sequence is still better than

that estimated if extrapolation were stopped at this point (SIGMA to the (P+1) power), the extrapolation sequence is continued unless the upper limit of the number of iterations to a sequence (NEXT) has been reached. If the error-reduction estimate is not better than that estimated to occur without extrapolation, SIGMA is again set to zero and the extrapolation sequence is ended.

If k_{eff} satisfies the input criterion EPSKEF, the problem becomes one of whether to keep the current SIGMA or to estimate a new value. To determine this, the term CPM1, the reciprocal of the optimal reduction ratio, is computed: $C_{(p-1)} = \cosh\left[(p-1) \cdot \cosh^{-1}\left(\frac{2-\sigma-a}{\sigma-a}\right)\right]$, where a is the interval shift factor AOUTER and σ is SIGMA. The optimal reduction ratio $\frac{1}{CMP1}$ is then compared to the actual reduction ratio, ENPM1; if the actual ratio is better, SIGMA is kept, and if the optimal ratio is better, SIGMA is re-estimated as

$$\sigma = .5\left[(\sigma+a) + (\sigma-a)\cosh\left(\frac{\cosh^{-1}\left(E^{(n)}_{p-1} \cdot C_{p-1}\right)}{p-1}\right)\right] .$$

In any case where extrapolation can be continued, then, the number of extrapolated iterations P in this sequence is compared to the maximum number NEXT in a sequence. If this number has been reached, ENPM1 is set to zero, P is set to -NUNEXT, and the sequence is ended. In all cases, if SIGMA is greater than .99995, it is reset to that value, and the acceleration monitoring values are printed. This concludes the tasks performed by this subroutine.

5. Subprograms called by this subroutine

INDOUT calls - None
ENDOUT calls CHEB
ERRMSG
DACOSH

6. Subprograms calling this subroutine

INDOUT is called by ISUBS
ENDOUT is called by OUTERS

7. Error messages generated by this subroutine

Message: *****FLUX = x.xxxxxxxExx FOR I = iiii G = iiii
ABSOLUTE VALUE OF FLUX WILL BE USED

ERROR NUMBER 1 IN SUBROUTINE ENDOUT

Significance: The flux for the specified mesh interval and group has gone negative following Chebyshev acceleration. It is assumed that the flux is very close to zero for this to have occured.

Action taken: Flux for this mesh interval and group is set to the absolute value of the flux.

Message: *****OUTER ITERATIONS HAVE DIVERGED TWELVE TIMES IN SUCCESSION

ERROR NUMBER 2 IN SUBROUTINE ENDOUT

Significance: Following Chebyshev extrapolation, the iterations have diverged on twelve successive passes.

Action taken: IRETRN is set to -1, and execution is terminated.

Subroutine INVERT (R, A, C, F, OMEGA, X, IMAX)

1. Arguments

Name	Type	Dimension	Definition
R	R*8	IMAX	The terms whose reciprocals are the diagonal elements of the lower triangular matrix L of the tri-diagonal matrix A=LU for a given group (R(1,G) from BLK array)
A	R*8	IMAX	Finite difference coefficient at right boundary of cell for given group (A(1,G) from BLK array)
F	R*8	IMAX	Total source by mesh interval for given group (TOTSRC in BLK array)
OMEGA	R*8	IMAX	The solution, $L\omega=F$, of the down sweep in the Choleski inversion of the tri-diagonal matrix A=LU, used in the up sweep $Ux=\omega$.
X	R*8	IMAX*GMAX	Current fluxes for given group (FLX(1,G,LTH) or FLX(1,G,NTH) from BLK array)
IMAX	I*4	--	Number of mesh intervals defined

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
I	I*4	Increasing index on mesh intervals, I=1,...,IMAX
ISUB	I*4	Decreasing index on mesh intervals, ISUB=IMAX-I+1

4. Functions and tasks performed by this subroutine

This subroutine performs a Choleski inversion of a given tri-diagonal matrix A, in order to solve the equation Ax=F for all mesh intervals for a given group.

The elements of the tri-diagonal matrix A are the finite difference coefficients a, c, and e by mesh point for this group. The matrix is divided into an upper triangular matrix U, whose diagonal elements are 1, and whose non-zero off-diagonal elements are the products $a_i r_i$, and a lower triangular matrix L, whose diagonal elements are $\frac{1}{r_i}$, where r_i is formed from the central finite difference coefficient e_i, and whose non-zero off-diagonal elements are $-c_i$, so that A=LU. The three matrices are displayed below.

A forward sweep over all intervals is then made to compute ω, from the relation $L\omega=F$. Here $\omega_1 = r_1 F_1$, and for each succeeding value i=2,...,I, $\omega_i = r_i(F_i + c_i \omega_{i-1})$. A reverse sweep over all intervals is then made to solve the second relation, $Ux=\omega$, for x. Here $x_I = \omega I$ and for each value $i = I-1,\ldots,1$, $x_i = \omega_i + a_i r_i x_{i+1}$.

$$\overset{A}{\begin{bmatrix} e_{i,g} & -a_{i,g} & & & \bigcirc \\ -c_{2,g} & e_{2,g} & \ddots & & \\ & -c_{3,g} & \ddots & \ddots & \\ \bigcirc & & \ddots & \ddots & -a_{I-1,g} \\ & & & -c_{I,g} & e_{I,g} \end{bmatrix}} \overset{x}{\begin{bmatrix} x_{i,g} \\ x_{2,g} \\ \vdots \\ \vdots \\ x_{I,g} \end{bmatrix}} = \overset{F}{\begin{bmatrix} F_{i,g} \\ F_{2,g} \\ \vdots \\ \vdots \\ F_{I,g} \end{bmatrix}}$$

$$\overset{L}{\begin{bmatrix} \frac{1}{r_{1,g}} & & & \\ -e_{2,g} & \frac{1}{r_{2,g}} & & \\ & \ddots & \ddots & \\ & & -e_{I,g} & \frac{1}{r_{I,g}} \end{bmatrix}} \overset{\omega}{\begin{bmatrix} \omega_{1,g} \\ \omega_{2,g} \\ \vdots \\ \omega_{I,g} \end{bmatrix}} = \overset{F}{\begin{bmatrix} F_{1,g} \\ F_{2,g} \\ \vdots \\ F_{I,g} \end{bmatrix}}$$

$$\overset{U}{\begin{bmatrix} 1 & a_i r_i & \ddots & \\ & 1 & & a_{I-1} r_{I-1} \\ & & \ddots & \\ & & & 1 \end{bmatrix}} \overset{x}{\begin{bmatrix} x_{1,g} \\ x_{2,g} \\ \vdots \\ x_{I,g} \end{bmatrix}} = \overset{\omega}{\begin{bmatrix} \omega_{1,g} \\ \omega_{2,g} \\ \vdots \\ \omega_{I,g} \end{bmatrix}}$$

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

   SWEEP (entry in ISWEEP)
   SSWEEP (entry in ISSWEP)

7. Error messages generated by this subroutine - None

Function IPHVOL (REGHHT, REGMSH, RMAX, IMAX)

Entry Points

Entry PHYVOL (REGNO)

1. Arguments

Name	Type	Dimension	Definition
REGHHT	R*8	RMAX*2	Actual region half-height $(H_L)_r$ by direction
REGMSH	I*2	IMAX	Region-to-mesh correspondence

Name	Type	Dimension	Definition
RMAX	I*4	--	Number of regions defined
IMAX	I*4	--	Number of mesh intervals defined
REGNO	I*4	--	Region number index

2. Common Blocks

/GNAME/

Variables used from this common block

Name	Type	Definition
GEOM	R*8	Geometry type 'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'

3. Local Variables

Name	Type	Definition
RSLAB	R*8	'RECTSLAB' for geometry type

4. Functions and tasks performed by this function

This function computes the actual physical volume of a finite region, or the volume per unit height of an infinite region or a region in a sphere.

This function is initialized by calling IPHVOL; the volume routine VOL must also have been initialized through a call to IVOL. (Both can be accompolished by initializing CELCAL). The physical volumes are calculated in PHYVOL.

The volumes are calculated as follows for a given region r:

$$V_r = 2(H_L)_r \sum_{i\varepsilon r} V_i$$ for a cylinder, cylindrical slab, or sphere,

or $$V_r = 4(H_L)_{r,1} \cdot (H_L)_{r,2} \cdot \sum_{i\varepsilon r} V_i$$ for a rectangular slab.

This calculation can be done without checking to see whether the reactor is finite or infinite or spherical because whenever buckling is not defined in a given direction, the actual half-height is set to .5. Hence, the product

$2(H_L)_r$ (or $4(H_L)_{r,1} \cdot (H_L)_{r,2}$ for an infinite or a spherical reactor is simply 1.

5. Subprograms called by this function

IPHVOL calls - None

PHYVOL calls VOL (entry in IVOL)

6. Subprograms calling this function

IPHVOL is called by CELCAL

PHYVOL is called by PGEOM
FVCAL

7. Error messages generated by this function - None

Subroutine ISOURC (CMPMSH, FSCMP, NUFIS, CHI, SCTLIM, SCTLOC, SCTXS, NEWFLX, OLDFLX, FIXSRC, TOTSRC, EXTBC, CMAX, FSCMAX, CHIDIM, SCTDIM, IMAX, GMAX)

Entry Points

Entry SOURCE (NEWFLX, OLDFLX, TOTSRC, G)

1. Arguments

Name	Type	Dimension	Definition
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
NUFIS	R*8	FSCMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi^{g'g}$
SCTLIM	I*2	2*CMAX	Maximum number of groups of down-scattering and of upscattering for each composition
SCTLOC	I*2	CMAX*GMAX	Location of scattering band for given composition and group in array SCTXS

Name	Type	Dimension	Definition
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$
NEWFLX	R*8	IMAX*GMAX	Flux $\phi_{i,g}^{(n)}$ for this generation (FLX(1,1,NTH) in BLK array)
OLDFLX	R*8	IMAX*GMAX	Flux $\phi_{i,g}^{(n-1)}$ for previous generation (FLX(1,1,NM1) in BLK array)
FIXSRC	R*8	IMAX*GMAX	External fixed source $S_{i,g}$, if present
TOTSRC	R*8	IMAX	Total source F_i by mesh interval for given group g
EXTBC	R*8	IMAX*GMAX	Source due to inhomogeneous interface conditions, $B_{i,g}$, if present
CMAX	I*4	--	Number of compositions defined
FSCMAX	I*4	--	Number of fissionable compositions defined
CHIDIM	I*4	--	First dimension of fission spectrum array CHI
SCTDIM	I*4	--	First dimension of scattering array SCTXS
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
G	I*4	--	Group for which calculations are being performed

2. Common Blocks

/SINGLE/FLT(100),INT(100)	Single variable container. FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(6)	MU	$1/k_{fixed}$ factor for source calculation	

Name	Local Name	Definition	
INT(39)	MTRCHI	Fission spectrum array type	0-χ-vector 1-χ-matrix

3. Local Variables

Name	Type	Definition
ADJ	R*8	'ADJOINT' for problem type
CMPNO	I*4	Composition number corresponding to given mesh interval I
SCTSRC	R*8	Scattering source for given mesh interval I
ILIM1	I*4	Lower limit of scattering band for given composition and group
HLIM2	I*4	Upper limit of scattering band for given composition and group
FISSRC	R*8	Fission source for given mesh interval I
NFSCMP	I*4	Fissionable composition index of composition corresponding to given mesh interval I
FISLOC	I*4	Location of specified element in fission-spectrum array CHI for χ-matrix

4. Functions and tasks performed by this subroutine

This subroutine computes the scattering source, the fission source, and then the total source for each mesh interval I for a given group G, for the one-dimensional diffusion theory source calculation.

For each mesh interval, the scattering source (without the volume term) is first computed:

$$\text{SCTSRC} = \sum_{g'=1}^{G} \sigma_{c(i)}^{g'g} \phi_{i,g'}$$

using the current generation flux $\phi_{i,g}^{(n)}$ for downscattering and the previous generation flux $\phi_{i,g}^{(n-1)}$ for upscattering.

The fission source is then computed for each interval, again without the volume term or the μ factor: For a real problem, the portion computed is

$$\text{FISSRC} = \sum_{g'=1}^{G} \chi^{g'g}_{c(i)} \, (\nu\cdot\sigma_f)^{g'}_{c(i)} \, \phi_{i,g'} \qquad \text{for a } \chi\text{-matrix,}$$

or $$\text{FISSRC} = \chi^{g}_{c(i)} \sum_{g'=1}^{G} (\nu\cdot\sigma_f)^{g'}_{c(i)} \, \phi_{i,g'} \qquad \text{for a } \chi\text{-vector.}$$

For an adjoint problem it is

$$\text{FISSRC} = (\nu\cdot\sigma_f)^{g}_{c(i)} \sum_{g'=1}^{G} \chi^{g'g}_{c(i)} \cdot \phi_{i,g'} \qquad \text{for a } \chi\text{-matrix}$$

or $$\text{FISSRC} = (\nu\cdot\sigma_f)^{g}_{c(i)} \sum_{g'-1}^{G} \chi^{g'}_{c(i)} \cdot \phi_{i,g'} \qquad \text{for a } \chi\text{-vector.}$$

The current generation flux is used below the diagonal, the previous generation flux on or above it.

Finally, the total source term TOTSRC(I) is computed, including the scattering source SCTSRC, the fission source FISSRC, the fixed external source FIXSRC(I,G), and the source due to inhomogeneous boundary conditions EXTBC(I,G):

```
TOTSRC(I) = (SCTSRC + MU*FISSRC + FIXSRC(I,G))*VOL(I) + EXTBC(I,G).
```

5. <u>Subprograms called by this subroutine</u>

   ISOURC calls - None
   SOURCE calls VOL (entry in IVOL)

6. <u>Subprograms calling this subroutine</u>

   ISOURC is called by ISSUBS
   SOURCE is called by SSWEEP (entry in ISSWEP)

7. <u>Error messages generated by this subroutine</u> - None

Subroutine ISSUBS (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return for error in POINTR dynamic storage allocation

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage allocation array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(32)	IMAX	Number of mesh intervals defined
INT(35)	CMAX	Number of compositions defined
INT(36)	FSCMAX	Number of fissionable compositions defined
INT(37)	GMAX	Number of groups defined
INT(40)	CHIDIM	First dimension of fission spectrum array CHI
INT(41)	SCTDIM	First dimension of scattering array SCTXS

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy variable used for checking for POINTR errors

4. Functions and tasks performed by this subroutine

This subroutine initializes the four subroutines used within each source iteration by calling ISSWEP, ISOURC, INDOT1, and INDOT2. Hence, within the iterations, the entry points SSWEEP, SOURCE, ENDOT1, and ENDOT2 can be used.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
ISSWEP
ISOURC
INDOT1
INDOT2
IPTERR (entry in POINTR)

6. Subprograms calling this subroutine

SRCCAL

7. Error messages generated by this subroutine - None

Subroutine ISSWEP (FLX, TOTSRC, R, A, C, OMEGA, IMAX, GMAX)

Entry Points

Entry SSWEEP

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux ϕ by mesh interval and group for four or three generations
TOTSRC	R*8	IMAX	Total source for all intervals for given group
R	R*8	IMAX*GMAX	Reciprocals of diagonal elements of lower triangular matrix L of tri-diagonal matrix A=LU for Choleski inversion
A	R*8	IMAX*GMAX	Finite difference coefficient on right of cell for all groups
C	R*8	IMAX*GMAX	Finite difference coefficient on left of cell for all groups
OMEGA	R*8	IMAX	Intermediate results ω of Choleski inversion of tri-diagonal matrix A=LU for given group

Name	Type	Dimension	Definition
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(51)	NTH	Index of nth-generation flux in array FLX
INT(52)	NM1	Index of (n-1)th-generation flux in array FLX

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine performs the mesh sweep for each group. ISSWEP is the initialization entry; SSWEEP performs the operations described. For each group, the subroutine first calls SOURCE to compute the total source TOTSRC for each mesh interval in the given group, and then calls INVERT to perform Choleski inversion of the tri-diagonal matrix A=LU, to solve the equation Ax = F for the flux x, where F is the total source.

5. Subprograms called by this subroutine

ISSWEP calls None
SSWEEP calls SOURCE (entry in ISOURC)
INVERT

6. Subprograms calling this subroutine

ISSWEP is called by ISSUBS
SSWEEP is called by SOUTER

7. Error messages generated by this subroutine - None

Subroutine ISUBS (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return for POINTR dynamic storage allocation error

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(32)	IMAX	Number of mesh intervals defined
INT(35)	CMAX	Number of compositions defined
INT(36)	FSCMAX	Number of fissionable compositions defined
INT(37)	GMAX	Number of groups defined
INT(40)	CHIDIM	First dimension of fission spectrum array CHI
INT(41)	SCTDIM	First dimension of scattering array SCTXS

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy variable used for checking for POINTR errors

4. Functions and tasks performed by this subroutine

This subroutine initializes the four subroutines used within each outer iteration by calling ISWEEP, IFSRC, ITSRC, and INDOUT. Hence, within the iterations, the entry points SWEEP, FSRC, TSRC, and ENDOUT can be used.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
ISWEEP
IFSRC
ITSRC
INDOUT
IPTERR (entry in POINTR)

6. Subprograms calling this subroutine

KEFCAL

7. Error messages generated by this subroutine - None

Subroutine ISWEEP (FLX, FISSRC, TOTSRC, R, A, C, OMEGA, IMAX, GMAX)

Entry Points

Entry SWEEP

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux ϕ by mesh interval and group for four or three generations
FISSRC	R*8	IMAX*GMAX	Fission source by mesh interval and group
TOTSRC	R*8	IMAX	Total source for all intervals for given group
R	R*8	IMAX*GMAX	Reciprocals of diagonal elements of lower triangular matrix L of tri-diagonal matrix A=LU for Choleski inversion
A	R*8	IMAX*GMAX	Finite difference coefficient on right of cell for all groups
C	R*8	IMAX*GMAX	Finite difference coefficient on left of cell for all groups

Name	Type	Dimension	Definition
OMEGA	R*8	IMAX	Intermediate results ω of Choleski inversion of tri-diagonal matrix A=LU for given group
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(8)	SCTEPS	Convergence criterion ε_S on upscatter iterations
INT(3)	LMAX	Maximum number of upscatter iterations
INT(38)	GMAXUP	Number of lowest-numbered group in which upscattering occurs (no upscattering if GMAXUP=GMAX)
INT(51)	LTH	Position of ℓth-generation of flux in array FLX for upscatter iterations (nth-generation otherwise)
INT(52)	NM1	Position of (n-1)th-generation flux in array FLX
INT(54)	LM1	Position of (ℓ-1)th-generation flux in array FLX for upscatter iterations

3. Local Variables

Name	Type	Definition
IDUM	I*4	Dummy variable used in updating flux generation positions for upscatter iterations
SUM1	R*8	Sum of flux differences over all mesh intervals for one group

Name	Type	Definition
SUM	R*8	Maximum average relative flux difference over groups, for testing convergence of upscatter iterations

$$SUM = \max_{G} \left\{ \frac{GMAX^{*} \sum_{I=1}^{IMAX} \phi_{I,G}^{(\ell)} - \phi_{I,G}^{(\ell-1)}}{\sum_{I=1}^{IMAX} \sum_{G=1}^{GMAX} \phi_{I,G}^{(n-1)}} \right\}$$

GMMAX	I*4	Number of group in which maximum average relative flux difference (SUM) occurs

4. Functions and tasks performed by this subroutine

This subroutine performs the mesh sweep for each group, and the upscatter iterations, if any. ISWEEP is the initialization entry; SWEEP performs the operations described.

The routine first calls FSRC to compute the fission source FISSRC for all mesh intervals and groups. Then, for each group, it calls TSRC to compute the total source TOTSRC for each mesh interval in the given group, and then calls INVERT to perform Choleski inversion of the tri-diagonal matrix A=LU, to solve the equation Ax = F for the flux x, where F is the total source.

If the problem does not require upscatter iterations, this completes the subroutine's function. For upscatter iterations, however, the flux generation positions are updated on each pass so that the new flux will be written over the previous upscattering flux; this new flux will also be the nth generation of flux for the problem as a whole at the conclusion of the upscatter iterations.

The upscatter iterations are performed as above; for each group, where the index of groups runs from GMAXUP to GMAX, the subroutine calls first TSRC and then INVERT, and checks for convergence against the upscattering convergence criterion SCTEPS, printing the maximum average relative flux difference and the group in which it occurred. When the problem has converged, or when LMAX-1 upscatter iterations have been done, the subroutine returns control to the calling program.

5. Subprograms called by this subroutine

ISWEEP calls - None
SWEEP calls FSRC (entry in IFSRC)
TSRC (entry in ITSRC)
INVERT
ERRMSG

6. Subprograms calling this subroutine

ISWEEP is called by ISUBS
SWEEP is called by OUTERS

7. Error messages generated by this subroutine

Message: *****UPSCATTERING ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE SWEEP

Significance: The maximum number of upscatter iterations IMAX has been exceeded and the iterations have not converged.

Action taken: Execution is continued as through iterations had converged.

Subroutine ITSRC (CMPMSH, SCTLIM, SCTLOC, SCTXS, NEWFLX, OLDFLX, TOTSRC, FISSRC, CMAX, SCTDIM, IMAX)

Entry Points

Entry TSRC (NEWFLX, OLDFLX, FISSRC, G)

1. Arguments

Name	Type	Dimension	Definition
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence
SCTLIM	I*2	2*CMAX	Maximum number of groups of downscattering and of upscattering for each composition
SCTLOC	I*2	CMAX*GMAX	Location of scattering band in array SCTXS for each composition and group
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$

Name	Type	Dimension	Definition
NEWFLX	R*8	IMAX*GMAX	Current generation flux $\phi_{i,g}$ by mesh interval and group (FLX(1,1,NTH) IN BLK ARRAY)
OLDFLX	R*8	IMAX*GMAX	Previous generation flux $\phi_{i,g}^{(n-1)}$ by mesh interval and group (FLX(1,1,NM1) in BLK array)
TOTSRC	R*8	IMAX	Total source by mesh interval for given group G
FISSRC	R*8	IMAX	Fission source by mesh interval for given group G (FISSRC(1,G) in BLK array)
CMAX	I*4	--	Number of compositions defined
SCTDIM	I*4	--	First dimension of scattering array SCTXS
IMAX	I*4	--	Number of mesh intervals defined
G	I*4	--	Group number for which mesh sweep is being performed

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(2)	KEFF	k_{eff}
INT(37)	GMAX	Number of groups defined

3. Local Variables

Name	Type	Definition
CMPNO	I*4	Number of composition corresponding to mesh interval I
HLIM1	I*4	Lower limit of scattering band for composition CMPNO and group G

Name	Type	Definition
HLIM2	I*4	Upper limit of scattering band for composition CMPNO and group G
GSUB	I*4	First subscript of scattering element in array SCTXS
SUM	R*8	Sum of scattering source over g' (H)

4. Functions and tasks performed by this subroutine

For k-effective calculations, this subroutine computes the scattering source and then the total source, by adding the scattering source and the fission source, for each mesh point I for a given group G. ITSRC is the initialization entry; TSRC performs the calculation. For each mesh interval and group, the scattering source is computed as $\sum_{g'=1}^{G} \sigma^{g'g} \phi_{i,g'}$, using the current generation flux $\phi_{i,g'}^{(n)}$ for downscattering and the previous generation flux $\phi_{i,g'}^{(n-1)}$ for upscattering. To obtain TOTSRC(I) the subroutine then adds the result, SUM, to the portion of the fission source previously calculated in FISSRC(I,G), dividing the fission source term by k_{eff}, and multiplies the resulting quantity by the volume of the mesh interval, VOL(I), thus producing the total source term.

5. Subprograms called by this subroutine

ITSRC calls None
TSRC calls VOL (entry in IVOL)

6. Subprograms calling this subroutine

ITSRC is called by ISUBS
TSRC is called by SWEEP (entry in ISWEEP)

7. Error messages generated by this subroutine - None

Function IVOL (X, DX)

Entry Points

Entry VOL (I)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	IMAX+1	Mesh line locations
DX	R*8	IMAX	Mesh interval lengths
I	I*4	--	Mesh interval index

2. Common Blocks

/GNAME/

Variables used from this common block

Name	Type	Definition
GEOM	R*8	Geometry type 'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'

3. Local Variables

Name	Type	Definition
CSLAB	R*8	'CYLSLAB' for geometry type
CYL	R*8	'CYLINDER' for geometry type
SPH	R*8	'SPHERE' for geometry type
PI	R*8	3.14159265

4. Functions and tasks performed by this function

This function calculates the volume per unit height V_i of a specified mesh interval I according to the reactor geometry type. The initialization call is made to IVOL, and calls for the volume value are made to VOL.

The volume is computed as follows:

$$V_i = \frac{4\pi}{3}\left(X_{i+1}^3 - X_i^3\right) \text{ for a sphere,}$$

$$V_i = \pi\left(X_{i+1}^2 - X_i^2\right) \text{ for a cylinder,}$$

$$V_i = \left(X_{i+1} - X_i\right) = \Delta X_i \text{ for a rectangular slab, and}$$

$$V_i = \pi\left(X_{i+1} - X_i\right) = \pi\Delta X_i \text{ for a cylindrical slab.}$$

5. Subprograms called by this function - None

6. Subprograms calling this function

IVOL is called by CELCAL
VOL is called by FDCCEL (entry in CELCAL)
LKCELL (entry in CELCAL)
FLXCEL (entry in CELCAL)
DERCEL (entry in CELCAL)
CURCEL (entry in CELCAL)
LKFACE (entry in CELCAL)
PGEOM
FDC1D
TSRC (entry in ITSRC)
SOURCE (entry in ISOURC)
FVCAL
PWRCAL

7. Error messages generated by this function - None

Subroutine KEFCAL (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program if error is encountered in any subprogram called

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(19)	MAXBLK	Size of POINTR container array in bulk storage
INT(32)	IMAX	Number of mesh intervals defined
INT(37)	GMAX	Number of groups defined

3. Local Variables

Name	Type	Definition
LAST	I*4	Last location used in POINTR core container array
LASTB	I*4	Last location used in POINTR bulk container array

4. Functions and tasks performed by this subroutine

This subroutine controls the calculation of k-effective for the one-dimensional diffusion theory k-effective and source calculation module. It allocates storage space for the fission source, the total source, and the intermediate results ω of the Choleski inversion, determines and prints the size of the POINTR container arrays required for the calculation section, calls ISUBS to initialize subroutines used for every iteration, and calls OUTERS to control the outer iterations.

5. Subprograms called by this subroutine

REDEF (entry in POINTR)	ILASTB (entry in POINTR)
IGET (entry in POINTR)	ISUBS
ILAST (entry in POINTR)	OUTERS

6. Subprograms calling this subroutine

D1DCAL

7. Error messages generated by this subroutine - None

Subroutine LKCAL (LEAK, CURR, FLX, REGMSH, REGNAM, RBOUND, RMAX, RMAXP1, GMAX, IMAX, *)

1. Arguments

Name	Type	Dimension	Definition
LEAK	R*8	RMAX*GMAX	Leakage contribution to balance by region and group
CURR	R*8	(RMAXP1)* GMAX	Current across region interfaces by group
FLX	R*8	IMAX*GMAX*4 for upscattering problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
REGMSH	I*2	IMAX	Region-to-mesh correspondence
REGNAM	R*8	RMAX	Region names
RBOUND	R*8	RMAXP1*2	Pairs of region names delinating region interfaces
RMAX	I*4	--	Number of regions defined
RMAXP1	I*4	--	RMAX+1
GMAX	I*4	--	Number of groups defined
IMAX	I*4	--	Number of mesh intervals defined
*	--	--	Error return for error in entry LKCELL or CURCEL of CELCAL

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(31)	GEOM	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'
INT(34)	BCMAX	Number of non-continuity boundary conditions defined	
INT(35)	CMAX	Number of compositions defined	
INT(51)	NTH	Index of nth-generation flux in array FLX	

3. Local Variables

Name	Type	Definition	
TITLE(9)	R*8	Title for leakages for TWODPR	
ROWT	R*8	'REGION' for row title for TWODPR	
COLT	R*8	'GROUP' for column title for TWODPR	
TITLE2(9)	R*8	Title for currents for TWODCR	
ADJ	R*8	'ADJOINT' for problem type	
RO	I*4	Cumulative index on region interfaces	
CURLIG	R*8	Current on left of cell, from CURCEL	
CURRIG	R*8	Current on right of cell, from CURCEL	
IERR	I*4	Error flag returned from LKCELL and CURCEL	0-no error 1-error in interface condition definition
BOUND	R*8	'0' for outside region interface label	
IP1	I*4	I+1 for locating region interfaces	

RP1	I*4	Number of region corresponding to interval I+1	
LEAKIG	R*8	Leakage across cell, from LKCELL	
JPONE	I*4	Problem type for TWODPR and TWODCR	0-real 2-adjoint

4. Functions and tasks performed by this subroutine

This subroutine computes the leakage contribution to the balance equation by region and group, including $D \cdot B^2$ terms, and the current across region interfaces by group.

The current is computed in the direction of the increasing X-dimension. At the left boundary, the current is found by calling CURCEL, and the negative of the current on the left of the cell for each group is used. The name RBOUND(1,1) of the "region" on the left is set equal to BOUND, and the name RBOUND(1,2) of the region on the right is set to the name of the region corresponding to the first mesh interval.

For every other interface, the current on the right of the cell I, where the interface lies between cell I and cell I+1, is taken to be the current across the interface. The names used to define the interface are those of the regions associated with I and I+1. At the right boundary, the current is again that on the right of cell IMAX; the names delineating the interface are the name of the region corresponding to interval IMAX on the left, and BOUND on the right.

The leakage by region and group is computed by calling LKCELL, which returns the leakage $L_{i,g}$ from the cell, including diffusion coefficient times buckling (axial and transverse leakage), if present. The balance leakage by region and group is then computed as

$$L_{r,g} = \sum_{i \varepsilon r} W_r \cdot L_{i,g} \quad ,$$

where W_r is the balance integration weighting factor for the region, from function BALW.

The leakage by region and group and the current across region interfaces by group are printed by calling TWODPR and TWODCR, respectively.

5. Subprograms called by this subroutine

CURCEL (entry in CELCAL)	IGET (entry in POINTR)
LKCELL (entry in CELCAL)	TWODPR
BALW (entry in IBAL)	TWODCR

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Main Program for NUC002

1. Arguments - None

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this program

This program is the dummy driver for the ARC System one-dimensional diffusion theory k-effective and source calculation module, NUC002. It calls subroutine DIF1D, which is the actual driver for the module, and FREE, an entry point in POINTR which frees storage space allocated in bulk (LCS) and main storage by the module. It also serves to pass the error flag from DIF1D to the path level.

5. Subprograms called by this program

DIF1D
FREE

6. Programs calling this program

This module is linked to by standard path STP001 and by any other paths which use the ARC System one-dimensional diffusion theory k-effective and source calculation facility. The statement used to link to this module is CALL LINK('NUC002', IERR).

7. Error messages generated by this program

No error messages are actually generated by this program. However, the error flag IERR is passed through this routine to the path from the driver subroutine DIF1D. IERR has a value of 0 if execution of the one-dimensional diffusion theory k-effective calculation or source calculation was successful, and a value of 1 if execution of the module was terminated due to some error condition. The path may test this flag before continuing to execute other modules dependent on the result of this module.

Subroutine NRMCAL (FLX, FLXVOL, REGFLX, REGVOL, FISDEN, FISSOR, FISXS, NUFIS, X, DX, XBAR, REGBRN, CMPREG, FSCMP, IMAX, RMAX, GMAX, FSCMAX)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
FLXVOL	R*8	RMAX*GMAX	Regional flux integrals by group
REGFLX	R*8	RMAX*GMAX	Average regional flux integrals by group
REGVOL	R*8	RMAX	Physical region volumes (from function PHYVOL)
FISDEN	R*8	IMAX*GMAX	Fission density by mesh interval and group
FISSOR	R*8	IMAX*GMAX	Fission source integrals by mesh interval and group
FISXS	R*8	CMAX*GMAX	Fission cross section σ_f
NUFIS	R*8	CMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
X	R*8	IMAX+1	Mesh line locations X
DX	R*8	IMAX	Mesh interval lengths ΔX

Name	Type	Dimension	Definition
XBAR	R*8	IMAX	Centers of mesh intervals, $\overline{X}_i = X_i + .5\ \Delta X_i$
REGBRN	R*8	RMAX*GMAX	Regional flux integrals, power-normalized for burnup
CMPREG	I*2	RMAX	Composition-to-region correspondence
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
IMAX	I*4	--	Number of mesh intervals defined
RMAX	I*4	--	Number of regions defined
GMAX	I*4	--	Number of groups defined
FSCMAX	I*4	--	Number of fissionable compositions defined

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(2)	KEFF	k_{eff} for k-effective calculation	
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
FLT(6)	MU	$\frac{1}{k_{fixed}}$ factor for source calculation	
FLT(71)	POWIN	Power factor P_0 input for burnup normalization	

Name	Local Name	Definition	
INT(7)	KOUT	Output flag	< 2-no printed output 2-balance output 3-balance plus power output
INT(9)	FSOPT	Fission source option	0-don't write FSR.D1 or FSA.D1 1-write FSR.D1 or FSA.D1
INT(35)	CMAX	Number of compositions defined	
INT(40)	CHIDIM	First dimension of fission spectrum array CHI	
INT(51)	NTH	Index of nth-generation flux in array FLX	
INT(57)	ITER	Number of iteration at which fluxes are output on data sets	

3. Local Variables

Name	Type	Definition
TITLE(9)	R*8	k-effective calculation flux title for TWODFL
TITLES(9)	R*8	Source calculation flux title for TWODFL
TITLE2(9)	R*8	k-effective calculation regional flux integrals title for TWODPR
TITL2S(9)	R*8	Source calculation regional flux integrals title for TWODPR
TITLE3(9)	R*8	Regional flux averages for burnup title for TWODPR
TITLE4(9)	R*8	Fission source density title for TWODFL
TITLE5(9)	R*8	Fission source integrals title for TWODFL
TITLE6(9)	R*8	k-effective calculation regional flux averages title for TWODPR
TITL6S(9)	R*8	Source calculation regional flux averages title for TWODPR

Name	Type	Definition
ROWT2	R*8	'REGION' for row title for TWODPR
COLT2	R*8	'GROUP' for column title for TWODPR
FRD1	R*8	'FR.D1' for SNIFF call
FAD1	R*8	'FA.D1' for SNIFF call
FRPN	R*8	'FR.PN' for SNIFF call
FSRD1	R*8	'FSR.D1' for SNIFF call
FSAD1	R*8	'FSA.D1' for SNIFF call
ADJ	R*8	'ADJOINT' for problem type
SORC	R*8	'SOURCE' for calculation type
PTOTL1	R*8	Normalization factor for k-effective calculation
JPONE	I*4	Problem type for TWODPR 0-real 2-adjoint
KIND	I*4	Problem type for flux output 1-real 3-adjoint
FIXK	R*8	k_{eff} for k-effective calculation $k_{fixed} = 1./\mu$ for source calculation
NFRPN	I*4	Data set reference number for FR.PN
NFRD1	I*4	Data set reference number for FR.D1
NFSRD1	I*4	Data set reference number for FSR.D1
NFAD1	I*4	Data set reference number for FA.D1
NFSAD1	I*4	Data set reference number for FSA.D1

4. Functions and tasks performed by this subroutine

This subroutine controls the writing of all flux and fission source data sets output by the one-dimensional diffusion theory k-effective and source calculation module and the one-dimensional diffusion theory search module; the normalization of flux-related terms for a k-effective calculation or search; and printing of the quantities output, if desired.

For a source calculation, the regional flux integrals by groups already exist in FLXVOL. The regional flux averages REGFLX are computed as

$$\overline{\Phi}_{r,g} = \frac{\Phi_{r,g}}{V_r} ,$$

where $\Phi_{r,g}$ is FLXVOL(R,G) and V_r is the physical volume of the region, REGVOL(R). For a real problem for which a power factor for burnup P_0 has been input, the regional flux averages by group for burnup REGBRN are computed as $\overline{\Phi}^{P}_{r,g} = \Phi_{r,g} \cdot 3.1 \times 10^{10} \cdot P_0$.

For a k-effective calculation, the calculations are performed similarly, except that the flux-related terms are first normalized to one fission in the reactor. The normalization factor PTOTL1 is computed as

$$\sum_{g=1}^{G} \sum_{r=1}^{R} (\sigma_f)^g_{c(r)} \Phi_{r,g} .$$

The fluxes by interval and group, FLX, are then normalized, so that

$$\phi_{i,g} = \frac{\phi_{i,g}}{PTOTL1} .$$

Then, for a k-effective calculation, the regional flux averages by groups, REGFLX, are computed as

$$\overline{\Phi}_{r,g} = \frac{\Phi_{r,g}}{PTOTL1 \quad V_r} ,$$

and for a real problem for which a power factor for burnup P_0 has been input, the regional flux averages by groups for burnup REGBRN are computed as $\overline{\Phi}^{P}_{r,g} = \overline{\Phi}_{r,g} \cdot 3.1 \times 10^{10} \cdot P_0$. Finally, the regional flux integrals are normalized:

$$\Phi_{r,g} = \frac{\Phi_{r,g}}{PTOTL1} .$$

Following this, for either type of calculation, FISCAL is called to compute the fission density by interval and group FISDEN, and the fission source integral by interval and group FISSOR.

Then the subroutine begins the writing of data sets. For a real problem, the fluxes FLX are written on FR.D1, the regional flux averages for burnup REGBRN, if present, on FR.PN, and the fission source integral FISSOR, if desired, on FSR.D1. For an adjoint problem, FLIP is called to re-order

all quantities computed by group; the fluxes FLX are written on FA.D1, and the fission source integral FISSOR, if desired, is written on FSA.D1. If printing of output is not desired (KOUT < 2), this completes the tasks assigned to this subroutine.

If results are to be printed, TWODFL is called to print the flux; TWODPR is called to print the regional flux integrals, the regional flux averages, and the regional flux averages for burnup, if present; and TWODFL is called again to print the fission density and the fission source integrals, if present. All are printed with appropriate titles, and the number of the data set if the quantity was written on a data set and the normalization factor if the quantity was normalized are printed for each. Finally, for an adjoint problem, FLIP is called again to return all quantities to their original order.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
FISCAL
SNIFF (ARC System module)
FLIP
TWODFL
TWODPR

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine OUTERS (FLX, IMAX, GMAX, *)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux ϕ by mesh point and group, for four or three generations

Name	Type	Dimension	Definition
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
*	--	--	Error return for error in ENDOUT

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(1)	PROB	Problem type 'REAL' 'ADJOINT'
FLT(2)	KEFF	k_{eff}
INT(11)	NMAX	Maximum number of outer iterations
INT(12)	NSTART	Number of unextrapolated outer iterations to be done before beginning to extrapolate
INT(16)	P	Degree of Chebyshev polynomial for Chebyshev extrapolation, and iteration counter for extrapolation
INT(51)	NTH	Index in array FLX of nth-generation flux
INT(52)	NM1	Index in array FLX of (n-1)th-generation flux
INT(53)	NM2	Index in array FLX of (n-2)th-generation flux
INT(57)	ITER	Number of iteration at which fluxes are written on data set
INT(58)	NRST	Number of first iteration after which fluxes are to be written on FR.D1 or FA.D1
INT(59)	NRSTI	Increment in NRST after which FR.D1 or FA.D1 is to be written again

3. Local Variables

Name	Type	Definition
FRD1	R*8	'FR.D1' for SNIFF call
FAD1	R*8	'FA.D1' for SNIFF call
REAL	R*8	'REAL' for problem type
ADJ	R*8	'ADJOINT' for problem type
NCHECK	I*4	Number of next iteration at which fluxes are to be written on data set
NC	I*4	Number of next iteration at which new page will be begun
N1	I*4	Data set reference number for FR.D1 or FA.D1
KIND	I*4	Kind of fluxes written
NDUM	I*4	Dummy variable used in updating flux generation indices
IRETRN	I*4	Flag returned from ENDOUT -1-error in ENDOUT 0-normal return 1-problem has converged

4. Functions and tasks performed by this subroutine

This subroutine controls the performance of the outer iterations and the writing of the appropriate flux data set FR.D1 or FA.D1 for the restart option for the one-dimensional diffusion theory k-effective calculation.

It first sets up the page heading for the iteration history printout, and the number NC of the iteration at which the next new page will begin. There are 15 iteration records to a page, in general, except that writing out of the flux data set always causes a new page to be begun.

The number NCHECK of the iteration after which the appropriate flux data set is to be written is also initialized at NRST; it will be incremented thereafter by NRSTI each time the data set is written.

OUTERS then, for each iteration, updates the flux generation positions so that the new generation flux will replace the oldest flux now present, and calls SWEEP to control the mesh sweep by groups and to perform upscatter iterations, if any. It then calls ENDOUT to perform monitoring and acceleration if required and to test for convergence. If the problem has not converged,

the subroutine writes the flux data set if required after the iteration, and continued to iterate. If the problem has converged, or if the maximum number of iterations is exceeded, control is returned to the calling subprogram, and the output section will be the next to be performed.

5. Subprograms called by this subroutine

PAGHED
SNIFF (ARC System module)
SWEEP (entry in ISWEEP)
ENDOUT (entry in INDOUT)
ERRMSG

6. Subprograms calling this subroutine

KEFCAL

7. Error messages generated by this subroutine

Message: *****OUTER ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE OUTERS

Significance: The maximum number of outer iterations, NMAX, has been exceeded, and the problem has not converged.

Action taken: Execution of the remainder of the program, consisting of the output section, is performed as if the problem had converged.

Subroutine PAGHED

1. Arguments - None

2. Common Blocks

/HEADER/

Variables used from this common block

Name	Type	Definition
TITLE(10)	R*8	Page title (10A8)
NPAGE	I*4	Page number

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine updates the page number and prints the page heading and number from common block HEADER at the top of a page. The page heading must have been initialized and the page number set to zero before the first call to this subroutine.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

PGEOM, PCOMP, RCDEP, PRTBC, RVEL, RSORC, OUTERS, SOUTER, PWRCAL, SUMCAL, TWODPR, TWODFL, TWODCR, FLXIN	(when executing module NUC002)
SRCHIN, PGEOM, PCOMP, RCDEP, PRTBC, RVEL, OUTERS, CHANGE, PWRCAL, SUMCAL, TWODPR, TWODFL, TWODCR, FLXIN	(when executing module NUC004)

7. Error messages generated by this subroutine - None

Subroutine PCKSCT (SCTLIM, SCTLOC, SCTDUM, SCTXS, CMAX, DUMDIM, SCTDIM, GMAX, *)

1. Arguments

Name	Type	Dimension	Definition
SCTLIM	I*2	2*CMAX	Maximum number of groups of downscattering and maximum number of groups of upscattering for each composition
SCTLOC	I*2	CMAX*GMAX	Location of scattering band for each composition and group in SCTXS
SCTDUM	R*8	CMAX*GMAX *GMAX	Dummy scattering array into which all values were read from XS.C.MIN
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$ after packing
CMAX	I*4	--	Number of compositions defined
DUMDIM	I*4	--	First dimension of scattering array SCTDUM (CMAX*GMAX)
SCTDIM	I*4	--	First dimension of scattering array SCTXS
GMAX	I*4	--	Number of groups defined
*	--	--	Error return for error in POINTR dynamic storage allocation

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
K	I*4	Index on groups (K = 1,...,GMAX)
G	I*4	Decreasing group index (G=GMAX-K+1)
DUMMY	R*8	Dummy variable used for checking for POINTR errors
ML	I*4	Lowest-numbered group from which downscattering into group G can occur for given C, according to number of downscattering groups in G

Name	Type	Definition
HLIM1	I*4	Lower limit of group number from which downscattering into group G can occur for given C (max(1,ML))
MU	I*4	Highest-numbered group from which upscattering into given group G can occur for given C, according to number of upscattering groups in G
HLIM2	I*4	Upper limit of group number from which upscattering into group G can occur for given C (min(GMAX,MU))
H	I*4	Inner index on groups g' (H = 1,...,GMAX)
GSUB1	I*4	Position of desired scattering element $\sigma_c^{g'g}$ in array SCTXS
GSUB2	I*4	Position of desired scattering element $\sigma_c^{g'g}$ in array SCTDUM

4. Functions and tasks performed by this subroutine

This subroutine packs the scattering data from the SCTDUM array (CMAX*GMAX*GMAX), into which the data was read from data set XS.C.MIN, into the SCTXS array (GMAX* $\sum_{C=1}^{CMAX}$ (SCTLIM(1,C)+SCTLIM(2,C)+1)), where SCTLIM(1,C)+SCTLIM(2,C)+1 is the maximum band width associated with a given composition C.

At the beginning of this subroutine, the first CMAX locations of the array SCTLOC contain the locations of the last position assigned in SCTXS to the preceding composition; in other words, SCTLOC(CMPNO,1)+1 is the location of the first position in SCTXS assigned to composition CMPNO.

By the conclusion of the program, SCTLOC will contain a reference to the first element of the band of groups H from which scattering occurs into a given group G for a given composition CMPNO. Hence, the array SCTXS is "packed" from G=GMAX to G=1 in descending order by groups, so that the first CMAX locations of SCTLOC, used in the packing procedure, will not be destroyed until the final pass.

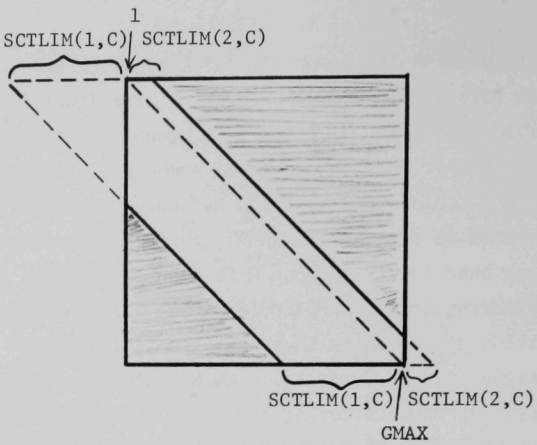
The packing is accomplished as follows: for a given composition CMPNO and group G, the number ML of the lowest-numbered group from which downscattering could occur is found by subtracting the maximum number of downscattering

groups associated with the composition (SCTLIM(1,CMPNO)) from the group number G. Since this number could be negative, the lower limit HLIM1 of the actual possible scattering band for this group G is then computed as max(1,ML). Similarly, the number MU of the highest-numbered group from which upscattering into group G could occur is found by adding SCTLIM(2,CMPNO) to G; since this number could be greater than GMAX, the upper limit HLIM2 of the possible scattering band for this group G is then computed as min(GMAX,MU).

The scattering location SCTLOC(CMPNO,G) associated with the band for that composition and group is then computed as the sum of the first location in SCTXS assigned to the composition CMPNO, and the difference between the number of possible downscattering groups in the composition and the group number G: SCTLOC(CMPNO,G)=SCTLOC(CMPNO,1)+SCTLIM(1,CMPNO)-G. This number may be negative; it corresponds to -ML plus the location of the scattering elements associated with the composition.

Finally, using the limits set up from H=HLIM1,...,HLIM2, the elements of the scattering band associated with CMPNO and G are packed. The scattering element σ^{HG}_{CMPNO} is fetched from SCTDUM(GSUB2,G), where GSUB2 is simply (CMPNO-1)*GMAX+H, and stored in SCTXS(GSUB1,G), where GSUB1 is SCTLOC(CMPNO,G)+H.

Although a full band width SCTLIM(1,CMPNO)+SCTLIM(2,CMPNO)+1 is allocated for use by each group G in the composition, then, only the elements actually present for that group G are actually stored or fetched. The other locations are not used. This can be seen in the diagram; for a given composition CMPNO, the array SCTDUM looks like the GMAX*GMAX square in the diagram, where the white area represents the maximum band width associated with the composition. Space for all of the elements in the parallelogram (in the form of a rectangle which is GMAX long and the composition band width wide) is allocated in the array SCTXS, but only the elements on the parallelogram which fall within the larger square actually exist and are referenced for a given group G.



5. Subprograms called by this subroutine

IPTERR (entry in POINTR)

6. Subprograms calling this subroutine

RCOMP

7. Error messages generated by this subroutine - None

Subroutine PCOMP (DC, REM, FSCMP, FISXS, NUFIS, CHI, SCTLIM, SCTLOC, SCTXS, CMPNAM, CMAX, FSCMAX, CHIDIM, SCTDIM, GMAX, IERR, *)

1. Arguments

Name	Type	Dimension	Definition
DC	R*8	CMAX*GMAX	Diffusion coefficients D_c^g
REM	R*8	CMAX*GMAX	Removal cross section σ_R
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX

Name	Type	Dimension	Definition
FISXS	R*8	FSCMAX*GMAX	Fission cross section σ_f
NUFIS	R*8	FSCMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi_c^{g'g}$
SCTLIM	I*2	2*CMAX	Table of scattering band widths by composition
SCTLOC	I*2	CMAX*GMAX	Table of scattering band locations by composition and group in SCTXS
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$
CMPNAM	R*8	CMAX	Composition names
CMAX	I*4	--	Number of compositions defined
FSCMAX	I*4	--	Number of fissionable compositions defined
CHIDIM	I*4	--	First dimension of fission spectrum array CHI
SCTDIM	I*4	--	First dimension of scattering cross section array SCTXS
GMAX	I*4	--	Number of groups defined
IERR	I*4	--	Error flag (input from search module) 0-no error ≠0-a given cross section has been made negative during concentration search
*	--	--	Error return, used if error flag is greater than zero or for error in POINTR dynamic storage allocation

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
INT(39)	MTRCHI	Fission spectrum array type	0-χ-vector 1-χ-matrix

3. Local Variables

Name	Type	Definition
AB(6)	R*8	Cross section data names corresponding to IERR flag values
DUMMY	R*8	Dummy argument for checking for POINTR errors
CLIM1	I*4	Lower limit of location of fission spectrum values $\chi_c^{g'g}$ for given fissionable composition c and group g for χ-matrix in array CHI
SLIM1	I*4	Lower limit of location of scattering band $\sigma_c^{g'g}$ for given composition c and group g in array SCTXS
SLIM2	I*4	Upper limit of location of scattering band $\sigma_c^{g'g}$ for given composition c and group g in array SCTXS

4. Functions and tasks performed by this subroutine

This subroutine prints cross section data by macroscopic composition, as read from data set XS.C.MIN and modified for use by one-dimensional and two-dimensional diffusion modules.

Data printed for a nonfissionable composition is simply the composition name and, for each group, the group number and diffusion coefficient and removal cross section, followed by a table containing the scattering cross section.

For a fissionable composition, the composition name is followed by the group number, diffusion coefficient, removal cross section, nu times fission cross section, fission spectrum cross section if present as a vector, and fission cross section. If the fission spectrum is a matrix, it is printed

in a separate table, and in either case the scattering cross section is printed in the last table for the composition.

The IERR flag must be set to zero before calling this subroutine from any subprogram other than the search driver, or return may be made to the calling program through the first return provided in the calling sequence.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)
PAGHED
ERRMSG

6. Subprograms calling this subroutine

RCOMP	(when executing module NUC002)	D1DSCH RCOMP	(when executing module NUC004)

7. Error messages generated by this subroutine

Message: *****SEARCH CALCULATED A NEGATIVE aaaaaaaa VALUE
COMPOSITION SEARCH TERMINATED

ERROR NUMBER 1 IN SUBROUTINE PCOMP

where aaaaaaaa is a cross section type:

'DIF COEF', 'REMOVAL', 'FISSION', 'NU FISS', 'CHI MTRX', or 'SCATTER'

Significance: In adding the modifier composition to the original composition on a search pass, the search calculated a negative value or values of some cross section type or types (only the most recently calculated type being indicated).

Action taken: The concentration search is terminated, without output of data sets.

Subroutine PGEOM (X, DX, BSQ, REGHHT, REGMSH, CMPMSH, CMPREG, REGNAM, CMPNAM, EXTDIS, IMAX, RMAX, CMAX, NIL)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	IMAX+1	Mesh line locations
DX	R*8	IMAX	Mesh interval lengths
BSQ	R*8	RMAX*2	Buckling B^2 by region and direction
REGHHT	R*8	RMAX*2	Actual region half-height $(H_L)_r$ by direction
REGMSH	I*2	IMAX	Region-to-mesh correspondence
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence
CMPREG	I*2	RMAX	Composition-to-region correspondence
REGNAM	R*8	RMAX	Region names
CMPNAM	R*8	CMAX	Composition names
ESTDIS	R*8	RMAX*2	Extrapolation distance $(H_r-(H_L)_r)$ by region and direction
IMAX	I*4	--	Number of mesh intervals defined
RMAX	I*4	--	Number of regions defined
CMAX	I*4	--	Number of compositions defined
NIL	I*4	--	Number of finite directions defined

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
INT(49)	NGEOM	Geometry type number	1-cylindrical slab 2-rectangular slab 3-cylinder 4-sphere
INT(88)	NFIN	Number of finite geometry type (0 for infinite reactor)	

3. Local Variables

Name	Type	Definition
TYPE(2,4)	R*8	Four geometry type titles, each 2A8
VLM	R*8	Volume of given interval (from function VOL)
AVOL	R*8	Physical volume of given region (from function PHYVOL)
LINLIM	I*4	Number of pairs of lines of mesh interval data to be printed to a page (22 for the first page, 24 thereafter)
NLINES	I*4	Number of pairs of lines of mesh interval data printed on current page

4. Functions and tasks performed by this subroutine

This subroutine prints geometry data, as read from data set GEOM and modified for use by the one-dimensional diffusion modules.

It first prints the name of the geometry type and, in the case of all geometries but spherical, prints whether the reactor is finite or infinite.

Second, a table of mesh-dependent data is printed, consisting of mesh line locations (X), mesh interval numbers, mesh interval lengths (DX), mesh interval volumes per unit height (function VOL), and the names of the region (REGNAM) and the composition (CMPNAM) corresponding to the interval. This data is printed for the first twenty-two intervals on the first page, and for groups of up to twenty-four intervals on succeeding pages.

Finally, a table of region-dependent data is printed, consisting for a finite reactor of the region name (REGNAM), the corresponding composition

name (CMPNAM), the buckling B^2 (BSQ) in the region, the actual half-height $(H_L)_r$ (REGHHT) of the region, and the extrapolation distance $(H_r-(H_L)_r)$ (EXTDIS) of the region, all in both directions, if both are present, followed by the physical volume (function PHYVOL) of the region. For an infinite or a spherical reactor, this table contains simply the region name, the composition name, and the physical volume per unit height (function PHYVOL). This table will appear on the same page as the mesh interval data if there is room; otherwise, it will appear on a separate page.

5. Subprograms called by this subroutine

PAGHED
VOL (entry in IVOL)
PHYVOL (entry in IPHVOL)

6. Subprograms calling this subroutine

RGEOM (when executing module NUC002)

DIDSCH (when RGEOM executing module NUC004)

7. Error messages generated by this subroutine - None

Subroutine PRTBC (BNDC, JBNDC, X, NBCDEF, *)

1. Arguments

Name	Type	Dimension	Definition
BNDC	R*4	8*NBCDEF	Name used to reference second four numbers (R*4) of each eight in array BNDC
JBNDC	I*4	8*NBCDEF	Name used to reference first four numbers (I*4) of each eight in array BNDC
X	R*8	IMAX+1	Mesh line locations
NBCDEF	I*4	--	Number of non-continuity boundary conditions defined
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition	
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
INT(7)	KOUT	Output flag	0-don't print boundary condition data > 0-print boundary condition data

3. Local Variables

Name	Type	Definition
SORC	R*8	'SOURCE' for calculation type
IP	I*4	Flag indicating non-homogeneous boundary conditions are defined
IIX	I*4	Absolute value of number of mesh line at which boundary condition is defined |JBNDC(1, NBC)|

4. Functions and tasks performed by this subroutine

This subroutine first checks to see if any boundary conditions defined are inhomogeneous. If any are found to be, it makes the IP flag non-zero.

The subroutine next prints the boundary condition data, if that is desired, and then checks the calculation type and the IP flag. If any inhomogeneous boundary conditions appear, and the calculation is not a source calculation, an error is flagged.

5. Subprograms called by this subroutine

ERRMSG
PAGHED

6. Subprograms calling this subroutine

RBNDC

7. Error messages generated by this subroutine

Message: *****CALCULATION TYPE IS aaaaaaaa
BOUNDARY CONDITIONS INDICATE CALCULATION SHOULD BE INHOMOGENEOUS

ERROR NUMBER 1 IN SUBROUTINE PRTBC

where aaaaaaaa is the calculation type in PROBT

Significance: One or more inhomogeneous boundary or interface conditions have been defined, and the calculation type is not source. Either the boundary conditions or the calculation type is in error.

Action taken: Execution is terminated.

Subroutine PWRCAL (FLX, FISXS, REGVOL, PWRDEN, PWRREG, PWRAVG, REGNAM, REGMSH, CMPREG, FSCMP, IMAX, RMAX, GMAX, FSCMAX)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
FISXS	R*8	CMAX*GMAX	Fission cross section σ_f
REGVOL	R*8	RMAX	Physical region volumes V_r (from function PHYVOL)
PWRDEN	R*8	IMAX	Average power by intervals
PWRREG	R*8	RMAX	Power by regions
PWRAVG	R*8	RMAX	Average power by regions
REGNAM	R*8	RMAX	Region names
REGMSH	I*2	IMAX	Region-to-mesh correspondence
CMPREG	I*2	RMAX	Composition-to-region correspondence

Name	Type	Dimension	Definition
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
IMAX	I*4	--	Number of mesh intervals defined
RMAX	I*4	--	Number of regions defined
GMAX	I*4	--	Number of groups defined
FSCMAX	I*4	--	Number of fissionable compositions defined

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition
INT(35)	CMAX	Number of compositions defined
INT(51)	NTH	Index of nth-generation flux, $\phi_{i,g}^{(n)}$, in array FLX

3. Local Variables

Name	Type	Definition
VOLI	R*8	Volume V_i of mesh interval I (from function VOL)
PWRMAX	R*8	Maximum average power by intervals
IPMAX	I*4	Interval in which maximum average power occurs
PWRMIN	R*8	Minimum average power by intervals
IPMIN	I*4	Interval in which minimum average power occurs
PWRRMX	R*8	Maximum power by regions
IPRMAX	I*4	Region in which maximum power occurs
PWRRMN	R*8	Minimum power by regions

Name	Type	Definition
IPRMIN	I*4	Region in which minimum power occurs
PWRTL	R*8	Total power
ILIM	I*4	Number of lines to be printed, at four data items per line
IS	I*4	Index of first item in PWRDEN to be printed on given line
IE	I*4	Index of last item in PWRDEN to be printed on given line

4. Functions and tasks performed by this subroutine

This subroutine computes power-related terms for the one-dimensional diffusion k-effective and source calculation module and the one-dimensional diffusion search module, when KOUT=3.

The average power by intervals PWRDEN is first computed as

$$\overline{P}_i = \sum_{g=1}^{G} W_r \, \phi_{i,g} \, (\sigma_f)^g_{c(i)} \quad ,$$

where $\phi_{i,g}$ is the flux FLX, $(\sigma_f)^g_{c(i)}$ the fission cross section FISXS, and W_r the balance integration weighting factor from function BALW. The maximum average power $\overset{\max}{i} \, [\overline{P}_i]$ and the minimum average power $\overset{\min}{i} \, [\overline{P}_i]$ and the interval in which each occurs are also calculated.

Following this, the power per region PWRREG is computed: $P_r = \sum_{i \varepsilon r} \overline{P}_i \, V_i$, where V_i is the mesh interval volume per unit height. Here again, the maximum power $\overset{\max}{r} \, [P_r]$ and the minimum $\overset{\min}{r} \, [P_r]$ and the number of the region in which each occurs are computed. Then the average power by regions PWRAVG is calculated $\overline{P}_r = \frac{P_r}{V_r}$, where V_r is the physical volume of the region, REGVOL, and the total power is summed: $P_T = \sum_{r=1}^{R} P_r$.

The average power by intervals is printed four data items to a line until the limit IMAX is reached; the other two arrays are printed in columns. Both pairs of maxima and minima computed are also printed, as well as the total power.

5. Subprograms called by this subroutine

VOL (entry in IVOL)
BALW (entry in IBAL)
PAGHED

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine RBLACK (BLKCMP, BLKGRP, N, NBLACK, *)

1. Arguments

Name	Type	Dimension	Definition
BLKCMP	R*8	NBLACK	Names of black compositions, from A.DIF1D
BLKGRP	I*4	NBLACK*2	Lower and upper limits on group numbers for each black composition name, from A.DIF1D
N	I*4	--	Data set reference number for A.DIF1D
NBLACK	I*4	--	Number of black composition cards in A.DIF1D
*	--	--	Error return (not used)

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine reads the names of the black compositions BLKCMP(K), where K=1,···,NBLACK, defined in A.DIF1D, if any are present, and the corresponding group numbers. BLKGRP(K,1) is the number of the initial energy group in which the composition is black, and BLKGRP(K,2) is the number of the final energy group. The composition names are left-justified using the

system routine SQUEZE, but the group number values, if any are zero, are reset when they are used, in subroutine RGROUP.

5. Subprograms called by this subroutine

SQUEZE (ARC System module)

6. Subprograms calling this subroutine

RCDEP

7. Error messages generated by this subroutine - None

Subroutine RBNDC (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program if error is encountered in any subprogram called or in POINTR dynamic storage allocation

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition
INT(32)	IMAX	Number of mesh intervals defined
INT(34)	NBCDEF	Number of non-continuity boundary conditions defined

/BCBLK/

Variables used from this common block

Name	Type	Definition
MESHI	I*4	Mesh line number, negative if boundary condition is on left side of line, positive if on right side
ITYPE	I*4	Boundary condition type
GINT	I*4	Number of initial group to which condition applies
GFIN	I*4	Number of final group to which condition applies
C(4)	R*4	Boundary condition coefficients

3. Local Variables

Name	Type	Definition
BC	R*8	'BC' for SNIFF call
NTAPE	I*4	Data set reference number for BC
ITYPEL	I*4	Boundary condition type on left reactor boundary
ITYPER	I*4	Boundary condition type on right reactor boundary
IBNDC	I*4	Location of BNDC array in BLK
IX	I*4	Location of X array in BLK
DUMMY	R*8	Dummy variable used to check for POINTR errors
NDUM	I*4	Dummy variable used to read unwanted data
CDUM(4)	R*8	Boundary condition coefficients as read from BC (stored in R*4 array C)

4. Functions and tasks performed by this subroutine

This subroutine controls the reading of the boundary and interface condition data from data set BC and the storing of this data in the array BNDC.

The first record of BC gives the boundary condition types at the left and right reactor boundaries, ITYPEL and ITYPER, which are ignored, and the number NBCDEF of boundary and interface conditions defined in the succeeding records of the data set.

The subroutine reads the contents of each of the succeeding records and stores each in the common block BCBLK. For each, it then calls STORBC to store the data in the BNDC array before reading the next record.

This subroutine then calls PRTBC to print the boundary condition data from the BNDC array, if desired, and to check for consistency of boundary condition types with the calculation type.

5. Subprograms called by this subroutine

SNIFF (ARC System module)
ERRMSG
PUTPNT (entry in POINTR)
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
STORBC
PRTBC

6. Subprograms calling this subroutine

D1DIN

7. Error messages generated by this subroutine

Message: *****DATA SET BC NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RBNDC

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****LESS THAN TWO BOUNDARY CONDITIONS DEFINED. NBCDEF=iii

ERROR NUMBER 2 IN SUBROUTINE RBNDC

Significance: Both external boundary conditions, at least, must be defined in data set BC. NBCDEF is the number of boundary conditions defined.

Action taken: Execution is terminated.

Subroutine RCDEP (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT' 'BOTH'
FLT(3)	ALPHA	Inverse reactor time period α	
FLT(4)	SCTEPS	Convergence criterion E_S for upscatter iterations	
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
FLT(6)	MU	$1./k_{fixed}$ factor for source calculation	
FLT(11)	EPOFLX	Outer iteration convergence criterion ε_f on sum of absolute value of flux differences	
FLT(12)	EPSLAM	Outer iteration convergence criterion ε_λ on the difference of flux bounds	
FLT(13)	EPSKEF	Outer iteration convergence criterion ε_k on the difference in k_{eff}	
FLT(14)	AOUTER	Interval shift factor when applying Chebyshev extrapolation	
FLT(15)	FLXMON	Flux monitoring ratio:	

Name	Local name	Definition	
		if $\phi_{i,g}^{(n-1)} < \text{FLXMON}^* \sqrt{\dfrac{\sum_{g=1}^{G} \sum_{i=1}^{I} \phi_{i,g}^{(n-1)}}{I\,G}}$ neglect $\lambda_{i,g}^{(n)} = \left(\dfrac{\phi_{i,g}^{(n)}}{\phi_{i,g}^{(n-1)}}\right)$ in computing $\bar{\lambda}^{(n)}$ and $\underline{\lambda}^{(n)}$	
INT(1)	MKRTHY	Kinetics option	0-no kinetics 1-source due to delayed neutrons 2-delayed data for alpha calculation
INT(2)	KINOPT	Kinetics option type	
INT(3)	LMAX	Maximum number of upscatter iterations	
INT(4)	MAXSIZ	Maximum size of POINTR container array in core	
INT(5)	IPRINT	POINTR debugging flag	0-no print out 1-standard full print out 2-debugging dump 3-full print out (standard plus debugging)
INT(7)	KOUT	Output flag	0-problem description and iteration history print out 1-above plus geometry and composition data print out 2-above plus standard balance output 3-above plus power calculations

Name	Local name	Definition	
INT(9)	FSOPT	Fission source flag	0-no fission source by intervals 1-fission source by intervals printed and written on FSR.D1 or FSA.D1
INT(10)	XTROUT	Outer iteration extrapolation type	0-no extrapolation 1-Chebyshev extrapolation 2-SOR extrapolation (source calculation only)
INT(11)	NMAX	Maximum number of outer iterations	
INT(12)	NSTART	Number of unextrapolated outer iterations to be done before beginning to extrapolate	
INT(13)	NTEST	Number of outer iterations to be done during an extrapolation sequence before testing for extrapolation payoff	
INT(14)	NEXT	Maximum number of extrapolated outer iterations to be done during an extrapolation sequence	
INT(15)	NUNEXT	Number of unextrapolated outer iterations to be done between extrapolation sequences	
INT(17)	SCHOPT	Search output data set flag	0-data sets written 1-data sets not written
INT(19)	MAXBLK	Maximum size of POINTR container array in bulk (LCS) storage	
INT(50)	NBLACK	Number of black composition cards (type 07) in A.DIF1D	
INT(51)	NTH	Index of current flux $\phi_{i,g}^{(n)}$ in array FLX	
INT(52)	NM1	Index of previous generation flux $\phi_{i,g}^{(n-1)}$ in array FLX	
INT(53)	NM2	Index of previous generation flux $\phi_{i,g}^{(n-2)}$ in array FLX	
INT(54)	NDUM (LM1)	Index of previous generation flux for upscatter iterations, $\phi_{i,g}^{(\ell-1)}$ in array FLX	

Name	Local Name	Definition
INT(58)	NRST	Number of first iteration after which fluxes are to be written on FR.D1 or FA.D1
INT(59)	NRSTI	Increment in NRST after which FR.D1 or FA.D1 is to be written again
INT(93)	IPROB	Problem type, from SP.CICN

/HEADER/

Variables used from this common block

Name	Type	Definition
TITLE(10)	R*8	Page title (10A6)
NPAGE	I*4	Page number

3. Local Variables

Name	Type	Definition
ADIF1D	R*8	'A.DIF1D' for SNIFF call
SPCICN	R*8	'SP.CICN' for SNIFF call
REAL	R*8	'REAL' for problem type
ADJ	R*8	'ADJOINT' for problem type
SORC	R*8	'SOURCE' for calculation type
BOTH	R*8	'BOTH' for problem type
CALCK	R*8	'K CALC' for calculation type
AKEFF	R*8	'KEFF' for print of calculation type
AKIN(2)	R*8	'ALPHA' and 'SOURCE' for print of kinetics option type
N	I*4	Data set reference number for A.DIF1D
N1	I*4	Number of type 01 cards in A.DIF1D
N2	I*4	Number of type 02 cards in A.DIF1D
N3	I*4	Number of type 03 cards in A.DIF1D
N4	I*4	Number of type 04 cards in A.DIF1D
N5	I*4	Number of type 05 cards in A.DIF1D

Name	Type	Definition
N6	I*4	Number of type 06 cards in A.DIF1D
NS	I*4	Data set reference number for SP.CICN
DUM	R*8	Dummy variable for reading unwanted data
IDUM	I*4	Dummy variable for reading unwanted data
IPROB	I*4	Calculation type from SP.CICN
FIXEDK	R*8	k_{fixed} from SP.CICN for source calculation

4. Functions and tasks performed by this subroutine

This subroutine reads and prints module-dependent data from data set A.DIF1D and module-independent data from data set SP.CICN, setting up default values where necessary.

It also initializes the data in the common block HEADER used by subroutine PAGHED, using the title from the first 01 card of A.DIF1D; initializes the dynamic storage allocation routine, POINTR, using the indicated array sizes; and defines the flux storage pointers NTH(LTH), NM1, NM2, and LTH(NDUM). If black compositions are present, RCDEP allocates storage for the relevant data and calls RBLACK to read them.

From data set SP.CICN, this subroutine reads MKRTHY, IPROB, ALPHA, and FIXEDK, and sets up the corresponding data for this module: the problem type PROB, the calculation type PROBT, and the MU = 1./FIXEDK factor for source calculations.

5. Subprograms called by this subroutine

```
SNIFF  (ARC System module)
BULK   (entry in POINTR)
POINTR (ARC System module)
PUTPNT (entry in POINTR)
IGET   (entry in POINTR)
RBLACK
PAGHED
ERRMSG
```

6. Subprograms calling this subroutine

```
D1DIN
```

7. Error messages generated by this subroutine

Message: *****DATA SET A.DIF1D NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RCDEP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****DATA SET SP.CICN NOT FOUND

ERROR NUMBER 2 in SUBROUTINE RCDEP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Subroutine RCOMP (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name		Definition
INT(2)	KINOPT	Kinetics option	0-no kinetics option 1-external source due to delayed neutrons

Name	Local name	Definition	
			2-delayed data for alpha calculation
INT(7)	KOUT	Output flag	0-don't print composition data > 0-print composition data
INT(35)	CMAX	Number of compositions defined	
INT(36)	FSCMAX	Number of fissionable compositions defined	
INT(37)	GMAX	Number of groups defined	
INT(38)	GMAXUP	Number of lowest-numbered group into which upscattering occurs	
INT(39)	MTRCHI	Fission spectrum array type	0-χ-vector 1-χ-matrix
INT(40)	CHIDIM	First dimension of fission spectrum array CHI	
INT(41)	SCTDIM	First dimension of scattering array SCTXS	
INT(44)	NFAM	Number of families defined	
INT(50)	NBLACK	Number of black compositions defined	

3. Local Variables

Name	Local name	Definition	
XSCMIN	R*8	'XS.C.MIN' for SNIFF call	
DELAY	R*8	'XS.DELAY' for SNIFF call	
NTAPE	I*4	Data set reference number for XS.C.MIN	
ISCHI	I*4	Set-wide fission spectrum flag, from XS.C.MIN	0-no set-wide χ 1-set-wide χ-vector GMAX-set-wide χ-matrix
NTAPE1	I*4	Data set reference number for XS.DELAY	
IERR	I*4	Flag set to zero before calling PCOMP	

4. Functions and tasks performed by this subroutine

This subroutine controls the reading of composition- and group-dependent data from XS.C.MIN, XS.DELAY, and XS.ISO2 or XS.ISO.

The first thing read is the first record of XS.C.MIN and the end-of-file following that record. If a set-wide fission spectrum $\chi^{g'g}$ is present, the MTRCHI flag is set for a χ-vector or χ-matrix, accordingly, and GETCHI is called to read the set-wide $\chi^{g'g}$ from the second record of XS.C.MIN. If no set-wide χ is present, MTRCHI is set up for a matrix temporarily.

Space for storage of arrays is then allocated, and RDELAY is called to read delayed data from XS.DELAY if the kinetics option is 2. Following this, the CMAX composition blocks of XS.C.MIN are read by calling RGROUP for each composition. If there was no set-wide $\chi^{g'g}$, the MTRCHI flag can now be reset if it is found that χ is actually a vector.

The subroutine then sets up scattering band positions in the scattering array and packs the scattering array by calling DIMSCT and PCKSCT. If a print of composition data is desired (KOUT > 0), PCOMP is then called, and finally RVEL is called to read the neutron speeds from data set XS.ISO2 or XS.ISO if they are needed.

5. Subprograms called by this subroutine

SNIFF (ARC System module)
ERRMSG
PUTPNT (entry in POINTR)
IGET (entry in POINTR)
GETCHI
DUMP (entry in POINTR)
RDELAY
RGROUP
REDEF (entry in POINTR)
WIPOUT (entry in POINTR)
DIMSCT
PCKSCT
PCOMP
RVEL

6. Subprograms calling this subroutine

D1DIN

7. Error messages generated by this subroutine

Message: *****DATA SET XS.C.MIN NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RCOMP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NO END-OF-FILE FOUND FOR FIRST FILE OF XS.C.MIN

ERROR NUMBER 2 IN SUBROUTINE RCOMP

Significance: The end of the first file of this data set was not found, and the second file thus could not be read.

Action taken: Execution is terminated.

Message: *****DATA SET XS.DELAY NOT FOUND

ERROR NUMBER 3 IN SUBROUTINE RCOMP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NUMBER OF FAMILIES DEFINED FOR DELAYED DATA IS ZERO

ERROR NUMBER 4 IN SUBROUTINE RCOMP

Significance: The number of families NFAM defined in XS.C.MIN was zero, and the kinetics option indicates that delayed data is desired for an α-calculation.

Action taken: Execution is terminated.

Subroutine RDELAY (DECCON, DICHI, NFAM, GMAX, NTAPE, *)

1. Arguments

Name	Type	Dimension	Definition
DECCON	R*8	NFAM	Decay constants by family
DICHI	R*8	NFAM*GMAX	Delayed neutron spectrum x_i^g by family and group

Name	Type	Dimension	Definition
NFAM	I*4	--	Number of families defined
GMAX	I*4	--	Number of groups defined
NTAPE	I*4	--	Data set reference number for XS.DELAY
*	--	--	Error return for error in POINTR dynamic storage array allocation

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
DUM	R*8	Dummy variable used to read unwanted data
DUMMY	R*8	Dummy variable used for checking for POINTR errors

4. Functions and tasks performed by this subroutine

This subroutine reads delayed data for kinetics options 1 or 2 from data set XS.DELAY. It skips the first three records and reads the decay constants (DECCON) from the fourth record and the delayed neutron spectrum (DICHI) from the fifth. In practice, the XS.DELAY data set has never been formally defined, and this data would now have to be read from the third and fourth files of XS.ISO2 or XS.ISO if the kinetics option were to be implemented.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)

6. Subprograms calling this subroutine

RCOMP

7. Error messages generated by this subroutine - None

Subroutine RDIST (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
INT(2)	KINOPT	Kinetics option	0-no kinetics 1-source due to delayed neutrons 2-delayed data for alpha calculations
INT(32)	IMAX	Number of mesh intervals defined	
INT(36)	FSCMAX	Number of fissionable compositions defined	
INT(37)	GMAX	Number of groups defined	
INT(38)	GMAXUP	Number of lowest-numbered group from which up scattering occurs	
INT(40)	CHIDIM	First dimension of fission spectrum array CHI	
INT(44)	NFAM	Number of families defined	

3. Local Variables

Name	Type	Definition
SORC	R*8	'SOURCE' for calculation type
DELAY	R*8	'XS.DELAY' for SNIFF call
ADJ	R*8	'ADJOINT' for problem type
LENFLX	I*4	Length of flux array FLX
NTAPE	I*4	Data set reference number for XS.DELAY

4. Functions and tasks performed by this subroutine

This subroutine controls the reading of the flux data from FR.D1 or FA.D1 and of the external source and delayed neutron data for a source problem from ES.D1D and XS.DELAY.

It first defines the size of the array needed to store the flux (four generations for upscattering problems, three otherwise) and calls RFLUX to read the flux data.

It then sets up storage for the decay constants, the delayed neutron spectrum, and the precursor constants, if the kinetics option is 1, and calls RDELAY to read the decay constants and delayed neutron spectrum.

For any source calculation, RDIST then sets up storage for external source data and calls RSORC to read this data from ES.D1D, weight it over groups if desired, and add to it the external source data due to delayed neutrons if the kinetics option has been specified.

5. Subprograms called by this subroutine

PUTPNT (entry in POINTR)
IGET (entry in POINTR)
RFLUX
DUMP (entry in POINTR)
REDEF (entry in POINTR)
SNIFF (ARC System module)
ERRMSG
RDELAY
RSORC
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

D1DIN

7. Error messages generated by this subroutine

Message: *****DATA SET XS.DELAY NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RDIST

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NUMBER OF FAMILIES DEFINED FOR DELAYED DATA IS ZERO

ERROR NUMBER 2 IN SUBROUTINE RDIST

Significance: The number of families NFAM defined in XS.C.MIN was zero, and the kinetics option indicates delayed neutron source data is required.

Action taken: Execution is terminated.

Subroutine REMCAL (RMVL, REM, VEL, FLXVOL, CMPREG, RMAX, CMAX)

1. Arguments

Name	Type	Dimension	Definition
RMVL	R*8	RMAX*GMAX	Removal contribution by region and group
REM	R*8	CMAX*GMAX	Removal cross section σ_R
VEL	R*8	GMAX	Neutron speeds by group
FLXVOL	R*8	RMAX*GMAX	Regional flux integrals by group
CMPREG	I*2	RMAX	Composition-to-region correspondence
RMAX	I*4	--	Number of regions defined
CMAX	I*4	--	Number of compositions defined

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(3)	ALPHA	Inverse reactor time period α	
INT(37)	GMAX	Number of groups defined	

3. Local Variables

Name	Type	Definition	
TITLE(9)	R*8	Title for removal for TWODPR	
ROWT	R*8	'REGION' for row title for TWODPR	
COLT	R*8	'GROUP' for column title for TWODPR	
ADJ	R*8	'ADJOINT' for problem type	
JPONE	I*4	Problem type flag for TWODPR	0-real 2-adjoint

4. Functions and tasks performed by this subroutine

This subroutine computes the removal contribution RMVL to the balance equation, including α/v terms by region and group, and prints the results by calling TWODPR. The removal term is computed as

$$R_{r,g} = ((\sigma_R)^g_{c(r)} + \frac{\alpha}{v_g})\,\Phi_{r,g}$$

where α is the inverse reactor time period, v_g is the neutron speed for group g, $(\sigma_R)^g_{c(r)}$ is the removal cross section for group g for the composition c assigned to region r, and $\Phi_{r,g}$ is the regional flux integral for r and g.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
TWODPR

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine REORDR (DC, REM, FISXS, NUFIS, CHI, SCTLIM, SCTLOC, SCTXS, VEL, FIXSRC, EXTBC, CMAX, FSCMAX, CHIDIM, SCTDIM, IMAX, GMAX, *)

1. Arguments

Name	Type	Dimension	Definition
DC	R*8	CMAX*GMAX	Diffusion coefficients D_c^g
REM	R*8	CMAX*GMAX	Removal cross section σ_R
FISXS	R*8	FSCMAX*GMAX	Fission cross section σ_f
NUFIS	R*8	FSCMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi_c^{g'g}$
SCTLIM	I*2	2*CMAX	Maximum number of groups of down-scattering and of upscattering for each composition
SCTLOC	I*2	CMAX*GMAX	Location in scattering array SCTXS of the scattering band $\sigma_c^{g'g}$ associated with each composition group
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$
VEL	R*8	GMAX	Neutron speeds by group
FIXSRC	R*8	IMAX*GMAX for source calculation 0 for k-effective calculation	Fixed external source

Name	Type	Dimension	Definition
EXTBC	R*8	IMAX*GMAX for source calculation 0 for k-effective calculation	External source due to inhomogeneous interface conditions
CMAX	I*4	--	Number of compositions defined
FSCMAX	I*4	--	Number of fissionable compositions defined
CHIDIM	I*4	--	First dimension of fission spectrum array CHI
SCTDIM	I*4	--	First dimension of scattering array SCTXS
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
*	--	--	Error return for error in POINTR dynamic storage array allocation

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition	
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
INT(39)	MTRCHI	Fission spectrum array type	0-χ-vector 1-χ-matrix

3. Local Variables

Name	Type	Definition
SORC	R*8	'SOURCE' for calculation type
DUMMY	R*8	Dummy variable for checking for POINTR errors

Name	Type	Definition
NGMAX	I*4	GMAX/2 for index on re-ordering
G	I*4	Increasing group index G=1,..., GMAX or G=1,...,NGMAX
GSUB	I*4	Decreasing group index GSUB=GMAX-G+1
SAVE	R*8	Dummy variable used in reordering
GSUB1	I*4	First subscript of fission spectrum matrix element or scattering array element (depending on G)
GSUB2	I*4	First subscript of fission spectrum matrix element or scattering array element (depending on GSUB)
GSUB3	I*4	Second subscript of fission spectrum matrix element or scattering array element whose first subscript is GSUB2
HLIM1	I*4	Scattering band width lower limit
HLIM2	I*4	Scattering band width upper limit

4. Functions and tasks performed by this subroutine

This subroutine re-orders group-dependent input by group for an adjoint problem, so that the same subroutines can be used for both real and adjoint problems.

The arrays are transposed in a straightforward manner, so that G=1 appears to be G=GMAX and vice-versa, for the following quantities: neutron speeds (VEL), diffusion coefficients (DC), removal cross section (REM), fixed external source (FIXSRC) and source due to inhomogeneous interface conditions (EXTBC) if present, fission cross section (FISXS), nu times fission cross section (NUFIS), and the fission spectrum (CHI) if the fission spectrum is in vector form.

The packed scattering array (SCTXS) and the fission spectrum (CHI), if in matrix form, are transposed in a somewhat more complicated manner, across the counter diagonal for each composition, where the counter diagonal elements remain unchanged. Of course, in the case of the scattering array, the entire GMAX*GMAX array is not present, and only that part which is present is transposed. The results for one composition for a χ-matrix, where GMAX=4, are shown below.

All of the operations performed on these arrays are reversible (that is, if the operation is repeated, the array will resume its original form).

$$\begin{bmatrix} x_c^{11} & x_c^{21} & x_c^{31} & x_c^{41} \\ x_c^{12} & x_c^{22} & x_c^{32} & x_c^{42} \\ x_c^{13} & x_c^{23} & x_c^{33} & x_c^{43} \\ x_c^{14} & x_c^{24} & x_c^{34} & x_c^{44} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} x_c^{44} & x_c^{43} & x_c^{42} & x_c^{41} \\ x_c^{34} & x_c^{33} & x_c^{32} & x_c^{31} \\ x_c^{24} & x_c^{23} & x_c^{22} & x_c^{21} \\ x_c^{14} & x_c^{13} & x_c^{12} & x_c^{11} \end{bmatrix}$$

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)
DUMP (entry in POINTR)

6. Subprograms calling this subroutine

FDCGEN

7. Error messages generated by this subroutine - None

Subroutine RFLUX (FLX, IMAX, *)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for upscatter problem IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ by interval and group for four or three generations
IMAX	I*4	--	Number of mesh intervals defined
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition
FLT(1)	PROB	Problem type 'REAL' 'ADJOINT'
FLT(2)	KEFF	k-effective
FLT(15)	FLXMON	Flux monitoring factor used in computing FLXNGL
FLT(72)	FLXCVG	Sum of fluxes $\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}$
FLT(73)	FLXNGL	Number less than which fluxes will be neglected in computing $\bar{\lambda}$ and $\underline{\lambda}$: if $\phi_{I,G}^{(n-1)} < FLXMON * \sqrt{\dfrac{\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(n-1)}}{FLXCVG}}$, neglect $\lambda_{I,G}^{(n)}$ in computing $\bar{\lambda}^{(n)}$ and $\underline{\lambda}^{(n)}$

3. Local Variables

Name	Type	Definition
FRD1	R*8	'FR.D1' for SNIFF call
FAD1	R*8	'FA.D1' for SNIFF call
ADJ	R*8	'ADJOINT' for problem type
DUMMY	R*8	Dummy variable used for checking for POINTR errors
NTAPE	I*4	Data set reference number for FR.D1 or FA.D1
IMX1	I*4	Number of mesh intervals defined on FR.D1 or FA.D1
GMX1	I*4	Number of groups defined on FR.D1 or FA.D1. Also, decreasing subscript on groups while re-ordering fluxes; GMX1=GMAX-G+1

Name	Type	Definition
KIND	I*4	Kind of problem from which fluxes originated on FR.D1 or FA.D1 — -1 - input guess; >0 - guess from problem
ITER	I*4	Number of iteration at which fluxes were written on FR.D1 or FA.D1
NGMAX	I*4	GMAX/2 for re-ordering fluxes from uncompleted adjoint problem
SAVE	R*8	Dummy location used in re-ordering fluxes by group

4. Functions and tasks performed by this subroutine

This subroutine reads the fluxes input to the problem from data set FR.D1 for a real problem or data set FA.D1 for an adjoint problem.

It first reads the initial record of the appropriate flux data set and checks to see that the number of mesh intervals and groups defined in the data set are the same as those defined for the problem in data sets GEOM and XS.C.MIN, and then reads the actual flux input.

RFLUX then computes the initial values for the sum of the fluxes (FLXCVG) and the limit (FLXNGL) below which fluxes are neglected in computing the flux bounds $\bar{\lambda}^{(n)}$ and $\underline{\lambda}^{(n)}$. Finally, for an adjoint problem for which the fluxes are output from a previous run (KIND > 0), the routine re-orders the fluxes by group, since this is not done on output.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)
SNIFF (ARC System module)
ERRMSG

6. Subprograms calling this subroutine

RDIST

7. Error messages generated by this subroutine

Message: *****DATA SET FR.D1 NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RFLUX

or

*****DATA SET FA.D1 NOT FOUND

ERROR NUMBER 2 IN SUBROUTINE RFLUX

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****FLUXES DIMENSIONED IMPROPERLY ON FR.D1 OR FA.D1
IMXI=iiii GMX1=iiii

ERROR NUMBER 3 IN SUBROUTINE RFLUX

Significance: The number of mesh intervals (IMX1) or of groups (GMX1) defined on this data set are not the same as the corresponding numbers defined on data sets GEOM and XS.C.MIN.

Action taken: Execution is terminated.

Subroutine RGEOM (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program if error is encountered in any subprogram called

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local name	Definition	
FLT(31)	GEOM	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'

Name	Local name	Definition	
INT(7)	KOUT	Output flag	0-don't print geometry data > 0-print geometry data
INT(31)	RMAX	Number of regions defined	
INT(32)	IMAX	Number of mesh intervals defined	
INT(34)	BCMAX	Number of non-continuity boundary conditions defined (NBCDEF)	
INT(35)	CMAX	Number of compositions defined	
INT(49)	NGEOM	Geometry type	1-cylindrical slab 2-rectangular slab 3-cylinder 4-sphere
INT(88)	NFIN	Finite geometry type	
INT(89)	NIL	Number of finite directions defined	

3. Local Variables

Name	Type	Definition
GNAME(4)	R*8	Four A8 geometry type names
DSGEOM	R*8	'GEOM' for SNIFF call
NTAPE	I*4	Data set reference number for GEOM
NDUM	I*4	Dummy variable used to read unwanted data
LAREA	I*4	Length of area record (record type 6 of GEOM)
LZONE	I*4	Length of zone record (record type 7 of GEOM)

4. Functions and tasks performed by this subroutine

This subroutine reads geometry data from data set GEOM and modifies it for use by the one-dimensional diffusion modules. Only record type 1 is actually read within RGEOM; record types 2-9 are read in RMESH.

RGEOM modifies the geometry type NGEOM so that its possible values are 1, 2, 3, or 4, and sets up the corresponding alphanumeric name in GEOM. The space needed for geometry-related arrays is then allocated, and RMESH is called to read record types 2-9 and modify the contents appropriately. CELCAL is initialized, and the geometry-related data is printed by PGEOM if the output option flag KOUT is greater than zero.

5. Subprograms called by this subroutine

SNIFF (ARC System module)
PUTPNT (entry in POINTR)
RMESH
IGET (entry in POINTR)
WIPOUT (entry in POINTR)
CELCAL
PGEOM
ERRMSG

6. Subprograms calling this subroutine

D1DIN

7. Error messages generated by this subroutine

Message: *****DATA SET GEOM NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RGEOM

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****GEOMETRY TYPE CANNOT BE IDENTIFIED. NGEOM-iii

ERROR NUMBER 2 IN SUBROUTINE RGEOM

Significance: The geometry type number read from data set GEOM is not 1, 2, or 3. It may be a two-dimensional type (4, 5, 6, 7, or 8), or may be meaningless. The geometry type selected in data set A.NIP (card type 3, cols. 13-18) should be checked.

Action taken: Execution is terminated.

Subroutine RGROUP (DC, REM, FSCMP, NUFIS, CHI, DNSCT, UPSCT, SCTLIM, SCTXS, SETCHI, DECCON, DICHI, DCCHI, FISXS, CMPNAM, BLKCMP, BLKGRP, CMAX, FSCMAX, CHIDIM, NFAM, ISCHI, SCTDIM, NFSCMP, CMPNO, NBLACK, NTAPE, *)

1. Arguments

Name	Type	Dimension	Definition
DC	R*8	CMAX*GMAX	Diffusion coefficients D_c^g
REM	R*8	CMAX*GMAX	Removal cross section σ_R
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
NUFIS	R*8	FSCMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi_c^{g'g}$
DNSCT	I*4	GMAX	Number of groups of downscatter for each group in this composition
UPSCT	I*4	GMAX	Number of groups of upscatter for each group in this composition
SCTLIM	I*2	2*CMAX	Maximum number of groups of downscattering and maximum number of groups of upscattering for each composition
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$ as read from XS.C.MIN(SCTDUM in BLK array)
SETCHI	R*8	ISCHI*GMAX	Set-wide fission spectrum $\chi^{g'g}$, if present
DECCON	R*8	NFAM for KINOPT =1 or 2 0 for KINOPT =0	Decay constants
DICHI	R*8	NFAM*GMAX for KINOPT =1 or 2 0 for KINOPT=0	Delayed neutron spectrum χ_i^g

Name	Type	Dimension	Definition
DCCHI	R*8	NFAM*GMAX for KINOPT =1 or 2 0 for KINOPT =0	Delayed neutron-emitter-yield spectrum $\chi_i^{g'}$
FISXS	R*8	FSCMAX*GMAX	Fission cross section σ_f
CMPNAM	R*8	CMAX	Composition names
BLKCMP	R*8	NBLACK	Names of black compositions
BLKGRP	I*4	NBLACK*2	Initial and final group numbers for each black composition name
CMAX	I*4	--	Number of compositions defined
FSCMAX	I*4	--	Number of fissionable compositions defined
CHIDIM	I*4	--	First dimension of fission spectrum array CHI
NFAM	I*4	--	Number of families defined
ISCHI	I*4	--	Set-wide fission spectrum flag 0 - no set-wide χ 1 - set wide χ-vector GMAX - set-wide χ-matrix
SCTDIM	I*4	--	First dimension of scattering cross section array
NFSCMP	I*4	--	Index of this composition in arrays dimensioned FSCMAX
CMPNO	I*4	--	Number of this composition
NBLACK	I*4	--	Number of black compositions defined
NTAPE	I*4	--	Data set reference number for XS.C.MIN
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container; FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(3)	ALPHA	Inverse reactor time period α	
INT(2)	KINOPT	Kinetics option	0-no kinetics 1-source due to delayed neutrons 2-delayed data for alpha calculation
INT(37)	GMAX	Number of groups defined	
INT(38)	GMAXUP	Number of lowest-numbered group from which up-scattering occurs	
INT(39)	MTRCHI	Fission spectrum array type	0-χ-vector 1-χ-matrix

3. Local Variables

Name	Type	Definition	
DUMMY	R*8	Dummy variable used for checking for POINTR errors	
ICHI	I*4	Fission spectrum flag for this composition	-1-uses set-wide χ 0-non-fissionable 1-uses own χ-vector GMAX-uses own χ-matrix
CLIM1	I*4	Position of first element of fission spectrum for this composition in array CHI	
CLIM2	I*4	Position of last element of fission spectrum for this composition in array CHI	
NB	I*4	Black composition flag	0-this composition not defined as black 1-this composition defined as black for some group or groups

Name	Type	Definition
SCTDN	I*4	Number of groups of downscatter for given group in this composition
SCTUP	I*4	Number of groups of upscatter for given group in this composition
SLIM1	I*4	Position of first element of scattering cross section for given group in dummy array
SLIM2	I*4	Position of last element of scattering cross section for given group in dummy array
NDUM	I*4	Dummy variable for reading unwanted data
DUM	R*8	Dummy variable for reading unwanted data
H	I*4	Index on groups, H=1,...,GMAX
K	I*4	Index on black composition data; decreasing index on groups, K=GMAX-H+1, for re-ordering scattering array for given group G
IH	I*4	Position of Hth element in scattering array for given group G
IK	I*4	Position of Kth element in scattering array for given group G
DUMSCT	R*8	Dummy variable used in flipping positions of scattering array elements for given group
IG	I*4	Position of Gth (self-scattering) element in scattering array for given group G
SUM	R*8	Sum of delayed data over families, to be added to fission spectrum

4. Functions and tasks performed by this subroutine

This subroutine reads a composition block from data set XS.C.MIN for all groups G of a given composition CMPNO, and modifies the data for use by all diffusion modules.

RGROUP first reads the initial record of the composition block, containing the fission spectrum flag ICHI and the number of upscattering and downscattering groups defined for each group, UPSCT and DNSCT. Using the fission spectrum flag, the locations for the fission spectrum elements for this composition are set aside in the array CHI, if any are needed. The MTRCHI flag is set to zero if the fission spectrum is found to be a vector and if, in the absence of a set-wide χ, it was assumed to be a matrix. The scattering array elements for this composition are then all set to zero.

Next, the subroutine checks to see if this composition is black for any groups by comparing this composition name with those defined as black (BLKCMP). If any match is found a flag NB is set and the group limits for that composition are set up in BLKFGP, if either or both is zero. If the first group limit is zero, the limits are set to 1 and GMAX; if the first is non-zero and the second is zero, the second is set to the first, so that the condition applies to only one group. The composition name is compared to all compositions defined as black, even after a match is found, so that group limits will be made uniform for all occurrences of definition of this composition as a black composition. (A composition name might appear more than once if it were defined as black for two or more non-contiguous sets of groups, e.g., for G=1 to 7 and for G=16.)

The subroutine then begins to perform a series of tasks for each group in this composition. It sets up the locations in the scattering array assigned to the band associated with this composition and group (SLIM1, SLIM2), and also the maximum scattering band width which will be associated with the scattering for this composition after the data has been packed, defined by SCTLIM(1,CMPNO)+SCTLIM(2,CMPNO)+1. Here SCTLIM(1,CMPNO) is the maximum number of groups of downscattering, and SCTLIM(2,CMPNO) is the maximum number of groups of upscattering.

The cross section data for this composition and group is then read according to the proper format, depending on whether the composition is fissionable or non-fissionable, on whether the fission spectrum is present as a matrix or vector or the set-wide χ is used, and on whether or not delayed data for an alpha calculation is to be read. If the set-wide fission spectrum is to be used for this composition, the fission spectrum for this composition and group in the array CHI is set equal to the set-wide χ from the array SETCHI.

The diffusion coefficient for this composition and group is then set up using the transport coefficient: $D_c^g = \dfrac{1}{3(\sigma_{TR})_c^g}$, unless $(\sigma_{TR})_c^g = 0$, in

which case D_c^g is set to 1.0 E+15. Then, if this composition was defined as black over any group or groups, the subroutine checks to see if this group is between the group limits. If it is, the diffusion coefficient D_c^g is set to zero.

The subroutine then reverses the order of the elements in the scattering array for this composition and group, since they are read in order from the scattering from highest-number group into g through scattering from the lowest-number group into g and are stored in the opposite order, and then sets the self-scattering term to zero. This completes the list of tasks performed for each group for this composition.

Finally, if delayed data for an alpha calculation is present, the delayed fission spectrum terms are added into the fission spectrum array CHI.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)

6. Subprograms calling this subroutine

RCOMP

7. Error messages generated by this subroutine

Message: *****TRANSPORT CROSS-SECTION IS ZERO DIFFUSION COEFFICIENT IS SET TO 1.0 E+15
FOR COMPOSITION aaaaaaaa, GROUP iii

Significance: A transport cross section of zero for this composition and group was read from data set XS.C.MIN, and hence the diffusion coefficient, $D_c^g = \frac{1}{3(\sigma_{TR})_c^g}$, could not be calculated. The diffusion coefficient is set to 1.0×10^{15}.

Action taken: Processing continues. If this composition is found to be defined as black for this group the diffusion coefficient will be set to zero, as it should be.

Subroutine RMESH (REGMSH, CMPMSH, MSHREG, MSHCMP, CMPREG, CRDUM, X, DX, BSQ, REGHHT, EXTDIS, REGNAM, CMPNAM, LAREA, LZONE, NFIN, NIL, RMAX, CMAX, IMAX, NTAPE, *)

1. Arguments

Name	Type	Dimension	Definition
REGMSH	I*2	IMAX	Region-to-mesh correspondence
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence
MSHREG	I*4	IMAX	Region-to-mesh correspondence as read from GEOM
MSHCMP	I*4	IMAX	Composition-to-mesh correspondence as read from GEOM
CMPREG	I*2	RMAX	Composition-to-region correspondence
CRDUM	I*4	RMAX	Composition-to-region correspondence as read from GEOM
X	R*8	IMAX+1	Mesh line locations
DX	R*8	IMAX	Mesh interval lengths
BSQ	R*8	RMAX*2	Buckling B^2 by region and direction
REGHHT	R*8	RMAX*2	Actual region half-heights $(H_L)_r$ by direction
EXTDIS	R*8	RMAX*2	Extrapolation distance $(H_r-(H_L)_r)$ by region and direction
REGNAM	R*8	RMAX	Region names
CMPNAM	R*8	CMAX	Composition names
LAREA	I*4	--	Length of area record (record type 6 of GEOM)
LZONE	I*4	--	Length of zone record (record type 7 of GEOM)
NFIN	I*4	--	Number of finite geometry type
NIL	I*4	--	Number of finite directions defined

Name	Type	Dimension	Definition
RMAX	I*4	--	Number of regions defined
CMAX	I*4	--	Number of compositions defined
IMAX	I*4	--	Number of mesh intervals defined
NTAPE	I*4	--	Data set reference number for GEOM
*	--	--	Error return for error in POINTR dynamic storage array allocation

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(32)	XL	Left reactor boundary (X(1))
FLT(33)	XR	Right reactor boundary (X(IMAX+1))

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy argument for checking for POINTR errors

4. Functions and tasks performed by this subroutine

This subroutine reads record types 2- 9 of data set GEOM and modifies the data for use by one-dimensional diffusion theory modules.

It first reads mesh line locations from record type 2 into X and sets up XL and XR and the mesh interval lengths DX. It then reads region-to-mesh correspondence (record type 3), composition-to-mesh correspondence (record type 4), and composition-to-region correspondence (record type 5) into the temporary I*4 arrays MSHREG, MSHCMP, and CRDUM and transfers them to the I*2 arrays REGMSH, CMPMSH, and CMPREG for permanent storage.

Record types 6 and 7 (area and zone definitions) are skipped if present.

Finite geometry data, including buckling, actual half-height, and extrapolated half-height, are then read from record type 8 into the arrays BSQ, REGHHT, and EXTDIS. If buckling data for a given region R and direction N is not present, BSQ(R,N) is set to zero, and for any region and direction in which BSQ(R,N) is zero, both the actual and the extrapolated half-heights are set to .5. For all regions and directions, then, the extrapolation distance EXTDIS is set up by subtracting the contents of REGHHT from those of EXTDIS.

Finally, the region and composition names are read from record type 9 into REGNAM and CMPNAM.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)

6. Subprograms calling this subroutine

RGEOM

7. Error messages generated by this subroutine - None

Subroutine RSORC (FIXSRC, DECCON, DICHI, PRECON, NUFIS, CHI, FISSUM, WFLAG, CMPMSH, FSCMP, NFAM, IMAX, FSCMAX, CHIDIM, NTAPE1, *)

1. Arguments

Name	Type	Dimension	Definition
FIXSRC	R*8	IMAX*GMAX	External fixed source
DECCON	R*8	NFAM for KINOPT=1 or 2 0 for KINOPT =0	Decay constants by family
DICHI	R*8	NFAM*GMAX for KINOPT=1 or 2 0 for KINOPT =0	Delayed neutron spectrum χ_i^g

Name	Type	Dimension	Definition
PRECON	R*8	NFAM*IMAX for KINOPT=1 0 for KINOPT =0 or 2	Precursor constants
NUFIS	R*8	FSCMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi_c^{g'g}$
FISSUM	R*8	FSCMAX*GMAX for PROB ='REAL' FSCMAX for PROB='ADJOINT'	Weighting factor for external fixed source
WFLAG	I*4	IMAX	Weighting flags for fixed source by intervals: 0-no weighting; 1-distribute external source over groups
CMPMSH	I*2	IMAX	Composition-to-mesh correspondence
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
NFAM	I*4	--	Number of families defined
IMAX	I*4	--	Number of mesh intervals defined
FSCMAX	I*4	--	Number of fissionable compositions
CHIDIM	I*4	--	First dimension of fission spectrum array CHI
NTAPE1	I*4	--	Data set reference number of XS.DELAY
*	--	--	Error return for error in POINTR dynamic storage array allocation

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'

Name	Local Name	Definition	
INT(2)	KINOPT	Kinetics option	0-no kinetics 1-source due to delayed neutrons 2-delayed data for alpha calculation
INT(35)	CMAX	Number of compositions defined	
INT(37)	GMAX	Number of groups defined	
INT(39)	MTRCHI	Fission spectrum array type	0-χ-vector 1-χ-matrix

3. Local Variables

Name	Type	Definition	
ADJ	R*8	'ADJOINT' for problem type	
ESD1	R*8	'ES.D1D' for SNIFF call	
GMX1	I*4	Number of groups defined on ES.D1D	
IMX1	I*4	Number of mesh intervals defined on ES.D1D	
IPRT	I*4	Message printing flag	0-message has not been printed 1-message has been printed once
DUMMY	R*8	Dummy variable for checking for POINTR errors	
NTAPE	I*4	Data set reference number for ES.D1D	

4. Functions and tasks performed by this subroutine

This subroutine reads the external fixed source data from data set ES.D1D and distributes it over groups using fission spectrum weighting for the mesh intervals for which this option is specified. If no external fixed source data set is present, the external source is set to zero, and the source is assumed to be due to inhomogeneous boundary conditions.

If external fixed source data is present, the subroutine first sets up the array FISSUM, used in calculating the weighting factor which will be used in distributing the external fixed source. For a real problem with a χ-vector this array is not used, since the weighting factor is simply χ_N^G

for all groups G in the specified mesh interval I with corresponding fissionable composition N. For a real problem with a χ-matrix, the weighting factor is FISSUM(N,G), where FISSUM(N,G) $= \sum_{G=1}^{GMAX} \chi_N^{G'G}/GMAX$. For an adjoint problem, the weighting factor used is $\frac{(\nu\cdot\sigma_f)_N^G}{FISSUM(N)}$, where FISSUM(N) $= \sum_{G=1}^{GMAX} (\nu\cdot\sigma_f)_N^G$.

After setting up the contents of the array FISSUM, if needed, the subroutine reads the first record of data set ES.D1D, containing the number of intervals, the number of groups, and the flags WFLAG(I) for each interval I, which determine whether distribution of source data over groups is to be performed in that interval. It then reads the external fixed source data into the array FIXSRC, checks the number of groups and of intervals specified in the data set against those specified in the problem, and performs fission spectrum distribution of the fixed source over groups in the intervals for which it is specified. If distribution is specified for any interval to which a non-fissionable composition is assigned, the fixed source for every group in that interval is set to zero.

If desired, the subroutine then prints external fixed source data in its modified form and returns, unless the kinetics option 1 is specified. In that case, the subroutine reads past the first end-of-file on XS.DELAY and reads the precursor constants into the array PRECON. It then adds the external source due to delayed neutrons into the external source already calculated.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)
SNIFF (ARC System module)
DUMP (entry in POINTR)
ERRMSG
PAGHED

6. Subprograms calling this subroutine

RDIST

7. Error messages generated by this subroutine

Message: *****EXTERNAL FIXED SOURCE DATA DIMENSIONED IMPROPERLY ON ES.D1D

IMX1=iiii GMX1=iiii

ERROR NUMBER 1 IN SUBROUTINE RSORC

Significance: The number of intervals IMX1 and/or the number of groups GMX1 defined on this data set do not agree with those defined for the problem in data sets GEOM and XS.C.MIN.

Action taken: Execution is terminated.

Message: *****NO END-OF-FILE FOUND FOR THE FIRST FILE OF XS.DELAY

ERROR NUMBER 2 IN SUBROUTINE RSORC

Significance: The end-of-file for the first file of XS.DELAY was not found and the precursor constants could not be read for the kinetics option.

Action taken: Execution is terminated.

Message: *****FISSION SPECTRUM WEIGHTING USING A NON-FISSIONABLE COMPOSITION CAUSES THE EXTERNAL SOURCE DATA AFFECTED TO BE SET TO ZERO

INTERVALS AFFECTED ARE

I = iiii
.
.
I = iiii

Significance: As indicated, fission-spectrum weighting has been specified for the indicated interval or intervals, each of which contains a non-fissionable composition.

Action taken: FIXSRC(I,G) is set to zero for G=1, ..., GMAX.

Subroutine RVEL (VEL, GMAX, *)

1. Arguments

Name	Type	Dimension	Definition
VEL	R*8	GMAX	Neutron speeds by group
GMAX	I*4	--	Number of groups defined
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container; FLT contains 100 R*8 words and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(3)	ALPHA	Inverse reactor time period α	
INT(2)	KINOPT	Kinetics option	0-no kinetics 1-source due to delayed neutrons 2-delayed neutrons for alpha calculation
INT(7)	KOUT	Output flag	0-don't print speeds >0-print speeds
INT(93)	IPROB	Problem type	5-α-search
INT(94)	KFLAG	Type of variable prescribed to be searched to	0-k_{eff} 1-α

3. Local Variables

Name	Type	Definition
XSVEL	R*8	'XS.ISO' for SNIFF call
XSVEL2	R*8	'XS.ISO2' for SNIFF call

Name	Type	Definition
DUMMY	R*8	Dummy variable used for checking for POINTR errors
NTAPE	I*4	Data set reference number for XS.ISO or XS.ISO2
NGROUP	I*4	Number of groups defined on XS.ISO or XS.ISO2
NISO	I*4	Number of isotopes defined on XS.ISO or XS.ISO2
ISO	I*4	Dummy variable used to read isotope names from XS.ISO or XS.ISO2
LOC	I*4	Dummy variable used to read number of records in second file to be skipped to read isotope data for given isotope I
NGP1	I*4	NGROUP+1, for use in reading record type 3 of XS.ISO or XS.ISO2
ICHI	I*4	Fission spectrum flag for isotope set
E	R*8	Dummy variable used to read energy boundaries of groups
U	R*8	Dummy variable used to read lethargy boundaries of groups

4. Functions and tasks performed by this subroutine

This subroutine reads neutron speeds by groups, if needed, from record type 3 of data set XS.ISO2, if that data set is present, or from XS.ISO otherwise. If neither data set is present, or if speeds are not needed, the speeds are set to 1. in all groups.

If the speeds are read from XS.ISO or XS.ISO2, and if the output flag KOUT is greater than zero, the speeds and the data set reference number are printed.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)
SNIFF (system routine)
PAGHED
ERPMSG

6. Subprograms calling this subroutine

RCOMP

7. Error messages generated by this subroutine

Message: *****NUMBER OF GROUPS DEFINED IN XS.ISO OR XS.ISO2
DOES NOT AGREE WITH NUMBER DEFINED IN GEOM
DSRN = iii GMAX = iii NGROUP - iii

ERROR NUMBER 1 IN SUBROUTINE RVEL

Significance: As stated, the number of groups defined in the two data sets do not agree. DSRN is the data set reference number associated with XS.ISO or XS.ISO2, GMAX is the number of groups defined in GEOM, and NGROUP is the number defined in XS.ISO or XS.ISO2.

Action taken: Execution is terminated.

Subroutine SCTCAL (SCAT, SCTLIM, SCTLOC, SCTXS, FLXVOL, CMPREG, RMAX, CMAX, SCTDIM)

1. Arguments

Name	Type	Dimension	Definition
SCAT	R*8	RMAX*GMAX	Scattering source by region and group
SCTLIM	I*2	2*CMAX	Maximum number of groups of down-scattering and of upscattering for each composition
SCTLOC	I*2	CMAX*GMAX	Location of scattering band for each composition and group in SCTXS
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$
FLXVOL	R*8	RMAX*GMAX	Regional flux integrals by group
CMPREG	I*2	RMAX	Composition-to-region correspondence
RMAX	I*4	--	Number of regions defined
CMAX	I*4	--	Number of compositions defined
SCTDIM	I*4	--	First dimension of scattering array SCTXS

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
INT(37)	GMAX	Number of groups defined	

3. Local Variables

Name	Type	Definition
TITLE(9)	R*8	Title for scattering for TWODPR
ROWT	R*8	'REGION' for row title for TWODPR
COLT	R*8	'GROUP' for column title for TWODPR
ADJ	R*8	'ADJOINT' for problem type
ML	I*4	Minimum lower limit of scattering band for given composition and group
GPRMLO	I*4	Actual lower limit of scattering band
MU	I*4	Maximum upper limit of scattering band for given composition and group
GPRMHI	I*4	Actual upper limit of scattering band
GPRM	I*4	Index on band width
GSUB	I*4	Location of given element in scattering array SCTXS

4. Functions and tasks performed by this subroutine

This subroutine computes and prints the scattering source contribution SCAT to the balance equation by region and group. The scattering source by region and group is computed as

$$SS_{r,g} = \sum_{g'=1}^{G} \sigma_{c(r)}^{g'g} \cdot \Phi_{r,g'} ,$$

where $\sigma_{c(r)}^{g'g}$ is the scattering cross section for the given group and the composition c corresponding to region r, and $\Phi_{r,g'}$ is the regional flux integral for r and g'.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
TWODPR

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine SOUTER (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return for error in ENDOT1 or ENDOT2

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name		Definition
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'

Name	Local Name	Definition
FLT(6)	MU	$1/k_{fixed}$ factor
INT(10)	XTROUT	Extrapolation type 0-no extrapolation 1-Chebyshev extrapolation 2-SOR extrapolation
INT(11)	NMAX	Maximum number of iterations
INT(12)	NSTART	Number of first iteration at which an extrapolation sequence is to begin
INT(16)	P	Degree of Chebyshev polynomial for Chebyshev extrapolation, and iteration counter for extrapolation
INT(32)	IMAX	Number of mesh intervals defined
INT(37)	GMAX	Number of groups defined
INT(51)	NTH	Index of nth-generation flux in array FLX
INT(52)	NM1	Index of (n-1)th-generation flux in array FLX
INT(53)	NM2	Index of (n-2)th-generation flux in array FLX
INT(57)	ITER	Number of iteration at which fluxes are written on data set
INT(58)	NRST	Number of first iteration at which fluxes are to be written on FR.D1 or FA.D1
INT(59)	NRSTI	Increment in NRST after which FR.D1 or FA.D1 is to be written again

3. Local Variables

Name	Type	Definition
FRD1	R*8	'FR.D1' for SNIFF call
ADJ	R*8	'ADJOINT' for problem type
FAD1	R*8	'FA.D1' for SNIFF call
NCHECK	I*4	Number of next iteration at which fluxes are to be written on data set
NC	I*4	Number of next iteration at which new page will be begun

Name	Type	Definition	
N1	I*4	Data set reference number for FR.D1 or FA.D1	
KIND	I*4	Kind of fluxes written	
NDUM	I*4	Dummy variable used in updating flux generation indices	
IRETRN	I*4	Flag returned from ENDOT1 or ENDOT2	-1-error in routine 0-normal return 1-problem has converged
FIXK	R*8	$1/\mu$ factor (k_{fixed})	

4. Functions and tasks performed by this subroutine

This subroutine controls the performance of the source iterations and the writing of the appropriate flux data set FR.D1 or FA.D1 for the restart option for the one-dimensional diffusion theory source calculation.

It first sets up the page heading for the iteration history printout, and the number NC of the iteration at which the next new page will be begun. There are 15 iteration records to a page, in general, except that writing out of the flux data set always causes a new page to be begun.

The number NCHECK of the iteration after which the appropriate flux data set is to be written is also initialized at NRST; it will be incremented thereafter by NRSTI each time the data set is written.

OUTERS then, for each iteration, updates the flux generation positions so that the new generation flux will replace the oldest flux now present, and calls SSWEEP to control the mesh sweep by groups. It then calls either ENDOT1 (for no extrapolation or Chebyshev extrapolation) or ENDOT2 (for SOR extrapolation) to perform monitoring and acceleration if required and to test for convergence. If the problem has not converged, the routine writes the flux data set if required after the iteration, calling WFLX to write the second record, and continues to iterate. If the problem has converged, or if the maximum number of iterations is exceeded, control is returned to the calling program, and the output section will be the next to be performed.

5. Subprograms called by this subroutine

SNIFF	ENDOT2 (entry in INDOT2)
PAGHED	IGET (entry in POINTR)
SSWEEP (entry in ISSWEP)	WFLX
ENDOT1 (entry in INDOT1)	ERRMSG

6. Subprograms calling this subroutine

SRCCAL

7. Error messages generated by this subroutine

Message: *****SOURCE ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE SOUTER

Significance: The maximum number of iterations, NMAX, has been exceeded, and the problem has not converged.

Action taken: Execution of the remainder of the program, consisting of the output section, is performed as if the problem had converged.

Subroutine SRCCAL (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return to calling program for error encountered in a subprogram called

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(19)	MAXBLK	Size of POINTR container array in bulk storage
INT(32)	IMAX	Number of mesh intervals defined

3. Local Variables

Name	Type	Definition
LAST	I*4	Last location used in POINTR core container array
LASTB	I*4	Last location used in POINTR bulk container array

4. Functions and tasks performed by this subroutine

This subroutine controls the source calculation for the one-dimensional diffusion theory k-effective and source calculation module. It allocates storage for the total source and the intermediate results ω of the Choleski inversion, determines and prints the amount of space required in the POINTR container arrays for the calculation section, calls ISSUBS to initialize subroutines used for every iteration, and calls SOUTER to control the source iterations.

5. Subprograms called by this subroutine

PUTPNT (entry in POINTR)
ILAST (entry in POINTR)
ILASTB (entry in POINTR)
ISSUBS
SOUTER

6. Subprograms calling this subroutine

D1DCAL

7. Error messages generated by this subroutine - None

Subroutine STORBC (SAVE, BNDC, NBC)

1. Arguments

Name	Type	Dimension	Definition
SAVE	R*4	8	Address of MESHI from subroutine RBNDC (location of common block BCBLK)
BNDC	I*4 / R*4	8*NBCDEF	Boundary condition data from BNDC
NBC	I*4	--	Number of boundary condition record just read in RBNDC

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

For each boundary condition NBC, NBC=1,...,NBCDEF, defined in data set BC, this subroutine stores the eight boundary condition definition terms in the locations (1,NBC),..., (8, NBC) of the array BNDC. The boundary condition terms are obtained from the common block BCBLK through referencing its first element, MESHI. The first four elements of each eight are actually I*4; the second four are R*4.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

RBNDC

7. Error messages generated by this subroutine - None

Subroutine SUMCAL (LEAK, RMVL, SCAT, FISS, EXTS, BLNC, LEAKR, LEAKG, RMVLR, RMVLG, SCATR, SCATG, FISSR, FISSG, EXTSR, EXTSG, BLNCR, BLNCG, REGNAM, RMAX)

1. Arguments

Name	Type	Dimension	Definition
LEAK	R*8	RMAX*GMAX	Leakage by region and group
RMVL	R*8	RMAX*GMAX	Removal by region and group
SCAT	R*8	RMAX*GMAX	Scattering source by region and group
FISS	R*8	RMAX*GMAX	Fission source by region and group
EXTS	R*8	RMAX*GMAX	External source by region and group
BLNC	R*8	RMAX*GMAX	Balance by region and group
LEAKR	R*8	RMAX	Leakage by regions, summed over groups
LEAKG	R*8	GMAX	Leakage by groups, summed over regions
RMVLR	R*8	RMAX	Removal by regions, summed over groups
RMVLG	R*8	GMAX	Removal by groups, summed over regions
SCATR	R*8	RMAX	Scattering source by regions, summed over groups
SCATG	R*8	GMAX	Scattering source by groups, summed over regions
FISSR	R*8	RMAX	Fission source by regions, summed over groups
FISSG	R*8	GMAX	Fission source by groups, summed over regions
EXTSR	R*8	RMAX	External source by regions, summed over groups

Name	Type	Dimension	Definition
EXTSG	R*8	GMAX	External source by groups, summed over regions
BLNCR	R*8	RMAX	Balance by regions, summed over groups
BLNCG	R*8	GMAX	Balance by groups, summed over regions
REGNAM	R*8	RMAX	Region names
RMAX	I*4	--	Number of regions defined

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(1)	PROB	Problem type	'REAL' 'ADJOINT'
FLT(5)	PROBT	Calculation type	'K CALC' 'SOURCE'
INT(37)	GMAX	Number of groups defined	

3. Local Variables

Name	Type	Definition
ROW1	R*8	'REGION' for row title
ROW2	R*8	'GROUP' for row title
ADJ	R*8	'ADJOINT' for problem type
SORC	R*8	'SOURCE' for calculation type

Name	Type	Definition
LEAKTL	R*8	Total leakage for reactor
RMVLTL	R*8	Total removal for reactor
SCATTL	R*8	Total scattering source for reactor
FISSTL	R*8	Total fission source for reactor
EXTSTL	R*8	Total external source for reactor
BLNCTL	R*8	Total balance for reactor
G1	I*4	Group number for printing G1=1,2,...,GMAX, for real problem G1=GMAX, GMAX-1,...,1 for adjoint problem

4. Functions and tasks performed by this subroutine

This subroutine sums each quantity computed by region and group for the balance calculation over groups, over regions, and over the reactor, and prints the results. Quantities summed include the leakage contribution LEAK, the removal contribution RMVL, the scattering source contribution SCAT, the fission source contribution FISS, the external source contribution EXTS when present, and the balance term BLNC.

5. Subprograms called by this subroutine

PAGHED

6. Subprograms calling this subroutine

D1DOUT

7. Error messages generated by this subroutine - None

Subroutine TWODCR (ARAY, ICT, JCT, TITLE, RNAM, JPONE)

1. Arguments

Name	Type	Dimension	Definition
ARAY	R*8	ICT*JCT	Two-dimensional array to be printed
ICT	I*4	--	Number of rows in array
JCT	I*4	--	Number of columns in array
TITLE	R*8	9	Title of array (9A8)
RNAM	R*8	ICT*2	Pairs of region names delineating region interfaces (RBOUND in BLK array)
JPONE	I*4	--	Column numbering flag 0-number columns with integers 1, 2, ...,JCT 2-number columns with integers JCT, JCT-1, ..., 1

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
COLT	R*8	' GROUP' for column title
II	I*4	Number of first row to be printed on page
IMAX	I*4	Number of last row to be printed on page
IFLAG	I*4	New page flag 0-start new page 1-stay on same page (NROW+ IDIF < 57)
IEND	I*4	Number of last column (JCT)
MIN	I*4	Number of first column to be printed on page
MAX	I*4	Number of last column to be printed on page

Name	Type	Definition
NROW	I*4	Approximate number of rows printed on page (sum of IDIF's)
JCI	I*4	Increasing index on columns, JCI=1,...,MAX-MIN+1
JFLIP(7)	I*4	Flipped column integer indices for JPONE=2: IEND-MIN+1, IEND-(MIN+1)+1,..., IEND-MAX+1
IDIF	I*4	Approximate number of rows printed in column pass, IDIF=(IMAX-II+1)+4

4. Functions and tasks performed by this subroutine

This subroutine is a special version of TWODPR designed to print currents by groups across region interfaces. ICT is RMAX+1, and JCT is GMAX. The column heading is ' GROUP', and the row heading is INTERFACE / LEFT RIGHT, where under "LEFT" and "RIGHT" are printed the names of the regions delineating the region interface on the left and on the right, from array RNAM(ICT,2), the array RBOUND in the common block BLK, in 2(4X, A6) format.

The array members are printed in 1PE15.6 format. No transposition option is given; however, the decreasing numbering option on columns (JPONE=2) is offered for use in adjoint problems.

5. Subprograms called by this subroutine

PAGHED

6. Subprograms calling this subroutine

LKCAL

7. Error messages generated by this subroutine - None

Subroutine TWODFL (ARAY, ICT, JCT, TITLE, RNUM)

1. Arguments

Name	Type	Dimension	Definition
ARAY	R*8	ICT*JCT	Two-dimensional array to be printed
ICT	I*4	--	Number of rows in array
JCT	I*4	--	Number of columns in array
TITLE	R*8	9	Title of array (9A8)
RNUM	R*8	ICT	Numbers for rows of array (XBAR in BLK array)

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
COLT	R*8	' GROUP' for column title
II	I*4	Number of first row to be printed on page
IMAX	I*4	Number of last row to be printed on page
IFLAG	I*4	New page flag 0-start new page 1-stay on same page (NROW+IDIF<57)
IEND	I*4	Number of last column (JCT)
MIN	I*4	Number of first column to be printed on page
MAX	I*4	Number of last column to be printed on page
NROW	I*4	Approximate number of rows printed on page (sum of IDIF's)
IDIF	I*4	Approximate number of rows printed in column pass, IDIF=(IMAX-II+1)+4
ID	I*4	Integer row number I

4. Functions and tasks performed by this subroutine

This subroutine is a special version of TWODPR designed to print fluxes, fission densities, and other quantities for which ICT is IMAX and JCT is GMAX. The column heading is ' GROUP', and the row headings are two:

MESH MESH
INTERVAL and INTERVAL
NUMBER CENTER.

The mesh interval numbers are simply the integers 1, 2, ..., ICT, and the mesh interval centers the members of the floating point array RNUM(ICT), the array XBAR in the common block BLK, printed in F10.3 format.

The array members are printed in 1PE15.6 format. No transposition option is given. Adjoint arrays are re-ordered by group before printing by calling subroutine FLIP.

5. Subprograms called by this subroutine

PAGHED

6. Subprograms calling this subroutine

NRMCAL

7. Error messages generated by this subroutine - None

Subroutine TWODPR (ARAY, ICT, JCT, TITLE, ROWT, COLT, INVRT, IPONE, JPONE, RNUM, CNUM)

1. Arguments

Name	Type	Dimension	Definition	
ARAY	R*8	ICT*JCT	Two-dimensional array to be printed	
ICT	I*4	--	Number of rows in array	
JCT	I*4	--	Number of columns in array	
TITLE	R*8	9	Title of array (9A8)	
ROWT	R*8	--	Row title (A8)	
COLT	R*8	--	Column title (A8)	
INVRT	I*4	--	Array transposition flag	0-print array as given 1-transpose rows and columns in printing
IPONE	I*4	--	Row numbering flag	0-number rows with integers 1,2,...,ICT for INVRT=0, or 1,2,..., JCT for INVRT=1 1-number rows with members of floating-point array RNUM 2-label rows with members of alphanumeric array RNUM
JPONE	I*4	--	Column numbering flag	0-number columns with integers 1,2,...,JCT for INVRT=0, or 1,2,...,ICT for INVRT=1 1-number columns with members of floating-point array CNUM

Name	Type	Dimension	Definition
			2-number columns with integers JCT, JCT-1, ...,1 for INVRT=0, or ICT, ICT-1, ...,1 for INVRT=1
RNUM	R*8	ICT for INVRT=0 JCT for INVRT=1	Numbers or labels for rows, if IPONE=1 or IPONE=2
CNUM	R*8	JCT for INVRT=0 ICT for INVRT=1	Numbers for columns, if JPONE=1

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
II	I*4	Number of first row to be printed on page for INVRT=0
IJ	I*4	Number of first row to be printed on page for INVRT=1
IMAX	I*4	Number of last row to be printed on page for INVRT=0
JMAX	I*4	Number of last row to be printed on page for INVRT=1
MIN	I*4	Number of first column to be printed on page
MAX	I*4	Number of last column to be printed on page
IFLAG	I*4	New page flag 0-start new page 1-stay on same page (NROW+IDIF<57)
IEND	I*4	Number of last column (JCT for INVRT=0, ICT for INVRT=1)

Name	Type	Definition
NROW	I*4	Approximate number of rows printed on page (sum of IDIF's)
JCI	I*4	Increasing index on columns, JCI=1,..., MAX-MIN+1
JFLIP(7)	I*4	Flipped column integer indices for JPONE=2: IEND-MIN+1,IEND-(MIN+1)+1, ...,IEND-MAX+1
IDIF	I*4	Approximate number of rows printed in pass, IDIF=(IMAX-II+1)+4 for INVRT=0, IDIF=(JMAX-IJ+1)+4 for INVRT=1
ID	I*4	Integer row number I for IPONE=0
FID	R*8	Floating-point row number RNUM(I) for IPONE=1, or alphanumeric label RNUM(I) for IPONE=2

4. Functions and tasks performed by this subroutine

This subroutine prints a given two-dimensional array ARAY(ICT,JCT) in a format selected using the various options offered by the subroutine. The array is printed with a maximum of seven columns and 52 rows to a page. All rows are printed for a given group of seven or less columns before the next set of columns is begun.

The major option offered is that of transposing the array, by setting INVRT to 1, so that rows are printed as columns and columns as rows. This does not affect the array contents, but merely the order in which they are printed. It also does not affect column and row headings and labels; these are used as input and are not affected by the INVRT flag.

Other options concern the manner in which columns and rows are indexed. A general column title is printed above each column, and a general row title before every row; these general titles can be followed by various types of individual labels.

For rows, the options are as follows:

IPONE=0: the rows are numbered with the integers 1,2,...,ICT, for INVRT=0; for INVRT=1, of course, the range is 1,2,...,JCT.

IPONE=1 the rows are numbered with the members of the floating-point array RNUM(ICT) for INVRT=0, or RNUM(JCT) for INVRT=1, in F10.3 format.

IPONE=2: the rows are labeled with the members of the alphanumeric array RNUM(ICT) for INVRT=0, or RNUM(JCT) for INVRT=1, in A8 format.

For columns, the options are as follows:

JPONE=0: the columns are numbered with the integers 1,2,...,JCT for INVRT=0, or 1,2,...,ICT for INVRT=1.

JPONE=1: the columns are numbered with the members of the floating-point array CNUM(JCT) for INVRT=0, or CNUM(ICT) for INVRT=1, in F 10.3 format.

JPONE=2: the columns are numbered with the integers JCT, JCT-1, ..., 1 for INVRT=0, or ICT, ICT-1,...,. for INVRT=1.

If IPONE is zero, any array or dummy variable can be used in the calling sequence in place of RNUM, since RNUM will not be used; the same holds true for CNUM when JPONE is either zero or two.

As many complete sets of seven columns, each for all rows, as can be fit on one page will be printed to a page. Breaking up of sets of seven columns on successive pages will be done only if there are more than 52 rows. On each page, the general page heading and page number, set up in common block HEADER for subroutine PAGHED, are printed, followed by the input array title. Column and row headings are printed for each column and row. Array members are printed in 1PE15.6 format.

5. Subprograms called by this subroutine

PAGHED

6. Subprograms calling this subroutine

NRMCAL	FISREG (entry in FISCAL)
LKCAL	EXTCAL
REMCAL	BALCAL
SCTCAL	

7. Error messages generated by this subroutine - None

Subroutine WFLX (FLX, IMAX, GMAX, NTH, N1)

1. Arguments

Name	Type	Dimension	Definition
FLX	R*8	IMAX*GMAX*4 for up-scattering problems IMAX*GMAX*3 otherwise	Flux $\phi_{i,g}$ for four or three generations
IMAX	I*4	--	Number of mesh intervals defined
GMAX	I*4	--	Number of groups defined
NTH	I*4	--	Index of nth-generation flux in array FLX
N1	I*4	--	Data set reference number for FR.D1 or FA.D1

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine writes the second record of the flux data set FR.D1 or FA.D1 for the restart option for a source calculation.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

SOUTER

7. Error messages generated by this subroutine - None

Subroutine WIPE

1. Arguments - None

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine wipes out from the POINTR container array all arrays set up for the output section of the one-dimensional diffusion theory modules, so that the space can be used if necessary by the section following the output section.

5. Subprograms called by this subroutine

WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

DIF1D (when executing module NUC002)
D1DSCH (when executing module NUC004)

7. Error messages generated by this subroutine - None

5. DIF1D (NUC002) Common Block Descriptions

This section provides more detailed information about the common blocks used by the DIF1D subprograms. The common blocks appear in alphabetical order by name.

For each common block, all variables are listed in order of appearance in the block by name and type, and the definition of each is given. In addition, there appears for each variable a list of the subprograms in which the value of the variable is set or altered. Local names are listed for members of the /SINGLE/ common block.

A special case is the /BLKSTR/ common block, which contains the container array BLK used by the dynamic storage allocation module POINTR (see Volume I of this series). Section 6 contains a list of the names of all arrays which may be defined in the container array at any point in the execution of the module.

/BCBLK/

Name	Type	Subprograms where values are set or modified	Definition
MESHI	I*4	RBNDC	Mesh line number, negative if boundary condition is on left side of line, positive if on right side, from data set BC
ITYPE	I*4	RBNDC	Boundary condition type, from data set BC 1 - $C_1\phi' + C_2\phi = C_3$ 2 - $C_1\phi' + C_2/D\ \phi = C_3$ where D is the diffusion coefficient 3 - $\phi_{i+1} = C_1\phi_i + C_2\phi_i'$ $\phi_{i+1}' = C_3\phi_i + C_4\phi_i'$
GINT	I*4	RBNDC	Number of initial group to which condition applies, from data set BC
GFIN	I*4	RBNDC	Number of final group to which condition applies, from data set BC
C(4)	R*4	RBNDC	Four boundary condition coefficients

/BLKSTR/

Name	Type	Subprograms where values are set or modified	Definition
BLK(MAXSIZ + MAXBLK)	R*8	RCDEP	Large POINTR dynamic storage container array

/GNAME/

Name	Type	Subprograms where values are set or modified	Definition	
GEOM	R*8	CELCAL	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'

/HEADER/

Name	Type	Subprograms where values are set or modified	Definition
TITLE(10)	R*8	RCDEP	Page heading (10A6), from data set A.DIF1D
NPAGE	I*4	RCDEP PAGHED	Page number, initialized at zero and incremented with each printing

/NAMBLK/

Name	Type	Subprograms where values are set or modified	Definition
NAME	R*8	ERRMSG	Name of subroutine calling ERRMSG (A8)

/SINGLE/FLT(100),INT(100)

Name	Local Name	Subprograms where values are set or modified	Definition	
FLT(1)	PROB	RCDEP	Problem type, determined from SPCICN data	'REAL' 'ADJOINT' 'BOTH'
		DIF1D	Reset to	'REAL' 'ADJOINT'
FLT(2)	KEFF	RFLUX ENDOUT (entry in INDOUT)	k_{eff} in k-effective calculations, read from data set FR.D1 or FA.D1 and modified during iterations	
FLT(3)	ALPHA	RCDEP	Inverse reactor time period α, from data set SP.CICN	
FLT(4)	SCTEPS	RCDEP	Convergence criterion ε_S for up-scatter iterations in k-effective calculations, from data set A.DIF1D	

$$\max_G \left[\frac{GMAX + \sum_{I=1}^{IMAX} \left| \phi_{I,G}^{(\ell)} - \phi_{I,G}^{(\ell-1)} \right|}{\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(\ell-1)}} \right] < \varepsilon_S$$

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
FLT(5)	PROBT	RCDEP	Calculation type, determined from SP.CICN data 'K CALC' 'SOURCE'
FLT(6)	MU	RCDEP	1. k_{fixed} for source calculations, where k_{fixed} is read from data set SP.CICN
FLT(11)	EPOFLX	RCDEP	Convergence criterion ε_f on sum of absolute value of flux differences, from data set A.DIF1D $\frac{\sum_{I=1}^{GMAX}\sum_{I=1}^{IMAX}\left\lvert\phi_{I,G}^{(n)} - \phi_{I,G}^{(n-1)}\right\rvert}{\sum_{G=1}^{GMAX}\sum_{I=1}^{IMAX}\phi_{I,G}^{(n)}} \leq \varepsilon_f$ for k-effective calculation, or $\max_G\left[\frac{GMAX + \sum_{I=1}^{IMAX}\left\lvert\phi_{I,G}^{(n)} - \phi_{I,G}^{(n-1)}\right\rvert}{\sum_{G=1}^{GMAX}\sum_{I=1}^{IMAX}\phi_{I,G}^{(n-1)}}\right] \leq \varepsilon_f$ for source calculation
FLT(12)	EPSLAM	RCDEP	Convergence criterion ε_λ on difference of flux bounds, from data set A.DIF1D $(\bar{\lambda}^{(n)} - \underline{\lambda}^{(n)}) \leq \varepsilon_\lambda$, where $\bar{\lambda}^{(n)} = \max_{I,G}\left(\frac{\phi_{I,G}^{(n)}}{\phi_{I,G}^{(n-1)}}\right)$ and $\underline{\lambda}^{(n)} = \min_{I,G}\left(\frac{\phi_{I,G}^{(n)}}{\phi_{I,G}^{(n-1)}}\right)$

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
FLT(13)	EPSKEF	RCDEP	Convergence criterion ε_k on difference in k_{eff}, from data set A.DIF1D $\lvert k_{eff}^{(n)} - k_{eff}^{(n-1)} \rvert \leq \varepsilon_k$
FLT(14)	AOUTER	RCDEP	Interval shift factor a when applying Chebyshev extrapolation, from data set A.DIF1D
FLT(15)	FLXMON	RCDEP	Flux monitoring ratio, used in computing FLXNGI (FLT(73)), from data set A.DIF1D
FLT(31)	GEOM	RGEOM	Geometry type 'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'
FLT(32)	XL	RMESH	Left reactor boundary, X(1), from data set GEOM
FLT(33)	XR	RMESH	Right reactor boundary, X(IMAX+1), from data set GEOM
FLT(71)	POWIN	RCDEP	Power normalization factor P_0 for BURNUP calculation, used in writing data set FR.PN, from data set A.DIF1D. No power normalization performed if POWIN = 0.
FLT(72)	FLXCVG	RFLUX ENDOUT (entry in INDOUT) ENDOT1 (entry in INDOT1) ENDOT2 (entry in INDOT2)	Sum of fluxes used in checking for convergence $\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(n)}$ for k-effective calculation $\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(\ell-1)}$ for k-effective calculation upscatter iterations $\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(n-1)}$ for source calculation

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
FLT(73)	FLXNGL	RFLUX ENDOUT (entry in INDOUT) ENDOT1 (entry in INDOUT1) ENDOT2 (entry in INDOT2)	Factor below which fluxes are neglected in computation of flux bounds $\bar{\lambda}^{(n)}$ and $\underline{\lambda}^{(n)}$ $\text{FLXNGL}=\text{FLXMON}*\sqrt{\dfrac{\sum_{I=1}^{GMAX}\sum_{I=1}^{IMAX}[\phi_{I,G}^{(n-1)}]^2}{GMAX*IMAX}}$
INT(1)	MKRTHY	RCDEP	Problem type, from data set SP.CFCN 0 - real problem 1 - adjoint problem 2 - both real and adjoint problems
INT(2)	KINOPT	RCDEP	Kinetics option, from data set A.DIF1D 0 - no kinetics 1 - source due to delayed neutrons 2 - delayed data for alpha calculations
INT(3)	LMAX	RCDEP	Maximum number of upscatter iterations, from data set A.DIF1D
INT(4)	MAXSIZ	RCDEP	Maximum size of POINTR container array in core, from data set. A.DIF1D

/SINGLE/FLT(100),INT(100)(Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(5)	IPRINT	RCDEP	POINTR debugging flag, from data set A.DIF1D 0 - no print out 1 - standard print out 2 - debugging dump print out 3 - full print out
INT(7)	KOUT	RCDEP	Diffusion output flag, from data set A.DIF1D 0 - problem description and iteration history print out 1 - above plus geometry and composition input data print out 2 - above plus standard balance output 3 - above plus power calculations
INT(9)	FSOPT	RCDEP	Fission source flag, from data set A.DIF1D 0 - no fission source by intervals output 1 - fission source by intervals printed and written on FSR.D1 or FSA.D1

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(10)	XTROUT	RCDEP	Extrapolation type, from data set A.DIF1D 0 - no extrapolation 1 - Chebyshev extrapolation 2 - successive over-relaxation (for source calculation only)
INT(11)	NMAX	RCDEP	Maximum number of outer iterations, from data set A.DIF1D
INT(12)	NSTART	RCDEP	Number of unextrapolated iterations to be done before beginning to extrapolate, from data set A.DIF1D
INT(13)	NTEST	RCDEP	Number of outer iterations to be done in extrapolation sequence before testing for payoff, from data set A.DIF1D
INT(14)	NEXT	RCDEP	Maximum number of extrapolated iterations to be done in an extrapolation sequence (not applicable to SOR), from data set A.DIF1D
INT(15)	NUNEXT	RCDEP	Number of unextrapolated iterations to be done following one extrapolation sequence before beginning another, from data set A.DIF1D
INT(16)	P	OUTERS (entry in INDOUT) SOUTER ENDOT1 (entry in INDOT1) ENDOT2 (entry in INDOT2)	Extrapolation iteration counter (degree of Chebyshev polynomial)

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(17)	SCHOPT	RCDEP	Search data set output 0 - XS.C.MIN and B.HOMOG or GEOM written after search 1 - XS.C.MIN and B.HOMOG or GEOM not written after search
INT(19)	MAXBLK	RCDEP	Maximum size of POINTR container array in bulk (LCS) storage, from data set A.DIF1D
INT(31)	RMAX	RGEOM	Number of regions defined in problem, from data set GEOM
INT(32)	IMAX	RGEOM	Number of mesh intervals defined in problem, from data set GEOM
INT(34)	BCMAX NBCDEF	RBNDC	Number of boundary conditions and non-continuity interface conditions defined in problem, from data set BC
INT(35)	CMAX	RGEOM RCOMP	Number of compositions defined in problem, read from data set GEOM and then from data set XS.C.MIN
INT(36)	FSCMAX	RCOMP	Number of fissionable compositions defined in problem, from data set XS.C.MIN
INT(37)	GMAX	RCOMP	Number of energy groups defined in problem, from data set XS.C.MIN
INT(38)	GMAXUP	RCOMP	Number of lowest-numbered group from which up scattering occurs in problem (GMAXUP=GMAX if there is no upscattering)
INT(39)	MTRCHI	RCOMP	Fission spectrum array type 0 - χ-vector 1 - χ-matrix

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(40)	CHIDIM	RCOMP	First dimension of fission spectrum array CHI, CHI(CHIDIM, GMAX) CHIDIM=FSCMAX*GMAX**MTRCHI
INT(41)	SCTDIM	RCOMP DIMSCT	First dimension of packed scattering cross section array SCTXS, SCTXS(SCTDIM,GMAX) SCTDIM is the sum over compositions of the maximum possible band width for each composition $SCTDIM = \sum_{C=1}^{CMAX} SCTLIM(1,C)+SCTLIM(2,C)+1$
INT(44)	NFAM	RCOMP	Number of families of delayed data, from data set XS.C.MIN
INT(49)	NGEOM	RGEOM	Number of geometry type, read from data set GEOM as three values and modified to four 1 - cylindrical slab 2 - rectangular slab 3 - cylinder 4 - sphere
INT(50)	NBLACK	RCDEP	Number of black composition group intervals defined, from data set A.DIF1D
INT(51)	NTH	RCDEP OUTERS SWEEP (entry in ISWEEP) SOUTER SSWEEP (entry in ISSWEP)	Position in flux array of nth generation of fluxes; $\phi_{i,g}^{(n)}$ is in the FLX(I,G,NTH) Also position of ℓth generation of fluxes in an upscattering problem
INT(52)	NM1	RCDEP OUTERS SOUTER	Position in flux array of (n-1)th generation of fluxes; $\phi_{i,g}^{(n-1)}$ is in FLX (I,G,NM1)

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(53)	NM2	RCDEP OUTERS SOUTER	Position in flux array of (n-2)th generation of fluxes; $\phi_{i,g}^{(n-2)}$ is in FLX(I,G,NM2)
INT(54)	NDUM LM1	RCDEP SWEEP (entry in ISWEEP)	Position in flux array of (ℓ-1)th generation of fluxes for upscattering problem; $\phi_{i,g}^{(\ell-1)}$ is in FLX(I,G,LM1)
INT(57)	ITER	OUTERS SOUTER	Iteration number at which flux data set FR.D1 or FA.D1 is written out
INT(58)	NRST	RCDEP	First iteration number at which FR.D1 or FA.D1 will be written out for restart option, from data set A.DIF1D
INT(59)	NRSTI	RCDEP	Increment in NRST at which succeeding intervals FR.D1 or FA.D1 will be written again for restart option, from data set A.DIF1D
INT(88)	NFIN	RGEOM	Number of finite geometry type, from data set GEOM
INT(89)	NIL	RGEOM	Number of finite directions defined per region, from data set GEOM
INT(93)	IPROB	RCDEP	Problem type, from data set SP.CICN 1 - source calculation 2 - k_{eff} calculation 3 - concentration search 4 - buckling search 5 - alpha search 6 - dimension search
INT(94)	KFLAG	DIF1D	Dummy search flag needed by RVEL, set to zero

6. DIF1D (NUC002) Dynamic Storage Array Allocation

This section describes all arrays which may be defined by this module in common block /BLKSTR/BLK, using the facilities of the dynamic storage allocation module POINTR (see Volume I). The arrays are listed in alphabetical order by the name given them in the POINTR name table. For each array, the following information is given:

1. type
2. dimension, in terms of module variables
3. names of subprograms where action is taken that affects the status of the array in the container, followed in parentheses by the action that is taken in the subprogram (Here, significant POINTR calls are generally PUTPNT, which allocates space in the container for the array; REDEF, which sets aside space if the array is not present or redefines its space allocation if it is; and WIPOUT, which removes the array from the name table, thus making it inaccessible. The meaning of any other POINTR calls listed can be found in the POINTR documentation.)
4. names of subprograms in which the value of elements of the array are set or altered
5. the definition of the array contents.

If the meaning of a variable used as the dimension of an array is not known, it can be found in the subprogram in which a call to PUTPNT or REDEF is made for that array.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
A	R*8	IMAX*GMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	Finite difference coefficients $a_{i,g}$ across the interface of cell i with cell i+1 for group g
BLKCMP	R*8	NBLACK	RCDEP(PUTPNT) RCOMP(WIPOUT)	RBLACK	Names of compositions defined as black compositions in data set A.DIF1D, if any
BLKGRP	I*4	NBLACK*2	RCDEP(PUTPNT) RCOMP(WIPOUT)	RBLACK RGROUP	Numbers of initial and final energy groups between which a composition names in BLKCMP is defined as being black in data set A.DIF1D. For a composition BLKCMP(N), BLKGRP(N,1) is the initial energy group and BLKGRP(N,2) is the final energy group. If BLKGRP(N,1)=0, the condition is applied over all energy groups in the composition. If BLKGRP(N,2)=0 and BLKGRP(N,1)≠0, the condition applies only to the energy group indicated by BLKGRP(N,1).
BLNC	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	BALCAL	Output balance sum by regions and groups. BLNC(R,G)=-LEAK(R,G) -RMVL(R,G)+SCAT(R,G)+FISS(R,G)/KEFF for a k-effective calculation. BLNC(R,G)=-LEAK(R,G)-RMVL(R,G)+SCAT (R,G)+FISS(R,G)+EXTS(R,G) for a source calculation.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
BLNCG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Output balance sum by groups $BLNCG(G) = \sum_{R=1}^{RMAX} BLNC(R,G)$
BLNCR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Output balance sum by regions $BLNCR(R) = \sum_{G=1}^{GMAX} BLNC(R,G)$
{BNDC, JBNDC}	{R*4, I*4}	8*NBCDEF	RBNDC(PUTPNT)	STORBC	For each boundary condition NBC (NBC=1,...,NBCDEF) defined, eight terms as follows: JBNDC(1,NBC) - mesh line number, positive if the condition applies to the right side of the line, negative if it applies to the left JBNDC(2,NBC) - boundary condition type 1 - $C_1\phi' + C_2\phi = C_3$ 2 - $C_1D\phi' + C_2\phi = C_3D$ where D is the diffusion coefficient 3 - $\phi_{i+1} = C_1\phi_i + C_2\phi_i'$ $\phi_{i+1} = C_3\phi_i + C_4\phi_i$ JBNDC(3,NBC) - number of initial (higher) energy group to which condition applies JBNDC(4,NBC) - number of final (lower) energy group to which condition applies BNDC(5,NBC) = C_1, BNDC(6,NBC) = C_2, BNDC(7,NBC) = C_3, BNDC(8,NBC) = C_4

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
BSQ	R*8	RMAX*2	RGEOM(PUTPNT)	RMESH	Buckling values from data set GEOM, where BSQ(R,1) is B^2 in the traverse direction and BSQ(R,2) is B^2 in the axial direction for region R. If B^2 is not defined for a given region R in a given direction N, BSQ(R,N) = 0.
C	R*8	IMAX*GMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	Finite difference coefficients $c_{i,g}$ across the interface of cell i with cell i-1 for group g
CHI	R*8	CHIDIM*GMAX	RCOMP(PUTPNT)	RGROUP	Fission spectrum fractions, stored as vectors (χ_c^g) or matrices ($\chi_c^{g,g}$) for each fissionable composition
CMPMSH	I*2	IMAX	RGEOM(PUTPNT)	RMESH	Composition-to-mesh correspondence, defined in data set GEOM. For any mesh interval I, CMPMSH(I) is the number of the composition defined in I.
CMPNAM	R*8	CMAX	RGEOM(PUTPNT)	RMESH	Composition names, defined in data set GEOM. CMPNAM(C) is the name of the composition whose number is C.
CMPREG	I*2	RMAX	RGEOM(PUTPNT)	RMESH	Composition-to-region correspondence, defined in data set GEOM. For any region R, CMPREG(R) is the number of the composition defined in R.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
CRDUM	I*4	RMAX	RGEOM(PUTPNT) RGEOM(WIPOUT)	RMESH	Temporary storage for composition-to-region correspondence array as read from data set GEOM (stored in I*2 array CMPREG)
CURR	R*8	(RMAX+1) *GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	LKCAL	Output currents at region interfaces by group, in the direction of the positive x-axis
DC	R*8	CMAX*GMAX	RCOMP(PUTPNT)	RGROUP	Diffusion coefficients by composition and group. DC(C,G) = 0 for a black composition and group. Diffusion coefficients are formed from the transport cross section read into this array from XS.C.MIN.
DCCHI	R*8	NFAM*GMAX for kinetics option 2 for kinetics option 1 or 0	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Delayed neutron-emitter-yield spectrum $\chi^{g'1}$ by family and group, from data set XS.DELAY
DECCON	R*8	NFAM for kinetics option 1 or 2 0 for kinetics option 0	RCOMP(PUTPNT) RCOMP(WIPOUT) RDIST(REDEF)	RDELAY	Decay constants by family for delayed neutrons, from data set XS.DELAY

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
DICHI	R*8	NFAM*GMAX for kinetics option 1 or 2 0 for kinetics option 0	RCOMP(PUTPNT) RCOMP(WIPOUT)	RDELAY	Delayed neutron spectrum χ_i^g by family and group, from data set XS.DELAY
DNSCT	I*4	GMAX	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Number of groups of downscattering given for each group in a given composition block in data set XS.C.MIN
DX	R*8	IMAX	RGEOM(PUTPNT)	RMESH	Mesh interval lengths ΔX DX(I) = X(I+1) - X(I)
E	R*8	IMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	Central finite difference coefficient for cell i, including removal, $D.B^2$ terms, and boundary condition terms, if present
EXTBC	R*8	IMAX*GMAX for source calculation 0 for k-effective calculation	FDCGEN(REDEF)	FDC1D	Source due to inhomogeneous interface or boundary conditions for each mesh interval and group

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
EXTDIS	R*8	RMAX*2	RGEOM(PUTPNT)	RMESH	Extrapolation distance $[H_r-(H_L)_r]$ by region and direction, from data set GEOM, where the extrapolated half-height H_r is first read into the array, and the actual half-height is then subtracted from it. EXTDIS(R,1) is the extrapolated half-height in the transverse direction for region R, and EXTDIS(R,2) the extrapolated half-height in the axial direction. If buckling is not defined for a given region R in a given direction N, EXTDIS(R,N) = 0.
EXTS	R*8	RMAX*GMAX for source calculation 0 for k-effective calculation	D1DOUT(PUTPNT) WIPE(WIPOUT)	EXTCAL	External source by region and group for output edit, including external source due to inhomogeneous interface conditions and to delayed data, if present
EXTSG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	External source by groups for output edit $EXTSG(G) = \sum_{R=1}^{RMAX} EXTS(R,G)$ EXTSG(G) = 0 for any G for a k-effective calculation

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
EXTSR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	External source by regions for output edit $EXTSR(R) = \sum_{G=1}^{GMAX} EXTS(R,G)$ EXTSR(R) = 0 for any R for a k-effective calculation
FISAVG	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FISREG (entry in FISCAL)	Region-averaged fission source integrals by group, for output edit
FISDEN	R*8	IMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FISCAL	Fission source density by mesh interval and group, for output edit
FISS	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FISREG (entry in FISCAL)	Fission source integrals by region and group, for output edit, including $1/k_{fixed}$ term for source calculation
FISSG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Fission source integrals by groups $FISSG(G) = \sum_{R=1}^{RMAX} FISS(R,G)$
FISSOR	R*8	IMAX*GMAX	D1DOUT(PUTPNT)	FISCAL	Fission source integrals by mesh interval and group, for data set FSR.D1 or FSA.D1, printed and written only if FSOPT (INT(9))≠0

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
FISSR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Fission source integrals by regions $FISSR(R) = \sum_{G=1}^{GMAX} FISS(R,G)$
FISSRC	R*8	IMAX*GMAX	KEFCAL(REDEF) D1DOUT(WIPOUT)	RSORC	Temporary array for use in weighting external fixed source data, which contains $\sum_{G'=1}^{GMAX} \chi_N^{G'G} \quad GMAX$ for each fissionable composition N and group G for real problems with a χ-matrix, and $\sum_{G'=1}^{GMAX} (\nu \cdot \sigma_f)_N^{G'}$ for each fissionable composition N for adjoint problems
FISXS	R*8	FSCMAX* GMAX	RCOMP(PUTPNT)	RGROUP	Fission cross section σ_f by composition and group from XS.C.MIN. Data is stored only for fissionable compositions.
FIXSRC	R*8	IMAX*GMAX for source calculation 0 for k-effective calculation	RDIST(PUTPNT) RDIST(REDEF)	RSORC	External fixed source by interval and group from data sets ES.D1D, weighted using fission spectrum weighting over groups, if desired

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
FLX	R*8	IMAX*GMAX*4 for problem with up-scattering IMAX*GMAX*3 otherwise	RDIST(PUTPNT)	RFLUX INVERT ENDOUT (entry in INDOUT) ENDOT1 (entry in INDOT1) ENDOT2 (entry in INDOT2) NRMCAL FLIP	Three generations of fluxes, $\phi_{i,g}^{(n)}$, $\phi_{i,g}^{(n-1)}$, and $\phi_{i,g}^{(n-2)}$, during iterations, referenced by NTH(INT(51)), NM1(INT(52)) and NM2(INT(53)) One additional generation, $\phi_{i,g}^{(\ell-1)}$, for upscatter iterations, referenced by LM1(INT(54)) Fluxes (NTH generation only) for output edits, normalized for k-effective calculation, not normalized for source calculation, and written on data set FR.D1 or FA.D1
FLXVOL	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FVCAL NRMCAL	Regional flux integrals by group for output edit, normalized for k-effective calculations, not normalized for source calculations
FSCMP	I*2	CMAX	RCOMP(PUTPNT)	RGROUP	Indices corresponding to fissionable compositions. For any composition C, if FSCMP(C)=0, C is not a fissionable composition; if FSCMP(C) is some non-zero integer, it is the index of that composition in arrays dimensioned with FSCMAX.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
GAMMA	R*8	IMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	Off-diagonal elements γ, for a given group, of upper matrix U which results from splitting tri-diagonal matrix A, whose elements are the finite difference coefficients
LEAK	R*8	RMAX*GMAX	D1DOUT(PUTPNT WIPE(WIPOUT)	LKCAL	Leakage by region and group for output edits, including $D \cdot B^2$ terms if present
LEAKG	R*8	GMAX	D1DOUT(PUTPNT WIPE(WIPOUT)	SUMCAL	Leakage by groups $LEAKG(G) = \sum_{R=1}^{RMAX} LEAK(R,G)$
LEAKR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Leakage by regions $LEAKR(R) = \sum_{G=1}^{GMAX} LEAK(R,G)$
MSHCMP	I*4	IMAX	RGEOM(PUTPNT) RGEOM(WIPOUT)	RMESH	Temporary storage for composition-to-mesh correspondence as read from data set GEOM (stored in I*2 array CMPMSH)
MSHREG	I*4	IMAX	RGEOM(PUTPNT) RGEOM(WIPOUT)	RMESH	Temporary storage for region-to-mesh correspondence as read from data set GEOM (stored in I*2 array REGMSH)

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
NUFIS	R*8	FSCMAX*GMAX	RCOMP(PUTPNT)	RGROUP	Nu times fission cross section $\nu \cdot \sigma_f$ by composition and group, where ν is the number of neutrons released per fission for that group, from data set XS.C.MIN. Data stored only for fissionable compositions.
OMEGA	R*8	IMAX	KEFCAL(REDEF) SRCCAL(PUTPNT) D1DOUT(WIPOUT)	INVERT	The solution, $L\omega = F$, of the down sweep in the Choleski inversion of the tri-diagonal matrix $A = LU$, used in the up sweep $Ux = \omega$. This is the method used to solve a system of equations $Ax = F$.
PRECON	R*8	NFAM*IMAX if kinetics option = 1 0 if kinetics option = 0 or 2	RDIST(REDEF) D1DOUT(WIPOUT)	RSORC	Precursor concentrations ξ_g by family and group, from data set XS.DELAY
PWRAVG	R*8	RMAX if KOUT = 3 0 if KOUT < 3	D1DOUT(PUTPNT) WIPE(WIPOUT)	PWRCAL	Average power by region for output edits
PWRDEN	R*8	IMAX if KOUT = 3 0 if KOUT < 3	D1DOUT(PUTPNT) WIPE(WIPOUT)	PWRCAL	Average power by interval for output edits

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
PWRREG	R*8	RMAX if KOUT = 3 0 if KOUT < 3	D1DOUT(PUTPNT) WIPE(WIPOUT)	PWRCAL	Power by region for output edits
R	R*8	IMAX*GMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	The terms whose reciprocals are the diagonal elements of the lower triangular matrix L of the tri-diagonal matrix A = LU for the Choleski inversion of A
RBOUND	R*8	(RMAX+1)*2	D1DOUT(PUTPNT) WIPE(WIPOUT)	LKCAL	Pairs of region names delineating region interfaces for printing currents in output edits. The first name in each pair is that of the region on the left of the interface; the second name is that of the region on the right.
REGBRN	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	NRMCAL	Region-averaged power-normalized flux integrals for burnup, computed and written on data set FR.PN if input power factor from A.DIF1D is non-zero
REGFLX	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	NRMCAL	Region-averaged flux integrals by group, for output edit, normalized for k-effective calculations, not normalized for source calculations

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
REGHHT	R*8	RMAX*2	RGEOM(PUTPNT)	RMESH	Actual region half-height $(H_L)_r$ from data set GEOM, where REGHHT(R,1) is the half-height in the transverse direction and REGHHT(R,2) the half-height in the axial direction for region R. If a region half-height is not defined for a given region R and direction N, or if BSQ(R,N) = 0., REGHHT(R,N) = .5.
REGMSH	I*2	IMAX	RGEOM(PUTPNT)	RMESH	Region-to-mesh correspondence, defined in data set GEOM. For any mesh interval I, REGMSH(I) is the number of the region to which I belongs.
REGNAM	R*8	RMAX	RGEOM(PUTPNT)	RMESH	Region names, defined in data set GEOM. REGNAM(R) is the name of the region whose number is R.
REGVOL	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FVCAL	Physical region volume for $B^2 \neq 0$, or volume/unit height for $B^2 = 0$ (from PHYVOL function)
REM	R*8	CMAX*GMAX	RCOMP(PUTPNT)	RGROUP	Removal cross section σ_R by composition and group, from data set XS.C.MIN

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
RMVL	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	REMCAL	Removal contribution to balance by region and group for output edit, including α/v terms
RMVLG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Removal by groups $RMVLG(G) = \sum_{R=1}^{RMAX} RMVL(R,G)$
RMVLR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Removal by regions $RMVLR(R) = \sum_{G=1}^{GMAX} RMVL(R,G)$
SCAT	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SCTCAL	Scattering source contribution to balance by region and group for output edit
SCATG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Scattering source by groups $SCATG(G) = \sum_{R=1}^{RMAX} SCAT(R,G)$
SCATR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Scattering source by regions $SCATR(R) = \sum_{G=1}^{GMAX} SCAT(R,G)$
SCTDUM	R*8	CMAX*GMAX* GMAX	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Array used to read in full scattering cross section from XS.C.MIN before packing into SCTXS

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
SCTLIM	I*2	2*CMAX	RCOMP(PUTPNT)	RGROUP	Scattering array band widths for each composition C. SCTLIM(1,C) is the maximum width of the downscattering band for C, and SCTLIM(2,C) is the maximum width of the upscattering band for C. SCTLIM(1,C) + SCTLIM(2,C) + 1 is the maximum band width.
SCTLOC	I*2	CMAX*GMAX	RCOMP(PUTPNT)	DIMSCT	Location of specified scattering band in SCTXS. For a given composition C and group G, SCTLOC(C,G) gives the location of the band of elements $\sigma_c^{g'g}$.
SCTXS	R*8	SCTDIM* GMAX	RCOMP(PUTPNT)	RGROUP PCKSCT	Scattering cross section $\sigma_c^{g'g}$ by composition, band width, and group, packed from SCTDUM
SETCHI	R*8	ISCHI* GMAX	RCOMP(PUTPNT) RCOMP(WIPOUT)	GETCHI	Set-wide fission spectrum $\chi^{g'g}$ if one is defined in data set XS.C.MIN. Values are transferred into the CHI array for compositions using the set-wide $\chi^{g'g}$.
TOTSRC	R*8	IMAX	KEFCAL(REDEF) SRCCAL(PUTPNT) D1DOUT(WIPOUT)	TSRC (entry in ITSRC) SOURCE (entry in ISOURC)	Total source by mesh interval for one group, including fission source and scattering source and fixed external source and source due to inhomogeneous interface conditions if present, during iterations

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
UPSCT	I*4	GMAX	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Number of groups of upscattering given for each group in a given composition block in XS.C.MIN
VEL	R*8	GMAX	RCOMP(PUTPNT)	RVEL	Neutron speeds v_g for kinetics option 2, read from data set XS.ISO2 or XS.ISO. VEL(G) = 1, for all groups if data is not needed or not present.
WFLAG	I*4	IMAX	RDIST(PUTPNT) RDIST(WIPOUT)	RSORC	Flags from first record of ES.D1D determining whether fission-spectrum weighting of external fixed source over groups is to be done. For any interval I, if WFLAG(I) = 0, external source is used as written in ES.D1D; if WFLAG(I) = 1, the external source in that interval is distributed over all groups.
X	R*8	IMAX+1	RGEOM(PUTPNT)	RMESH	Mesh line locations, from data set GEOM
XBAR	R*8	IMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	NRMCAL	Mesh interval centers for output prints XBAR(I) = X(I) + .5*DX(I)

7. DIF1D (NUC002) Data Set Usage

This section contains descriptions of all data sets used by this module. For each data set used, the following information is given:

1. the name of the data set, if it is an ARC System data set, or a description of its contents and function if it is a scratch data set used only within the module;
2. the record structure of the data set, if it is a scratch data set;
3. names of subprograms in which the data set is read and/or written, along with a list of the record types which are read and/or written and the name of the variable containing the data set reference number.

A complete description of the record structure and contents of each ARC System data set can be found in Volume I of this series.

Name or description of data set - A.DIF1D, One-dimensional Diffusion Theory Module-dependent BCD Input

Record structure - ARC System data set

Read in subprograms	Card types	Reference variable
RCDEP	One of Problem Title (first type 01), one of Problem Options (second type 01), one of Upscatter Iteration and Normalization Control (type 02), one of Iteration and Array Control (type 03), one of Convergence Criteria (type 04)	N
RBLACK	As many of Black Region Control (type 07) as are present	N

Written in subprograms - None

Name or description of data set - BC, Boundary Condition Specifications

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RBNDC	One of Specifications (type 1) and all of Boundary Conditions (type 2) for vertical boundaries	NTAPE

Written in subprograms - None

Name or description of data set - ES.D1D, One-dimensional Distributed External Source

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RSORC	One of Specifications (type 1) and one of One-dimensional Distributed External Source (type 2)	NTAPE

Written in subprograms - None

Name or description of data set - FA.D1, One-dimensional Adjoint Group Fluxes

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RFLUX	One of Specifications (type 1) and one of One-dimensional Adjoint Flux (type 2)	NTAPE
FLXIN	Same as above	N1

Written in subprograms	Record types	Reference variable
OUTERS	One of Specifications (type 1) and one of One-dimensional Adjoint Flux (type 2)	N1
SOUTER	One of Specifications (type 1)	N1
WFLX	One of One-dimensional Adjoint Flux (type 2)	N1
NRMCAL	One of Specifications (type 1) and one of One-dimensional Adjoint Flux (type 2)	NFAD1

Name or description of data set - FR.D1, One-dimensional Real Group Fluxes

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RFLUX	One of Specifications (type 1) and one of One-dimensional Real Flux (type 2)	NTAPE

Written in subprograms	Record types	Reference variable
OUTERS	One of Specifications (type 1) and one of One-dimensional Real Flux (type 2)	N1
SOUTER	One of Specifications (type 1)	N1
WFLX	One of One-dimensional Real Flux (type 2)	N1
NRMCAL	One of Specifications (type 1) and one of One-dimensional Real Flux (type 2)	NFRD1

Name or description of data set - FR.PN, Power-normalized Real Regional Group Fluxes

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
NRMCAL	One of Specifications (type 1) and one of Power-normalized Regional Group Flux (type 2)	NFRPN

Name or description of data set - FSA.D1, One-dimensional Adjoint Fission Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
NRMCAL	One of Specifications (type 1) and one of One-dimensional Adjoint Fission Source (type 2)	NFSAD1

Name or description of data set - FSR.D1, One-dimensional Real Fission Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
NRMCAL	One of Specifications (type 1) and one of One-dimensional Real Fission Source (type 2)	NFSRD1

Name or description of data set - GEOM, Geometry Data

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RGEOM	One of Specifications (type 1)	NTAPE
RMESH	One of Boundaries (type 2), one of Region-Interval Correspondence (type 3), one of Composition-Interval Correspondence (type 4), one of Composition-Region Correspondence (type 5), one of Area Definitions (type 6) skipped if present, one of Zone Definitions (type 7) skipped if present, one of Finite Geometry Data (type 8) if present, and one of Labels (type 9)	NTAPE

Written in subprograms - None

Name or description of data set - SP.CICN, Module-independent Data

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RCDEP	One of Specifications (type 1)	NS

Written in subprograms - None

Name or description of data set - XS.C.MIN, Macroscopic Composition Cross Sections

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RCOMP	One of Specifications (type 1 of File 1)	NTAPE

Read in subprograms	Record types	Reference variable
GETCHI	One of Set Chi (type 2 of File 2) if present	NTAPE
RGROUP	One composition block, consisting of one record of Composition Specifications (type 3 of File 2) and GMAX records of Composition Macroscopic Group Cross Sections (type 4 of File 2)	NTAPE

Name or description of data set - XS.ISO2, Problem-dependent Microscopic Group Cross Sections, or XS.ISO, Microscopic Group Cross Sections

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RVEL	One of File Size (type 1 of File 1), one of Isotope Names (type 2 of File 1) skipped, and one of Group Structure (type 3 of File 1)	NTAPE

Written in subprograms - None

---

8. DIF1D (NUC002) Error Messages

This section lists the error messages generated by module DIF1D in alphabetical order by subprogram or entry name. For each error message, the significance of the message is given, along with the action taken by the subprogram.

In addition to the errors listed, execution of the module can be terminated for an error in POINTR dynamic storage array allocation. The names of the subprograms in which such an error is tested for, and from which an error return can occur, are

DIMSCT
FDC1D
ISSUBS
ISUBS
PCKSCT
PCOMP
RBNDC
RDELAY
REORDR
RFLUX
RGROUP
RMESH
RSORC
RVEL

No message except the one that execution has been terminated will appear for an error in dynamic storage array allocation.

CELCAL

Message: *****GENERAL INTERNAL INTERFACE CONDITIONS SPECIFIED FOR I=iiii, G=iiii

ERROR NUMBER $\left\{\begin{matrix}1\\2\end{matrix}\right\}$ IN SUBROUTINE CELCAL

Significance: An interface condition (ITYPER or ITYPEL) not equal to 0, 1, or 2 was returned to CELCAL from VERBND (entry in GETBND). This is assumed to indicate that an interface condition of 3, indicating general internal interface conditions, was specified in data set BC, and hence by implication that the internal interface condition type 2 was specified in A.NIP (card type 10, cols. 19-24), for this I and G. An error number 1 indicates that the error was encountered on the right of cell i,g; an error number 2, that it was encountered on the left.

Action taken: The IERR flag is set to 1 and control is returned to the calling program. No calculations are performed.

DIF1D

Message: *****EXECUTION OF MODULE NUC002 HAS BEEN TERMINATED

Significance: An error has been encountered in the execution of the module which has made successful execution of the module impossible.

Action taken: The error flag IERR is set to 1 and control is returned to the main program for NUC002.

ENDOT1 (entry in INDOT1)

Message: *****OUTER ITERATIONS HAVE DIVERGED TWELVE TIMES IN SUCCESSION

ERROR NUMBER 1 IN SUBROUTINE ENDOT1

Significance: Following Chebyshev extrapolation, the iterations have diverged on twelve successive passes.

Action taken: IRETRN is set to -1, and execution is terminated.

ENDOUT (entry in INDOUT)

Message: *****FLUX=-x.xxxxxxxExx FOR I=iiii G=iiii
ABSOLUTE VALUE OF FLUX WILL BE USED

ERROR NUMBER 1 IN SUBROUTINE ENDOUT

Significance: The flux for the specified mesh interval and group has gone negative following Chebyshev acceleration. It is assumed that the flux is very close to zero for this to have occurred.

Action taken: Flux for this mesh interval and group is set to the absolute value of the flux.

Message: *****OUTER ITERATIONS HAVE DIVERGED TWELVE TIMES IN SUCCESSION

ERROR NUMBER 2 IN SUBROUTINE ENDOUT

Significance: Following Chebyshev extrapolation, the iterations have diverged on twelve successive passes.

Action taken: IRETRN is set to -1, and execution is terminated

FLXIN

Message: ERROR NUMBER 1 IN SUBROUTINE FLXIN

Significance: The data set FA.D1 could not be found.

Action taken: Execution is terminated.

Message: *****ERROR IN READING FLUX GUESS*****
IMAX=iiii IMX1=iiii
GMAX=iiii GMX1=iiii

ERROR NUMBER 2 IN SUBROUTINE FLXIN

Significance: The member of energy groups (GMX1) and/or the number of mesh intervals (IMX1) defined in the data set FA.D1 do not correspond to the values (GMAX,IMAX) for the current problem.

Action taken: Execution is terminated.

OUTERS

Message: *****OUTER ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE OUTERS

Significance: The maximum number of outer iterations, NMAX, has been exceeded, and the problem has not converged.

Action taken: Execution of the remainder of the program, consisting of the output section, is performed as if the problem has converged.

PCOMP

Message: *****SEARCH CALCULATED A NEGATIVE aaaaaaaa VALUE
COMPOSITION SEARCH TERMINATED

ERROR NUMBER 1 IN SUBROUTINE PCOMP

where aaaaaaaa is a cross-section type:

'DIF COEF', 'REMOVAL', 'FISSION', 'NU FISS', 'CHI MTRX', or 'SCATTER'

Significance: In adding the modifier composition to the original composition on a search pass, the search calculated a negative value or values of some cross-section type or types (only the most recently calculated type being indicated). This messsage could not be encountered in module NUC002 unless the input flag to PCOMP were not set to zero before the call to PCOMP.

Action taken: The concentration search is terminated, without output of data sets.

PRTBC

Message: *****CALCULATION TYPE IS aaaaaaaa
BOUNDARY CONDITIONS INDICATE CALCULATION SHOULD BE INHOMOGENEOUS

ERROR NUMBER 1 IN SUBROUTINE PRTBC

where aaaaaaaa is the calculation type in PROBT

Significance: One or more inhomogeneous boundary or interface conditions have been defined, and the calculation type is not source. Either the boundary conditions or the calculation type is in error.

Action taken: Execution is terminated.

RBNDC

Message: *****DATA SET BC NOT FOUND

ERROR NUMBER 1 SUBROUTINE RBNDC

Significance: This data set would not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated

Message: *****LESS THAN TWO BOUNDARY CONDITIONS DEFINED. NBCDEF=iii

ERROR NUMBER 2 IN SUBROUTINE RBNDC

Significance: Both external boundary conditions, at least, must be defined and two boundary conditions are not defined in data set BC. NBCDEF is the number of boundary conditions defined.

Action taken: Execution is terminated.

RCDEP

Message: *****DATA SET A.DIF1D NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RCDEP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****DATA SET SP.CICN NOT FOUND

ERROR NUMBER 2 IN SUBROUTINE RCDEP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NO END-OF-FILE FOUND FOR FIRST FILE OF XS.C.MIN

ERROR NUMBER 2 IN SUBROUTINE RCOMP

Significance: The end of the first file of this data set was not found, and the second file thus could not be read.

Action taken: Execution is terminated.

Message: *****DATA SET XS.DELAY NOT FOUND

ERROR NUMBER 3 IN SUBROUTINE RCOMP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NUMBER OF FAMILIES DEFINED FOR DELAYED DATA IS ZERO

ERROR NUMBER 4 IN SUBROUTINE RCOMP

Significance: The number of families NFAM defined in XS.C.MIN was zero, and the kinetics option indicates that delayed data is desired for an α-calculation.

Action taken: Execution is terminated.

RDIST

Message: *****DATA SET XS.DELAY NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RDIST

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NUMBER OF FAMILIES DEFINED FOR DELAYED DATA IS ZERO

ERROR NUMBER 2 IN SUBROUTINE RDIST

Significance: The number of families NFAM defined in XS.C.MIN was zero, and the kinetics option indicated delayed neutron source data is required.

Action taken: Execution is terminated.

RFLUX

Message: *****DATA SET FR.D1 NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RFLUX

or

*****DATA SET FA.D1 NOT FOUND

ERROR NUMBER 2 IN SUBROUTINE RFLUX

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****FLUXES DIMENSIONED IMPROPERLY ON FR.D1 OR FA.D1
IMX1=iiii GMX1=iiii

ERROR NUMBER 3 IN SUBROUTINE RFLUX

Significance: The number of mesh intervals (IMX1) and/or of groups (GMX1) defined on this data set are not the same as the corresponding numbers defined on data sets GEOM and XS.C.MIN.

Action taken: Execution is terminated.

RGEOM

Message: *****DATA SET GEOM NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RGEOM

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****GEOMETRY TYPE CANNOT BE IDENTIFIED. NGEOM=iiii

ERROR NUMBER 2 IN SUBROUTINE RGEOM

Significance: The geometry type number read from data set GEOM is not 1, 2, or 3. It may be a two-dimensional type (4, 5, 6, 7, or 8), or may be meaningless. The geometry type selected in data set A.NIP (card type 3, cols. 13-18) should be checked.

Action taken: Execution is terminated.

RGROUP

Message: *****TRANSPORT CROSS-SECTION IS ZERO DIFFUSION COEFFICIENT IS SET TO 1.0 E+15
FOR COMPOSITION aaaaaaaa, GROUP iii

Significance: A transport cross section of zero for this composition and group was read from data set XS.C.MIN, and hence the diffusion coefficient, $D_c^g = \frac{1}{3(\sigma_{TR})_c^g}$, could not be calculated. The diffusion coefficient is set to 1.0×10^{15}.

Action taken: Processing continues. If this composition is found to be defined as black for this group the diffusion coefficient will be set to zero, as it should be.

RSORC

Message: *****EXTERNAL FIXED SOURCE DATA DIMENSIONED IMPROPERLY ON ES.D1D
IMX1=iiii GMX1=iiii

ERROR NUMBER 1 IN SUBROUTINE RSORC

Significance: The number of intervals IMX1 and/or the number of groups GMX1 defined on this data set do not agree with those defined for the problem in data sets GEOM and XS.C.MIN.

Action taken: Execution is terminated.

Message: *****NO END-OF-FILE FOUND FOR THE FIRST FILE OF XS.DELAY

ERROR NUMBER 2 IN SUBROUTINE RSORC

Significance: The end-of-file for the first file of XS.DELAY was not found and the precursor constants could not be read for the kinetics option.

Action taken: Execution is terminated.

Message: *****FISSION SPECTRUM WEIGHTING USING A NON-FISSIONABLE COMPOSITION
CAUSES THE EXTERNAL SOURCE DATA AFFECTED TO BE SET TO ZERO
INTERVALS AFFECTED ARE
I = iiii
⋮
I = iiii

Significance: As indicated, fission-spectrum weighting has been specified for the indicated intervals, each of which contains a non-fissionable composition.

Action taken: FIXSRC(I,G) is set to zero for G=1, ... , GMAX.

RVEL

Message: *****NUMBER OF GROUPS DEFINED IN XS.ISO OR XS.ISO2
DOES NOT AGREE WITH NUMBER DEFINED IN GEOM
DSRN=iii GMAX=iii NGROUP=iii

ERROR NUMBER 1 IN SUBROUTINE RVEL

Significance: As stated, the number of groups defined in the two data sets do not agree. DSRN is the data set reference number associated with XS.ISO or XS.ISO2.

Action taken: Execution is terminated.

SOUTER

Message: *****SOURCE ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE SOUTER

Significance: The maximum number of iterations, NMAX, has been exceeded, and the problem has not converged.

Action taken: Execution of the remainder of the program, consisting of the output section, is performed as if the problem had converged.

SWEEP (entry in ISWEEP)

Message: *****UPSCATTERING ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE SWEEP

Significance: The maximum number of upscatter iterations LMAX has been exceeded and the iterations have not converged.

Action taken: Execution is continued as though iterations had converged.

9. DIF1D (NUC002) Timing Information

Times for some representative problem runs are given in the table below. For each problem the calculation type, number of groups, number of regions, number of compositions present and number used, type of fission spectrum χ present, number of mesh intervals, and time in minutes are given.

DIF1D (NUC002) Sample Problem Time

Calculation Type	Number of Groups	Number of Regions	Number of Compositions Present	Number of Compositions Used	Type of χ	Number of Mesh Intervals	Time (min.)
real k_{eff}	6	4	4	3	vector	40	.17
real k_{eff}	22	4	4	4	vector	40	.36
real k_{eff}	22	4	10	3	matrix	40	.43
real k_{eff}	22	4	4	3	matrix	40	.37
real k_{eff}	22	4	4	3	matrix	80	.60
real k_{eff}	22	4	4	3	matrix	120	.84
real k_{eff} (no printed output)	22	4	4	3	matrix	80	.47
adjoint k_{eff}	22	4	4	3	matrix	80	.56
both real and adjoint k_{eff}	22	4	4	3	matrix	80	1.04
real source	22	4	4	3	matrix	80	1.08
adjoint source	22	4	4	3	matrix	80	.93
adjoint k_{eff}	22	4	4	4	vector	80	.61
real source	22	4	4	4	vector	80	.56
adjoint source	22	4	4	4	vector	80	.57

10. DIF1D (NUC002) Sample Output

The following pages represent output from a sample DIF1D run. For a detailed explanation of the output, see section 2.f.

The module begins by printing a message that it has been entered. Page 1 consists of a list of module-dependent and module-independent options selected, from data sets A.DIF1D and SP.C1CN. Pages 2 and 3 describe the reactor structure, as defined in data set GEOM. Page 4 describes the boundary conditions defined in data set BC.

Page 5 shows the macroscopic group cross section data defined in data set XS.C.MIN for composition CP1. Since this composition is fissionable and has a χ-matrix, the χ-matrix is printed beginning on page 6. The scattering cross section begins on page 7. (Note that more than one page of printed output corresponds to each page number of these tables.) A similar set of tables is displayed for each composition defined on XS.C.MIN. Following this data, the amount of POINTR storage required for the input section appears, and then the amount needed for the calculation section.

The iteration history for this problem begins on page 16 and is concluded on page 17. It is followed by a print of the amount of storage needed for the output section.

The output section begins on page 18 with a table of the real normalized fluxes by interval and group. This table continues on pages 19, 20, and 21. The real normalized regional flux integrals by region and group appear on page 22.

Following this, but not shown here, are tables of the regional flux averages by group; fission source density by interval and group; and fission source integrals by interval and group, if desired.

If the output option is 3, average power by interval is printed as on page 28, followed by power and average power by region as on page 29.

The contributions to the balance terms by region and group are then printed beginning on page 30. The terms printed are the leakage including DB^2 term (shown); currents across region interfaces (shown); and removal plus

α/v, scattering source, fission source, average fission source, external source (if present), and total balance, all in a format like that shown for the leakage. On page 37 the balance sums by region over groups appear, and on page 38 the balance sums by group over regions.

For a 'BOTH' type problem, the iteration and output sections are then repeated for the adjoint calculation, and finally a message is printed that the module's execution is complete, along with the time for the problem.

Note that for this problem the amount of POINTR storage specified on the next run can be 9035 R*8 words, the amount needed for the largest section, rather than 30000 words.

```
                    ***  ENTERING MODULE DIF1D  (NUC002)  ***


ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

                K-EFF  CALCULATION          BOTH    SOLUTION

    NO KINETICS OPTION

                OUTER ITERATION CONTROL DATA

    MAXIMUM NUMBER OF OUTER ITERATIONS =   90

    CHEBYCHEV EXTRAPOLATION ON OUTER ITERATIONS

    DO    5 UNEXTRAPOLATED OUTER ITERATIONS BEFORE BEGINNING TO EXTRAPOLATE

    DO    3 EXTRAPOLATED ITERATIONS BEFORE TESTING FOR PAYOFF

    DO A MAXIMUM OF    8 ITERATIONS IN AN EXTRAPOLATION SEQUENCE

    DO AT LEAST    3 UNEXTRAPOLATED OUTER ITERATIONS
    BEFORE BEGINNING ANOTHER EXTRAPOLATION SEQUENCE

    INTERVAL SHIFT FACTOR FOR CHEBYCHEV EXTRAPOLATION = 0.0

                OUTER ITERATION CONVERGENCE CRITERIA

    DIFFERENCE IN SUCCESSIVE VALUES OF K-EFFECTIVE = 1.000000D-07

    SUM OF ABSOLUTE VALUE OF FLUX DIFFERENCE/SUM OF FLUX = 1.000000D-07

    DIFFERENCE IN UPPER AND LOWER FLUX BOUNDS LAMBDA = 1.000000D-07

    CORE STORAGE ALLOCATED FOR POINTR COMMON ARRAY       =    30000

    BULK (LCS) STORAGE ALLOCATED FOR POINTR COMMON ARRAY =        0

                          GEOMETRY DATA


    GEOMETRY TYPE  FINITE  RECTANGULAR SLAB

          MESH        MESH          MESH        MESH
       INTERVAL    INTERVAL      INTERVAL    INTERVAL
       BOUNDARY      NUMBER        LENGTH   VOLUME PER   REGION  COMPOSITION
                                            UNIT HEIGHT

         0.0
                       1          2.500        2.500      REG1       CP1
        2.500
                       2          2.500        2.500      REG1       CP1
        5.000
                       3          2.500        2.500      REG1       CP1
        7.500
                       4          2.500        2.500      REG1       CP1
       10.000
                       5          5.000        5.000      REG2       CP2
       15.000
```

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

MESH INTERVAL BOUNDARY	MESH INTERVAL NUMBER	MESH INTERVAL LENGTH	MESH INTERVAL VOLUME PER UNIT HEIGHT	REGION	COMPOSITION
	6	5.000	5.000	REG2	CP2
20.000					
	7	5.000	5.000	REG2	CP2
25.000					
	8	5.000	5.000	REG2	CP2
30.000					
	9	5.000	5.000	REG2	CP2
35.000					
	10	5.000	5.000	REG2	CP2
40.000					
	11	5.000	5.000	REG2	CP2
45.000					
	12	5.000	5.000	REG2	CP2
50.000					
	13	5.000	5.000	REG2	CP2
55.000					
	14	5.000	5.000	REG2	CP2
60.000					
	15	2.500	2.500	REG3	CP3
62.500					
	16	2.500	2.500	REG3	CP3
65.000					
	17	2.500	2.500	REG3	CP3
67.500					
	18	2.500	2.500	REG3	CP3
70.000					
	19	2.500	2.500	REG3	CP3
72.500					
	20	2.500	2.500	REG3	CP3
75.000					
	21	2.500	2.500	REG3	CP3
77.500					
	22	2.500	2.500	REGX	CP3
80.000					
	23	2.000	2.000	REG4	CP4
82.000					
	24	2.000	2.000	REG4	CP4
84.000					
	25	2.000	2.000	REG4	CP4
86.000					
	26	2.000	2.000	REG4	CP4
88.000					
	27	2.000	2.000	REG4	CP4
90.000					
	28	2.000	2.000	REG4	CP4
92.000					
	29	2.000	2.000	REG4	CP4
94.000					
	30	2.000	2.000	REG4	CP4
96.000					
	31	2.000	2.000	REG4	CP4
98.000					
	32	2.000	2.000	REG4	CP4
100.000					
	33	2.000	2.000	REG4	CP4
102.000					
	34	2.000	2.000	REG4	CP4
104.000					
	35	2.000	2.000	REG4	CP4
106.000					
	36	2.000	2.000	REG4	CP4
108.000					
	37	2.000	2.000	REG4	CP4
110.000					

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

REGION	COMPOSITION	TRANSVERSE BUCKLING (B**2)	ACTUAL TRANSVERSE HALF-HEIGHT	TRANSVERSE EXTRAPOLATION DISTANCE	REGION VOLUME
REG1	CP1	2.039174D-04	9.000000D 01	2.000000D 01	1.800000D 03
REG2	CP2	2.039174D-04	9.000000D 01	2.000000D 01	9.000000D 03
REG3	CP3	2.039174D-04	9.000000D 01	2.000000D 01	3.150000D 03
REGX	CP3	2.039174D-04	9.000000D 01	2.000000D 01	4.500000D 02
REG4	CP4	2.039174D-04	9.000000D 01	2.000000D 01	5.400000D 03

BOUNDARY AND INTERFACE CONDITIONS

MESH INTERVAL BOUNDARY	MESH LINE NUMBER	BOUNDARY TYPE	INITIAL GROUP	FINAL GROUP	COEF1	COEF2	COEF3	COEF4
0.0	1	1	1	22	1.000	0.0	0.0	0.0
110.000	-38	1	1	22	0.0	1.000	0.0	0.0

BOUNDARY AND INTERFACE CONDITION TYPES

TYPE	DESCRIPTION AND FORMULA	
1	GENERAL BOUNDARY	COEF1*Ø' + COEF2*Ø = COEF3
2	GENERAL BOUNDARY	COEF1*D*Ø' + COEF2*Ø = COEF3*D, WHERE D IS THE DIFFUSION COEFFICIENT
3	GENERAL INTERFACE	Ø(I+1) = COEF1*Ø(I) + COEF2*Ø'(I) Ø'(I+1) = COEF3*Ø(I) + COEF4*Ø'(I)

A POSITIVE MESH LINE NUMBER INDICATES THAT THE CONDITION APPLIES ON THE RIGHT OF THE MESH LINE
A NEGATIVE MESH LINE NUMBER INDICATES THAT THE CONDITION APPLIES ON THE LEFT OF THE MESH LINE

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

MACROSCOPIC GROUP CROSS SECTION DATA FOR COMPOSITION CP1

GROUP	DIF COEF	REMOVAL	NU*FISSION	FISSION
1	3.04696D 00	1.10523D-01	3.94997D-02	1.03281D-02
2	2.99173D 00	1.09579D-01	3.73940D-02	1.02390D-02
3	2.91843D 00	1.06041D-01	3.48609D-02	9.96030D-03
4	2.72956D 00	1.01713D-01	2.68601D-02	7.98047D-03
5	2.63468D 00	1.00138D-01	1.92755D-02	5.93598D-03
6	2.63025D 00	1.00397D-01	1.79680D-02	5.71743D-03
7	2.39421D 00	1.05166D-01	1.72787D-02	5.66439D-03
8	2.29946D 00	1.07571D-01	1.69064D-02	5.69430D-03
9	2.38075D 00	9.82329D-02	1.61712D-02	5.58171D-03
10	2.38482D 00	9.38972D-02	1.54025D-02	5.43531D-03
11	2.25738D 00	9.65276D-02	1.53481D-02	5.52512D-03
12	2.15966D 00	9.52113D-02	1.58311D-02	5.80199D-03
13	2.15237D 00	9.36226D-02	1.52749D-02	5.68889D-03
14	2.21882D 00	8.74926D-02	1.34074D-02	5.06588D-03
15	2.13696D 00	9.15696D-02	1.05654D-02	4.04385D-03
16	2.19726D 00	8.11522D-02	5.49606D-03	2.12737D-03
17	2.29593D 00	3.90259D-02	6.54662D-04	2.58498D-04
18	1.93915D 00	2.29295D-02	1.04277D-04	4.20387D-05
19	1.61510D 00	1.92949D-02	7.85863D-05	3.18972D-05
20	1.36639D 00	1.19253D-02	8.79360D-05	3.59455D-05
21	1.06054D 00	1.30463D-02	1.11935D-04	4.59532D-05
22	8.09086D-01	6.37134D-03	1.59158D-04	6.54485D-05

FISSION SPECTRUM FRACTIONS FOR COMPOSITION CP1

GROUP	FROM GROUP									
1	1	2.50235D-03	2	2.50235D-03	3	2.50235D-03	4	2.50235D-03	5	2.50235D-03
	6	2.50235D-03	7	2.50235D-03	8	2.50235D-03	9	2.50235D-03	10	2.50235D-03
	11	2.50235D-03	12	2.50235D-03	13	2.50235D-03	14	2.50235D-03	15	2.50235D-03
	16	2.50235D-03	17	2.50235D-03	18	2.50235D-03	19	2.50235D-03	20	2.50235D-03
	21	2.50235D-03	22	2.50235D-03						
2	1	4.67673D-03	2	4.67673D-03	3	4.67673D-03	4	4.67673D-03	5	4.67673D-03
	6	4.67673D-03	7	4.67673D-03	8	4.67673D-03	9	4.67673D-03	10	4.67673D-03
	11	4.67673D-03	12	4.67673D-03	13	4.67673D-03	14	4.67673D-03	15	4.67673D-03
	16	4.67673D-03	17	4.67673D-03	18	4.67673D-03	19	4.67673D-03	20	4.67673D-03
	21	4.67673D-03	22	4.67673D-03						
3	1	7.94990D-03	2	7.94990D-03	3	7.94990D-03	4	7.94990D-03	5	7.94990D-03
	6	7.94990D-03	7	7.94990D-03	8	7.94990D-03	9	7.94990D-03	10	7.94990D-03
	11	7.94990D-03	12	7.94990D-03	13	7.94990D-03	14	7.94990D-03	15	7.94990D-03
	16	7.94990D-03	17	7.94990D-03	18	7.94990D-03	19	7.94990D-03	20	7.94990D-03
	21	7.94990D-03	22	7.94990D-03						

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

4	1	1.24273D-02	2	1.24273D-02	3	1.24273D-02	4	1.24273D-02	5	1.24273D-02
	6	1.24273D-02	7	1.24273D-02	8	1.24273D-02	9	1.24273D-02	10	1.24273D-02
	11	1.24273D-02	12	1.24273D-02	13	1.24273D-02	14	1.24273D-02	15	1.24273D-02
	16	1.24273D-02	17	1.24273D-02	18	1.24273D-02	19	1.24273D-02	20	1.24273D-02
	21	1.24273D-02	22	1.24273D-02						
5	1	1.80392D-02	2	1.80392D-02	3	1.80392D-02	4	1.80392D-02	5	1.80392D-02
	6	1.80392D-02	7	1.80392D-02	8	1.80392D-02	9	1.80392D-02	10	1.80392D-02
	11	1.80392D-02	12	1.80392D-02	13	1.80392D-02	14	1.80392D-02	15	1.80392D-02
	16	1.80392D-02	17	1.80392D-02	18	1.80392D-02	19	1.80392D-02	20	1.80392D-02
	21	1.80392D-02	22	1.80392D-02						
6	1	2.45254D-02	2	2.45254D-02	3	2.45254D-02	4	2.45254D-02	5	2.45254D-02
	6	2.45254D-02	7	2.45254D-02	8	2.45254D-02	9	2.45254D-02	10	2.45254D-02
	11	2.45254D-02	12	2.45254D-02	13	2.45254D-02	14	2.45254D-02	15	2.45254D-02
	16	2.45254D-02	17	2.45254D-02	18	2.45254D-02	19	2.45254D-02	20	2.45254D-02
	21	2.45254D-02	22	2.45254D-02						
7	1	3.14694D-02	2	3.14694D-02	3	3.14694D-02	4	3.14694D-02	5	3.14694D-02
	6	3.14694D-02	7	3.14694D-02	8	3.14694D-02	9	3.14694D-02	10	3.14694D-02
	11	3.14694D-02	12	3.14694D-02	13	3.14694D-02	14	3.14694D-02	15	3.14694D-02
	16	3.14694D-02	17	3.14694D-02	18	3.14694D-02	19	3.14694D-02	20	3.14694D-02
	21	3.14694D-02	22	3.14694D-02						
8	1	3.83672D-02	2	3.83672D-02	3	3.83672D-02	4	3.83672D-02	5	3.83672D-02
	6	3.83672D-02	7	3.83672D-02	8	3.83672D-02	9	3.83672D-02	10	3.83672D-02
	11	3.83672D-02	12	3.83672D-02	13	3.83672D-02	14	3.83672D-02	15	3.83672D-02
	16	3.83672D-02	17	3.83672D-02	18	3.83672D-02	19	3.83672D-02	20	3.83672D-02
	21	3.83672D-02	22	3.83672D-02						
9	1	4.47114D-02	2	4.47114D-02	3	4.47114D-02	4	4.47114D-02	5	4.47114D-02
	6	4.47114D-02	7	4.47114D-02	8	4.47114D-02	9	4.47114D-02	10	4.47114D-02
	11	4.47114D-02	12	4.47114D-02	13	4.47114D-02	14	4.47114D-02	15	4.47114D-02
	16	4.47114D-02	17	4.47114D-02	18	4.47114D-02	19	4.47114D-02	20	4.47114D-02
	21	4.47114D-02	22	4.47114D-02						
.				.					.	
.				.					.	
.				.					.	
22	1	3.85481D-03	2	3.85481D-03	3	3.85481D-03	4	3.85481D-03	5	3.85481D-03
	6	3.85481D-03	7	3.85481D-03	8	3.85481D-03	9	3.85481D-03	10	3.85481D-03
	11	3.85481D-03	12	3.85481D-03	13	3.85481D-03	14	3.85481D-03	15	3.85481D-03
	16	3.85481D-03	17	3.85481D-03	18	3.85481D-03	19	3.85481D-03	20	3.85481D-03
	21	3.85481D-03	22	3.85481D-03						

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

TRANSFER CROSS SECTION FOR COMPOSITION CP1

GROUP	FROM GROUP										
1	1	0.0									
2	1	8.02179D-03	2	0.0							
3	1	3.05669D-04	2	8.28545D-03	3	0.0					
4	1	3.71674D-04	2	4.36423D-04	3	8.97139D-03	4	0.0			
5	1	5.61980D-04	2	5.56560D-04	3	6.15641D-04	4	1.18150D-02	5	0.0	
6	1	8.18658D-04	2	8.20808D-04	3	8.11758D-04	4	9.01528D-04	5	1.21219D-02	
	6	0.0									
7	1	1.14377D-03	2	1.15467D-03	3	1.15724D-03	4	1.14754D-03	5	1.25361D-03	
	6	1.31557D-02	7	0.0							
8	1	1.52992D-03	2	1.54860D-03	3	1.56876D-03	4	1.57672D-03	5	1.55197D-03	
	6	1.61066D-03	7	1.73847D-02	8	0.0					
9	1	1.96197D-03	2	1.98268D-03	3	2.02775D-03	4	2.06456D-03	5	2.04521D-03	
	6	1.93303D-03	7	1.97405D-03	8	1.75366D-02	9	0.0			
10	1	2.42204D-03	2	2.42975D-03	3	2.50757D-03	4	2.58709D-03	5	2.57798D-03	
	6	2.49629D-03	7	2.31905D-03	8	1.17880D-02	9	2.00848D-02	10	0.0	
11	1	2.89522D-03	2	2.86060D-03	3	2.97726D-03	4	3.11441D-03	5	3.12217D-03	
	6	3.19253D-03	7	2.97889D-03	8	1.41775D-03	9	7.17346D-03	10	2.02693D-02	
	11	0.0									
12	1	3.37307D-03	2	3.24990D-03	3	3.40559D-03	4	3.61381D-03	5	3.64621D-03	
	6	3.64709D-03	7	3.84808D-03	8	5.77080D-04	9	5.10565D-03	10	8.06344D-03	
	11	2.46571D-02	12	0.0							
13	1	3.85269D-03	2	3.58173D-03	3	3.76484D-03	4	4.05321D-03	5	4.11777D-03	
	6	4.23114D-03	7	4.36557D-03	8	1.12793D-03	9	6.46930D-04	10	1.24757D-02	
	11	3.95353D-03	12	2.36882D-02	13	0.0					
14	1	9.51336D-02	2	3.85302D-03	3	4.03377D-03	4	4.40465D-03	5	4.50705D-03	
	6	4.59185D-03	7	4.75593D-03	8	3.25816D-03	9	9.35228D-04	10	6.78344D-04	
	11	1.72407D-02	12	2.86644D-03	13	2.65535D-02	14	0.0			

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

15	2	8.89569D-02	3	4.19995D-03	4	4.64740D-03	5	4.79018D-03	6	4.70909D-03
	7	5.13704D-03	8	4.09127D-03	9	1.68159D-03	10	1.42659D-03	11	6.96587D-03
	12	1.15906D-02	13	3.74891D-03	14	2.38562D-02	15	0.0		
16	3	7.24038D-02	4	4.77007D-03	5	4.95195D-03	6	4.89809D-03	7	5.09925D-03
	8	2.46845D-03	9	3.99338D-03	10	1.80805D-03	11	1.44943D-03	12	1.62311D-02
	13	2.09681D-03	14	3.29601D-03	15	2.95953D-02	16	0.0		
17	4	4.91240D-02	5	1.90309D-02	6	1.91727D-02	7	1.94553D-02	8	1.79606D-02
	9	1.41435D-02	10	1.14991D-02	11	8.38353D-03	12	9.32257D-03	13	2.65817D-02
	14	2.73068D-02	15	1.19769D-02	16	3.43389D-02	17	0.0		
18	5	2.88454D-02	6	1.43160D-02	7	1.48164D-02	8	1.72819D-02	9	1.70762D-02
	10	1.06751D-02	11	1.15159D-02	12	8.27281D-03	13	1.23933D-02	14	1.14119D-02
	15	2.90065D-02	16	2.45333D-02	17	2.02420D-02	18	0.0		
19	6	1.54276D-02	7	8.71423D-03	8	1.20999D-02	9	1.07667D-02	10	9.55560D-03
	11	7.78500D-03	12	7.50824D-03	13	7.16429D-03	14	1.23886D-02	15	8.07071D-03
	16	1.26935D-02	17	6.01310D-03	18	1.96195D-02	19	0.0		
20	7	7.50540D-03	8	9.21233D-03	9	8.76082D-03	10	9.01028D-03	11	6.36279D-03
	12	7.60667D-03	13	6.78027D-03	14	3.19979D-03	15	7.28669D-03	16	5.55074D-03
	17	8.79007D-03	18	1.28010D-03	19	1.75670D-02	20	0.0		
21	8	2.01673D-03	9	1.35940D-03	10	1.79988D-03	11	1.64871D-03	12	1.41581D-03
	13	1.63691D-03	14	1.91967D-04	15	6.74356D-04	16	5.91587D-04	17	1.79874D-03
	18	3.33853D-04	19	1.05709D-04	20	9.95382D-03	21	0.0		
22	9	2.02756D-04	10	3.86845D-04	11	3.36939D-04	12	2.55891D-04	13	3.07514D-04
	14	2.74992D-05	15	8.21271D-06	16	1.45096D-04	17	3.17029D-04	18	7.22624D-05
	19	2.77321D-05	20	5.15637D-05	21	9.91840D-03	22	0.0		

SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR INPUT SECTION = 8109

SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR INPUT SECTION = 0

SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION = 9030

SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION = 0

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

ITERATION HISTORY REAL CALCULATION

ITER NO P	KEFF SIGMA	DIF KEFF ENPM1	LAM UPPER SIGMA BAR	LAM LOWER LAM RATIO	DIF LAM ALPHA	DIF FLUX BETA	NGL FLUX
1	1.3312415D 01	1.2312415D 01	2.3540283D 01	7.6222890D-06	2.3540276D 01	1.1139682D 00	0.0
-4	0.0	0.0	1.1139682D 00	2.3540276D 01	0.0	0.0	
2	9.5406745D-01	-1.2358348D 01	9.4280023D-02	2.5451929D-02	6.8828094D-02	1.3790418D 01	0.0
-3	0.0	0.0	1.2379544D 01	2.9238440D-03	0.0	0.0	
3	1.0266590D 00	7.2591575D-02	1.1710132D 00	7.2262637D-01	4.4838685D-01	7.0464440D-02	0.0
-2	0.0	0.0	5.1096668D-03	6.5145905D 00	0.0	0.0	
4	1.0437922D 00	1.7133162D-02	1.0437901D 00	8.8353589D-01	1.6025422D-01	2.0977883D-02	0.0
-1	0.0	0.0	2.9770879D-01	3.5740170D-01	0.0	0.0	
5	1.0482023D 00	4.4101519D-03	1.0129600D 00	9.5766711D-01	5.5292858D-02	7.7475511D-03	0.0
-1	0.0	0.0	3.6931997D-01	3.4503214D-01	0.0	0.0	
6	1.0494278D 00	1.2254150D-03	1.0049274D 00	9.8377120D-01	2.1156152D-02	3.4054610D-03	0.0
-1	0.0	0.0	4.3955322D-01	3.8261998D-01	0.0	0.0	
7	1.0497878D 00	3.6000321D-04	1.0027479D 00	9.9309967D-01	9.6482716D-03	1.6764449D-03	0.0
-1	0.0	0.0	4.9228134D-01	4.5605040D-01	0.0	0.0	
8	1.0498996D 00	1.1188188D-04	1.0016263D 00	9.9678619D-01	4.8401281D-03	8.7322649D-04	0.0
0	5.2087992D-01	0.0	5.2087992D-01	5.0165753D-01	0.0	0.0	
9	1.0499370D 00	3.7327670D-05	1.0009200D 00	9.9840535D-01	2.5146682D-03	4.6403887D-04	0.0
1	5.2087992D-01	1.0000000D 00	5.3140723D-01	5.1954580D-01	1.3521553D 00	0.0	
10	1.0499554D 00	1.8403242D-05	1.0003777D 00	9.9946236D-01	9.1532447D-04	1.7307711D-04	0.0
2	5.2087992D-01	3.7297977D-01	3.7297977D-01	3.6399414D-01	1.4415404D 00	6.6105655D-02	
11	1.0499604D 00	5.0346310D-06	1.0000933D 00	9.9988976D-01	2.0352106D-04	3.9370603D-05	0.0
-3	0.0	0.0	2.2747435D-01	2.2234854D-01	1.3983755D 00	3.4182661D-02	
12	1.0499607D 00	2.9609336D-07	1.0000211D 00	9.9996965D-01	5.1490947D-05	9.8625609D-06	0.0
-2	0.0	0.0	2.5050571D-01	2.5300058D-01	0.0	0.0	
13	1.0499608D 00	1.1072028D-07	1.0000113D 00	9.9998336D-01	2.7970969D-05	5.3536863D-06	0.0
-1	0.0	0.0	5.4282923D-01	5.4322110D-01	0.0	0.0	
14	1.0499609D 00	6.6306515D-08	1.0000061D 00	9.9990095D-01	1.5175210D-05	2.9009466D-06	0.0
0	5.4185965D-01	0.0	5.4185965D-01	5.4253429D-01	0.0	0.0	

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

ITERATION HISTORY REAL CALCULATION

ITER NO P	KEFF SIGMA	DIF KEFF ENPM1	LAM UPPER SIGMA BAR	LAM LOWER LAM RATIO	DIF LAM ALPHA	DIF FLUX BETA	NGL FLUX
15	1.0499609D 00	3.7746677D-08	1.0000033D 00	9.9999509D-01	8.2267919D-06	1.5715661D-06	0.0
1	5.4185965D-01	1.0000000D 00	5.4174253D-01	5.4212048D-01	1.3716101D 00	0.0	
16	1.0499609D 00	2.8765577D-08	1.0000012D 00	9.9999818D-01	3.0574444D-06	5.8214596D-07	0.0
2	5.4185965D-01	3.7042409D-01	3.7042409D-01	3.7164479D-01	1.4733398D 00	7.4168112D-02	
17	1.0499610D 00	1.3912246D-08	1.0000003D 00	9.9999965D-01	6.0922447D-07	1.1520760D-07	0.0
3	5.4185965D-01	7.3307508D-02	1.9790156D-01	1.9925938D-01	1.4244339D 00	3.8512246D-02	
18	1.0499610D 00	2.8686946D-09	1.0000000D 00	9.9999993D-01	1.1689718D-07	2.2103556D-08	0.0
4	5.4185965D-01	1.4064668D-02	1.9185849D-01	1.9187867D-01	1.4226152D 00	3.7186335D-02	
19	1.0499610D 00	5.4421823D-10	1.0000000D 00	9.9999999D-01	2.2347737D-08	4.2294162D-09	0.0

CONVERGED KEFF = 1.0499610D 00

SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR OUTPUT SECTION = 8639

SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR OUTPUT SECTION = 0

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

REAL NORMALIZED FLUXES BY MESH INTERVAL AND GROUP

MESH INTERVAL NUMBER	MESH INTERVAL CENTER	GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5	GROUP 6
1	1.250	1.254892D-06	2.584730D-06	4.757826D-06	7.929763D-06	1.240385D-05	1.761363D-05
2	3.750	1.457273D-06	2.999349D-06	5.520018D-06	9.221223D-06	1.443397D-05	2.049667D-05
3	6.250	1.900300D-06	3.905649D-06	7.184848D-06	1.204666D-05	1.887525D-05	2.680086D-05
4	8.750	2.668091D-06	5.472696D-06	1.006015D-05	1.693865D-05	2.656423D-05	3.770670D-05
5	12.500	4.534250D-06	9.273906D-06	1.702007D-05	2.881959D-05	4.520815D-05	6.410168D-05
6	17.500	5.753162D-06	1.177715D-05	2.161828D-05	3.659818D-05	5.744985D-05	8.149908D-05
7	22.500	6.375092D-06	1.306036D-05	2.398079D-05	4.057218D-05	6.371617D-05	9.042760D-05
8	27.500	6.665809D-06	1.366200D-05	2.509017D-05	4.243165D-05	6.665159D-05	9.461718D-05
9	32.500	6.727164D-06	1.379064D-05	2.532877D-05	4.282646D-05	6.727860D-05	9.551854D-05
10	37.500	6.596746D-06	1.352405D-05	2.483980D-05	4.199735D-05	6.597774D-05	9.367461D-05
11	42.500	6.285702D-06	1.288546D-05	2.366623D-05	4.001586D-05	6.286241D-05	8.924784D-05
12	47.500	5.787468D-06	1.186137D-05	2.178340D-05	3.684007D-05	5.786619D-05	8.214378D-05
13	52.500	5.065201D-06	1.037642D-05	1.905336D-05	3.223500D-05	5.061898D-05	7.183745D-05
14	57.500	4.007380D-06	8.204651D-06	1.506426D-05	2.549171D-05	4.001025D-05	5.676225D-05
15	61.250	2.738779D-06	5.613322D-06	1.031502D-05	1.741512D-05	2.732180D-05	3.877285D-05
16	63.750	2.152359D-06	4.412607D-06	8.109850D-06	1.367924D-05	2.144627D-05	3.042874D-05
17	66.250	1.732227D-06	3.550434D-06	6.524289D-06	1.100083D-05	1.723493D-05	2.444398D-05
18	68.750	1.419250D-06	2.907359D-06	5.340385D-06	9.004499D-06	1.409798D-05	1.998504D-05
19	71.250	1.173902D-06	2.403230D-06	4.411489D-06	7.438682D-06	1.163995D-05	1.649212D-05
20	73.750	9.682961D-07	1.981382D-06	3.633641D-06	6.125355D-06	9.581147D-06	1.356901D-05
21	76.250	7.805693D-07	1.597477D-06	2.925307D-06	4.924529D-06	7.702195D-06	1.090527D-05
22	78.750	5.904630D-07	1.210749D-06	2.211428D-06	3.706205D-06	5.800381D-06	8.214968D-06
23	81.000	4.017742D-07	8.305327D-07	1.512538D-06	2.495901D-06	3.919338D-06	5.580874D-06
24	83.000	2.862704D-07	5.965972D-07	1.085541D-06	1.762967D-06	2.777647D-06	3.985185D-06
25	85.000	2.039618D-07	4.284952D-07	7.790465D-07	1.245503D-06	1.968354D-06	2.845095D-06
26	87.000	1.453041D-07	3.076999D-07	5.590277D-07	8.800554D-07	1.394679D-06	2.030615D-06
27	89.000	1.034955D-07	2.208916D-07	4.010594D-07	6.218720D-07	9.879937D-07	1.448776D-06
28	91.000	7.368795D-08	1.584934D-07	2.876062D-07	4.393799D-07	6.996291D-07	1.033086D-06
29	93.000	5.242509D-08	1.136180D-07	2.060743D-07	3.102938D-07	4.950664D-07	7.359883D-07
30	95.000	3.724131D-08	8.130943D-08	1.474142D-07	2.188721D-07	3.498117D-07	5.234583D-07
31	97.000	2.637594D-08	5.799818D-08	1.051149D-07	1.539823D-07	2.464687D-07	3.711293D-07
32	99.000	1.856906D-08	4.110736D-08	7.448163D-08	1.077340D-07	1.726585D-07	2.615187D-07
33	101.000	1.291541D-08	2.876892D-08	5.211454D-08	7.451210D-08	1.195381D-07	1.820339D-07
34	103.000	8.759389D-09	1.961757D-08	3.553122D-08	5.028924D-08	8.074263D-08	1.235336D-07
35	105.000	5.619087D-09	1.264018D-08	2.289120D-08	3.213378D-08	5.162333D-08	7.928279D-08
36	107.000	3.130360D-09	7.063876D-09	1.279167D-08	1.785200D-08	2.869059D-08	4.418176D-08
37	109.000	1.004622D-09	2.270705D-09	4.111781D-09	5.720991D-09	9.196171D-09	1.418142D-08

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

REAL NORMALIZED FLUXES BY MESH INTERVAL AND GROUP

MESH INTERVAL NUMBER	MESH INTERVAL CENTER	GROUP 7	GROUP 8	GROUP 9	GROUP 10	GROUP 11	GROUP 12
1	1.250	2.088116D-05	2.611531D-05	3.799986D-05	5.639687D-05	6.114804D-05	7.520512D-05
2	3.750	2.460542D-05	3.089696D-05	4.437883D-05	6.496456D-05	7.052874D-05	8.634583D-05
3	6.250	3.284139D-05	4.149587D-05	5.833061D-05	8.346148D-05	9.085685D-05	1.103675D-04
4	8.750	4.734185D-05	6.022112D-05	8.246723D-05	1.148096D-04	1.255306D-04	1.510100D-04
5	12.500	8.319431D-05	1.066656D-04	1.407300D-04	1.883692D-04	2.078125D-04	2.465963D-04
6	17.500	1.058816D-04	1.358922D-04	1.793232D-04	2.398843D-04	2.645378D-04	3.139106D-04
7	22.500	1.172436D-04	1.504669D-04	1.991556D-04	2.672797D-04	2.945082D-04	3.498798D-04
8	27.500	1.224985D-04	1.571867D-04	2.084584D-04	2.804075D-04	3.088426D-04	3.671989D-04
9	32.500	1.235762D-04	1.585536D-04	2.104697D-04	2.834441D-04	3.121357D-04	3.712618D-04
10	37.500	1.211658D-04	1.554554D-04	2.064096D-04	2.780705D-04	3.062092D-04	3.642546D-04
11	42.500	1.154650D-04	1.481438D-04	1.966413D-04	2.648182D-04	2.916364D-04	3.468782D-04
12	47.500	1.063447D-04	1.364489D-04	1.809472D-04	2.434184D-04	2.681265D-04	3.187956D-04
13	52.500	9.309085D-05	1.194380D-04	1.581465D-04	2.123939D-04	2.340490D-04	2.781120D-04
14	57.500	7.347980D-05	9.420255D-05	1.247697D-04	1.676681D-04	1.847779D-04	2.195812D-04
15	61.250	4.935949D-05	6.301030D-05	8.501630D-05	1.164505D-04	1.277334D-04	1.528098D-04
16	63.750	3.843746D-05	4.894671D-05	6.656188D-05	9.193409D-05	1.007060D-04	1.208523D-04
17	66.250	3.076524D-05	3.911556D-05	5.335395D-05	7.396169D-05	8.101309D-05	9.735884D-05
18	68.750	2.513924D-05	3.193833D-05	4.354666D-05	6.038414D-05	6.619336D-05	7.957009D-05
19	71.250	2.077605D-05	2.639128D-05	3.589744D-05	4.969637D-05	5.455536D-05	6.555697D-05
20	73.750	1.713523D-05	2.177344D-05	2.952865D-05	4.080543D-05	4.488780D-05	5.392200D-05
21	76.250	1.379786D-05	1.754325D-05	2.375851D-05	3.285895D-05	3.625241D-05	4.358079D-05
22	78.750	1.037240D-05	1.319626D-05	1.796916D-05	2.511044D-05	2.782342D-05	3.359197D-05
23	81.000	6.889636D-06	8.755254D-06	1.232624D-05	1.791068D-05	1.990002D-05	2.430632D-05
24	83.000	4.804150D-06	6.090594D-06	8.857808D-06	1.332333D-05	1.482744D-05	1.825724D-05
25	85.000	3.352466D-06	4.238155D-06	6.357366D-06	9.882722D-06	1.103154D-05	1.368766D-05
26	87.000	2.341078D-06	2.949938D-06	4.557273D-06	7.311105D-06	8.194749D-06	1.024245D-05
27	89.000	1.635794D-06	2.053729D-06	3.262927D-06	5.394751D-06	6.077282D-06	7.649384D-06
28	91.000	1.143470D-06	1.429912D-06	2.333123D-06	3.970261D-06	4.498323D-06	5.700363D-06
29	93.000	7.993791D-07	9.953683D-07	1.665588D-06	2.913386D-06	3.321635D-06	4.236704D-06
30	95.000	5.584884D-07	6.923055D-07	1.186334D-06	2.130048D-06	2.444635D-06	3.137593D-06
31	97.000	3.894117D-07	4.804974D-07	8.418714D-07	1.549225D-06	1.790075D-06	2.311116D-06
32	99.000	2.702174D-07	3.318898D-07	5.935054D-07	1.117358D-06	1.299700D-06	1.687283D-06
33	101.000	1.855135D-07	2.268444D-07	4.131650D-07	7.939991D-07	9.294161D-07	1.212619D-06
34	103.000	1.244111D-07	1.515138D-07	2.803508D-07	5.484130D-07	6.455789D-07	8.459428D-07
35	105.000	7.909140D-08	9.599432D-08	1.798826D-07	3.569320D-07	4.221290D-07	5.550664D-07
36	107.000	4.378193D-08	5.300747D-08	1.002181D-07	2.008628D-07	2.383435D-07	3.141692D-07
37	109.000	1.400462D-08	1.693347D-08	3.216306D-08	6.479880D-08	7.702384D-08	1.016563D-07

ONE-DIMENSICNAL DIFFUSION THEORY SAMPLE OUTPUT

REAL NORMALIZED FLUXES BY MESH INTERVAL AND GROUP

MESH INTERVAL NUMBER	MESH INTERVAL CENTER	GROUP 13	GROUP 14	GROUP 15	GROUP 16	GROUP 17	GROUP 18
1	1.250	8.805462D-05	1.191381D-04	1.247401D-04	1.705517D-04	1.340348D-03	3.433569D-03
2	3.750	1.004762D-04	1.342159D-04	1.402564D-04	1.890704D-04	1.429766D-03	3.552412D-03
3	6.250	1.271055D-04	1.661099D-04	1.730724D-04	2.276504D-04	1.611346D-03	3.786605D-03
4	8.750	1.717460D-04	2.184150D-04	2.268939D-04	2.893529D-04	1.889959D-03	4.127842D-03
5	12.500	2.756291D-04	3.361110D-04	3.481221D-04	4.236460D-04	2.469243D-03	4.786316D-03
6	17.500	3.506294D-04	4.267548D-04	4.419149D-04	5.351347D-04	3.021718D-03	5.538446D-03
7	22.500	3.913053D-04	4.779039D-04	4.950540D-04	6.009504D-04	3.381390D-03	6.085899D-03
8	27.500	4.110793D-04	5.033569D-04	5.216538D-04	6.347969D-04	3.580830D-03	6.411806D-03
9	32.500	4.158458D-04	5.099020D-04	5.285867D-04	6.441713D-04	3.643602D-03	6.519122D-03
10	37.500	4.080631D-04	5.005761D-04	5.189739D-04	6.327773D-04	3.584265D-03	6.416303D-03
11	42.500	3.885403D-04	4.764471D-04	4.939294D-04	6.020462D-04	3.410347D-03	6.112464D-03
12	47.500	3.569226D-04	4.371536D-04	4.531028D-04	5.517046D-04	3.123739D-03	5.616848D-03
13	52.500	3.111740D-04	3.804760D-04	3.942541D-04	4.795164D-04	2.721215D-03	4.941219D-03
14	57.500	2.458042D-04	3.009008D-04	3.117992D-04	3.803897D-04	2.195274D-03	4.106767D-03
15	61.250	1.725205D-04	2.155522D-04	2.235641D-04	2.792894D-04	1.705223D-03	3.400877D-03
16	63.750	1.370110D-04	1.727218D-04	1.792840D-04	2.266085D-04	1.431712D-03	2.968093D-03
17	66.250	1.106212D-04	1.400785D-04	1.455131D-04	1.852043D-04	1.201691D-03	2.573255D-03
18	68.750	9.048270D-05	1.147258D-04	1.192766D-04	1.523143D-04	1.008048D-03	2.216862D-03
19	71.250	7.455361D-05	9.448787D-05	9.835665D-05	1.257100D-04	8.440229D-04	1.896708D-03
20	73.750	6.133719D-05	7.771340D-05	8.108480D-05	1.036181D-04	7.035355D-04	1.609132D-03
21	76.250	4.967168D-05	6.311453D-05	6.617855D-05	8.463122D-05	5.813052D-04	1.349875D-03
22	78.750	3.856694D-05	4.963955D-05	5.261750D-05	6.762974D-05	4.728880D-04	1.114655D-03
23	81.000	2.849390D-05	3.818944D-05	4.131548D-05	5.387611D-05	3.882224D-04	9.293040D-04
24	83.000	2.178454D-05	3.028533D-05	3.331480D-05	4.402696D-05	3.267078D-04	7.936921D-04
25	85.000	1.660162D-05	2.387223D-05	2.668702D-05	3.575712D-05	2.738030D-04	6.757672D-04
26	87.000	1.261321D-05	1.871432D-05	2.124964D-05	2.887361D-05	2.284975D-04	5.733616D-04
27	89.000	9.554174D-06	1.459570D-05	1.682432D-05	2.318526D-05	1.898280D-04	4.844934D-04
28	91.000	7.214342D-06	1.132607D-05	1.324581D-05	1.851181D-05	1.568954D-04	4.073587D-04
29	93.000	5.428200D-06	8.741906D-06	1.036620D-05	1.468877D-05	1.288735D-04	3.403205D-04
30	95.000	4.066039D-06	6.705266D-06	8.056262D-06	1.156943D-05	1.050116D-04	2.818962D-04
31	97.000	3.026541D-06	5.101272D-06	6.204877D-06	9.024961D-06	8.463171D-05	2.307421D-04
32	99.000	2.230659D-06	3.835020D-06	4.717540D-06	6.943341D-06	6.712441D-05	1.856385D-04
33	101.000	1.616612D-06	2.828348D-06	3.514251D-06	5.227588D-06	5.194120D-05	1.454731D-04
34	103.000	1.135790D-06	2.016711D-06	2.527201D-06	3.793621D-06	3.858638D-05	1.092245D-04
35	105.000	7.494238D-07	1.346254D-06	1.698405D-06	2.567928D-06	2.660797D-05	7.594559D-05
36	107.000	4.258228D-07	7.711340D-07	9.773896D-07	1.485194D-06	1.558820D-05	4.474678D-05
37	109.000	1.380584D-07	2.510488D-07	3.189578D-07	4.859196D-07	5.133795D-06	1.477965D-05

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

REAL NORMALIZED FLUXES BY MESH INTERVAL AND GROUP

MESH INTERVAL NUMBER	MESH INTERVAL CENTER	GROUP 19	GROUP 20	GROUP 21	GROUP 22
1	1.250	6.080383D-03	1.366554D-02	1.119945D-02	1.421385D-02
2	3.750	6.214987D-03	1.383883D-02	1.124266D-02	1.406006D-02
3	6.250	6.478748D-03	1.417641D-02	1.132025D-02	1.374083D-02
4	8.750	6.859938D-03	1.465962D-02	1.141429D-02	1.323296D-02
5	12.500	7.579757D-03	1.554161D-02	1.153719D-02	1.217049D-02
6	17.500	8.465991D-03	1.668425D-02	1.184660D-02	1.137997D-02
7	22.500	9.149516D-03	1.759478D-02	1.213736D-02	1.091839D-02
8	27.500	9.570159D-03	1.814542D-02	1.229795D-02	1.060217D-02
9	32.500	9.704110D-03	1.826518D-02	1.225873D-02	1.031458D-02
10	37.500	9.547031D-03	1.792365D-02	1.198344D-02	9.990388D-03
11	42.500	9.106103D-03	1.712162D-02	1.146363D-02	9.606727D-03
12	47.500	8.397685D-03	1.588710D-02	1.071560D-02	9.178978D-03
13	52.500	7.449294D-03	1.427582D-02	9.779808D-03	8.761189D-03
14	57.500	6.306406D-03	1.237624D-02	8.722341D-03	8.451306D-03
15	61.250	5.361899D-03	1.082336D-02	7.906125D-03	8.414108D-03
16	63.750	4.762559D-03	9.796660D-03	7.324090D-03	8.287599D-03
17	66.250	4.200476D-03	8.807364D-03	6.726829D-03	8.095032D-03
18	68.750	3.679145D-03	7.865126D-03	6.128369D-03	7.868678D-03
19	71.250	3.199101D-03	6.975551D-03	5.539233D-03	7.638829D-03
20	73.750	2.758982D-03	6.141288D-03	4.967120D-03	7.434761D-03
21	76.250	2.356361D-03	5.362947D-03	4.417498D-03	7.285830D-03
22	78.750	1.988404D-03	4.639861D-03	3.894120D-03	7.222718D-03
23	81.000	1.704440D-03	4.093331D-03	3.491854D-03	7.268466D-03
24	83.000	1.494316D-03	3.691740D-03	3.190902D-03	7.186613D-03
25	85.000	1.304604D-03	3.312685D-03	2.899718D-03	7.001339D-03
26	87.000	1.133743D-03	2.955832D-03	2.618805D-03	6.725256D-03
27	89.000	9.800857D-04	2.620425D-03	2.348345D-03	6.369873D-03
28	91.000	8.419524D-04	2.305360D-03	2.088243D-03	5.945665D-03
29	93.000	7.176675D-04	2.009261D-03	1.838157D-03	5.462149D-03
30	95.000	6.055826D-04	1.730533D-03	1.597533D-03	4.927966D-03
31	97.000	5.040921D-04	1.467416D-03	1.365639D-03	4.350966D-03
32	99.000	4.116414D-04	1.218022D-03	1.141588D-03	3.738294D-03
33	101.000	3.267291D-04	9.803637D-04	9.243720D-04	3.096482D-03
34	103.000	2.479048D-04	7.523863D-04	7.128775D-04	2.431544D-03
35	105.000	1.737639D-04	5.319826D-04	5.059127D-04	1.749062D-03
36	107.000	1.029398D-04	3.170109D-04	3.022252D-04	1.054287D-03
37	109.000	3.409409D-05	1.053068D-04	1.005205D-04	3.522327D-04

NORMALIZATION FACTOR = 1.249769D 02

FLUXES NORMALIZED TO ONE FISSION IN REACTOR

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

REAL NORMALIZED REGIONAL FLUX INTEGRALS BY GROUP

REGION	GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5	GROUP 6	GROUP 7
REG1	2.445959D-03	5.026741D-03	9.246510D-03	1.549984D-02	2.428211D-02	3.447526D-02	4.221975D-02
REG2	3.883535D-02	7.956554D-02	1.461047D-01	2.471493D-01	3.881252D-01	5.508564D-01	7.135341D-01
REG3	3.683905D-03	7.547562D-03	1.386161D-02	2.337871D-02	3.662754D-02	5.193805D-02	6.564969D-02
REGX	1.983706D-04	4.067606D-04	7.429463D-04	1.245128D-03	1.948683D-03	2.759882D-03	3.484688D-03
REG4	3.710409D-04	7.813313D-04	1.420522D-03	2.262223D-03	3.574976D-03	5.176151D-03	6.082418D-03

REGION	GROUP 8	GROUP 9	GROUP 10	GROUP 11	GROUP 12	GROUP 13	GROUP 14
REG1	5.332632D-02	7.497788D-02	1.073830D-01	1.169349D-01	1.420861D-01	1.637398D-01	2.143003D-01
REG2	9.154536D-01	1.212842D 00	1.629902D 00	1.795785D 00	2.134319D 00	2.388655D 00	2.922578D 00
REG3	8.355892D-02	1.134406D-01	1.565869D-01	1.717891D-01	2.061607D-01	2.338931D-01	2.951033D-01
REGX	4.433385D-03	6.036879D-03	8.436047D-03	9.347493D-03	1.128548D-02	1.295686D-02	1.667672D-02
REG4	7.676482D-03	1.155367D-02	1.813308D-02	2.034503D-02	2.533002D-02	3.092932D-02	4.530862D-02

REGION	GROUP 15	GROUP 16	GROUP 17	GROUP 18	GROUP 19	GROUP 20	GROUP 21
REG1	2.233994D-01	2.945091D-01	2.106932D 00	5.005914D 00	8.611959D 00	1.892799D 01	1.517744D 01
REG2	3.028585D 00	3.685546D 00	2.091782D 01	3.798687D 01	5.729829D 01	1.100703D 02	7.575353D 01
REG3	3.068160D-01	3.888293D-01	2.511465D 00	5.380296D 00	8.841911D 00	1.873713D 01	1.444929D 01
REGX	1.767726D-02	2.272073D-02	1.588704D-01	3.744770D-01	6.680198D-01	1.558797D 00	1.308260D 00
REG4	5.133922D-02	6.988394D-02	5.610367D-01	1.422511D 00	2.844502D 00	7.550087D 00	6.753205D 00

REGION	GROUP 22
REG1	1.856089D 01
REG2	6.811488D 01
REG3	1.848602D 01
REGX	2.426528D 00
REG4	1.818477D 01

NORMALIZATION FACTOR = 1.249769D 02

FLUXES NORMALIZED TO ONE FISSION IN REACTOR

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

AVERAGE POWER BY INTERVAL

INTERVAL	AVERAGE POWER	INTERVAL	AVERAGE POWER	INTERVAL	AVERAGE POWER	INTERVAL	AVERAGE POWER
1	4.835543D-06	2	5.268171D-06	3	6.194194D-06	4	7.743896D-06
5	9.813418D-05	6	1.057732D-04	7	1.111200D-04	8	1.140226D-04
9	1.143329D-04	10	1.119759D-04	11	1.069745D-04	12	9.945042D-05
13	8.960971D-05	14	7.770533D-05	15	2.527311D-05	16	2.246228D-05
17	2.002645D-05	18	1.788489D-05	19	1.598133D-05	20	1.427554D-05
21	1.273729D-05	22	1.134189D-05	23	0.0	24	0.0
25	0.0	26	0.0	27	0.0	28	0.0
29	0.0	30	0.0	31	0.0	32	0.0
33	0.0	34	0.0	35	0.0	36	0.0
37	0.0						

MAXIMUM AVERAGE POWER 1.143329D-04 OCCURS AT I = 9

MINIMUM AVERAGE POWER 4.835543D-06 OCCURS AT I = 1

POWER AND AVERAGE POWER BY REGION

REGION	POWER	AVERAGE POWER
REG1	1.081881D-02	6.010451D-06
REG2	9.261889D-01	1.029099D-04
REG3	5.788840D-02	1.837727D-05
REGX	5.103850D-03	1.134189D-05
REG4	0.0	0.0

MAXIMUM POWER 9.261889D-01 OCCURS IN REGION REG2

MINIMUM POWER 5.103850D-03 OCCURS IN REGION REGX

TOTAL POWER = 1.000000D 00

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

LEAKAGE BY REGION AND GROUP (INCLUDING D * B**2 TERMS)

REGION	GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5	GROUP 6	GROUP 7
REG1	-2.014556D-04	-4.024710D-04	-7.192881D-04	-1.148407D-03	-1.742014D-03	-2.464164D-03	-3.052797D-03
REG2	3.648660D-04	7.299803D-04	1.305446D-03	2.079722D-03	3.156566D-03	4.468687D-03	5.488742D-03
REG3	-1.045341D-04	-2.096082D-04	-3.742514D-04	-5.946385D-04	-9.052746D-04	-1.283674D-03	-1.594707D-03
REGX	4.239698D-06	7.976621D-06	1.437539D-05	2.468875D-05	3.636235D-05	5.019499D-05	7.136048D-05
REG4	-3.451781D-05	-6.830191D-05	-1.230601D-04	-1.988305D-04	-2.987727D-04	-4.202600D-04	-5.024245D-04

REGION	GROUP 8	GROUP 9	GROUP 10	GROUP 11	GROUP 12	GROUP 13	GROUP 14
REG1	-3.802585D-03	-4.934249D-03	-6.242299D-03	-6.598690D-03	-7.315434D-03	-7.906446D-03	-9.251783D-03
REG2	6.827536D-03	8.950971D-03	1.147222D-02	1.209017D-02	1.346603D-02	1.464463D-02	1.744738D-02
REG3	-1.994150D-03	-2.595579D-03	-3.313490D-03	-3.510097D-03	-3.902945D-03	-4.239879D-03	-5.038430D-03
REGX	8.872147D-05	9.160126D-05	7.578217D-05	8.294528D-05	7.362492D-05	5.117162D-05	-1.792287D-05
REG4	-6.146706D-04	-8.123819D-04	-1.033252D-03	-1.066076D-03	-1.183840D-03	-1.270986D-03	-1.476049D-03

REGION	GROUP 15	GROUP 16	GROUP 17	GROUP 18	GROUP 19	GROUP 20	GROUP 21
REG1	-9.175254D-03	-1.041568D-02	-4.606460D-02	-4.308756D-02	-3.874028D-02	-3.873413D-02	-1.442039D-03
REG2	1.733782D-02	2.013036D-02	9.638442D-02	1.079537D-01	1.149277D-01	1.532646D-01	5.281716D-02
REG3	-5.057261D-03	-5.757818D-03	-2.525538D-02	-2.173978D-02	-1.971296D-02	-1.888070D-02	1.701410D-03
REGX	-4.984246D-05	-1.198612D-04	-1.182310D-03	-1.941322D-03	-2.553651D-03	-3.559303D-03	-1.360601D-03
REG4	-1.374679D-03	-1.684427D-03	-1.001971D-02	-1.741700D-02	-1.949022D-02	-2.226059D-02	-8.079374D-03

REGION	GROUP 22
REG1	3.486222D-02
REG2	-1.888971D-02
REG3	4.758743D-03
REGX	-4.854882D-03
REG4	5.571234D-02

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

CURRENTS AT REGION INTERFACES BY GROUP (DIRECTION OF INCREASING DIMENSION)

INTERFACE LEFT	INTERFACE RIGHT	GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5	GROUP 6
0	REG1	0.0	0.0	0.0	0.0	0.0	0.0
REG1	REG2	-1.510424D-06	-3.017773D-06	-5.393468D-06	-8.609968D-06	-1.306012D-05	-1.847446D-05
REG2	REG3	1.026178D-06	2.055744D-06	3.677990D-06	5.849153D-06	8.884359D-06	1.258722D-05
REG3	REGX	2.312939D-07	4.617748D-07	8.317618D-07	1.327562D-06	2.001597D-06	2.827777D-06
REGX	REG4	2.619277D-07	5.192896D-07	9.354511D-07	1.506134D-06	2.264404D-06	3.190294D-06
REG4	0	3.246767D-09	7.244348D-09	1.293099D-08	1.659578D-08	2.585906D-08	4.052207D-08

INTERFACE LEFT	INTERFACE RIGHT	GROUP 7	GROUP 8	GROUP 9	GROUP 10	GROUP 11	GROUP 12
0	REG1	0.0	0.0	0.0	0.0	0.0	0.0
REG1	REG2	-2.287051D-05	-2.848267D-05	-3.698865D-05	-4.684013D-05	-4.950413D-05	-5.490280D-05
REG2	REG3	1.538452D-05	1.912937D-05	2.523775D-05	3.261980D-05	3.431749D-05	3.833687D-05
REG3	REGX	3.279231D-06	3.998503D-06	5.513159D-06	7.395772D-06	7.609103D-06	8.618640D-06
REGX	REG4	3.797599D-06	4.643246D-06	6.172993D-06	7.929152D-06	8.194322D-06	9.129574D-06
REG4	0	3.547261D-08	4.101435D-08	8.266377D-08	1.686748D-07	1.864700D-07	2.323010D-07

INTERFACE LEFT	INTERFACE RIGHT	GROUP 13	GROUP 14	GROUP 15	GROUP 16	GROUP 17	GROUP 18
0	REG1	0.0	0.0	0.0	0.0	0.0	0.0
REG1	REG2	-5.936992D-05	-6.956787D-05	-6.900127D-05	-7.848931D-05	-3.501261D-04	-3.353624D-04
REG2	REG3	4.185526D-05	5.044026D-05	5.021318D-05	5.906702D-05	2.956245D-04	3.586996D-04
REG3	REGX	9.542136D-06	1.195409D-05	1.158563D-05	1.492575D-05	9.898982D-05	1.812016D-04
REGX	REG4	9.880688D-06	1.176460D-05	1.115744D-05	1.395815D-05	8.964148D-05	1.656611D-04
REG4	0	3.154991D-07	6.128688D-07	7.457403D-07	1.168602D-06	1.293558D-05	3.145934D-05

INTERFACE LEFT	INTERFACE RIGHT	GROUP 19	GROUP 20	GROUP 21	GROUP 22
0	REG1	0.0	0.0	0.0	0.0
REG1	REG2	-3.093887D-04	-3.274821D-04	-3.515584D-05	2.366363D-04
REG2	REG3	4.058385D-04	5.782313D-04	2.338694D-04	8.325157D-06
REG3	REGX	2.374963D-04	3.985536D-04	2.231584D-04	2.071690D-05
REGX	REG4	2.168579D-04	3.688080D-04	2.109175D-04	-1.843195D-05
REG4	0	6.375205D-05	1.832240D-04	1.368449D-04	3.673668D-04

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

BALANCE SUMS BY REGION

REGION	LEAKAGE	REMOVAL	NON-FISSION SOURCE	FISSION SOURCE	EXTERNAL SOURCE	BALANCE
REG1	-2.085794D-01	1.045311D 00	8.092053D-01	2.890119D-02	0.0	6.902097D-11
REG2	6.464191D-01	7.563657D 00	5.651600D 00	2.686299D 00	0.0	2.810545D-10
REG3	-1.196050D-01	1.124199D 00	8.470711D-01	1.653933D-01	0.0	-3.759433D-10
REGX	-1.496665D-02	9.057695D-02	6.174635D-02	1.455660D-02	0.0	-3.636769D-11
REG4	-3.371708D-02	3.164121D-01	2.826950D-01	0.0	0.0	-4.776975D-15
TOTAL	2.695509D-01	1.014016D 01	7.652318D 00	2.895150D 00	0.0	-6.224033D-11

BALANCE SUMS BY GROUP

GROUP	LEAKAGE	REMOVAL	NON-FISSION SOURCE	FISSION SOURCE	EXTERNAL SOURCE	BALANCE
1	2.859812D-05	4.966814D-03	0.0	5.244987D-03	0.0	-1.269850D-13
2	5.757588D-05	1.012067D-02	3.885868D-04	1.027876D-02	0.0	-2.441789D-13
3	1.032214D-04	1.803801D-02	8.362062D-04	1.822210D-02	0.0	-4.258453D-13
4	1.625349D-04	2.969392D-02	1.699626D-03	2.956357D-02	0.0	-6.811878D-13
5	2.468674D-04	4.586835D-02	3.877581D-03	4.434787D-02	0.0	-1.009430D-12
6	3.507840D-04	6.520344D-02	6.435230D-03	6.207263D-02	0.0	-1.397948D-12
7	4.101739D-04	8.790352D-02	1.048069D-02	8.172162D-02	0.0	-1.823588D-12
8	5.049025D-04	1.149477D-01	1.837856D-02	1.019239D-01	0.0	-2.256277D-12
9	7.003625D-04	1.401226D-01	2.540219D-02	1.211873D-01	0.0	-2.664021D-12
10	9.589604D-04	1.806397D-01	5.003789D-02	1.381337D-01	0.0	-3.018100D-12
11	9.982501D-04	2.050677D-01	6.159775D-02	1.516860D-01	0.0	-3.296599D-12
12	1.137438D-03	2.417700D-01	8.940613D-02	1.611704D-01	0.0	-3.486426D-12
13	1.278485D-03	2.678702D-01	1.107255D-01	1.663381D-01	0.0	-3.583455D-12
14	1.663192D-03	3.061559D-01	1.484624D-01	1.673183D-01	0.0	-3.591751D-12
15	1.680784D-03	3.305790D-01	1.755586D-01	1.645301D-01	0.0	-3.520829D-12
16	2.152581D-03	3.610379D-01	2.121553D-01	1.585810D-01	0.0	-3.384124D-12
17	1.386242D-02	1.019228D 00	5.227422D-01	5.358462D-01	0.0	-1.137608D-11
18	2.376804D-02	1.201770D 00	8.887164D-01	3.536492D-01	0.0	-7.463934D-12
19	3.443064D-02	1.506436D 00	1.345769D 00	2.048449D-01	0.0	-4.306702D-12
20	6.982986D-02	1.785476D 00	1.697826D 00	1.653480D-01	0.0	-3.469636D-12
21	4.363656D-02	1.359792D 00	1.364384D 00	4.099525D-02	0.0	-8.574529D-13
22	7.158871D-02	8.574174D-01	9.174376D-01	1.214640D-02	0.0	-2.557796D-13
TOTAL	2.695509D-01	1.014016D 01	7.652318D 00	2.895150D 00	0.0	-6.224033D-11

ONE-DIMENSIONAL DIFFUSION THEORY SAMPLE OUTPUT

```
SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =      9035
SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =         0

                                  .
                                  .
                                  .
                                  .

***  LEAVING MODULE DIF1D  (NUC002)        TIME =   0.496 MINUTES  ***
```

PART TWO.
ONE-DIMENSIONAL, MULTIGROUP DIFFUSION THEORY CRITICALITY SEARCH MODULE, D1DSCH (NUC004)

Analysis Done By - G. K. Leaf and A. S. Kennedy

Programmed By - A. S. Kennedy and D. E. Neal

Documented By - D. E. Neal and G. K. Leaf

1. D1DSCH (NUC004) Introduction

a. Description of Capabilities

This module performs a diffusion theory search, to either a prescribed k-effective or a prescribed inverse reactor period α, sharing the input, output, and k-effective calculation portions of module DIF1D (NUC002). The parameters which may be varied in the search include material volume fractions, mesh interval dimensions, geometric buckling, and α.

b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path

Modules Utilized by Each Path.	Paths Utilizing This Module			
	STP001 1-D Diffusion Theory	STP003 2 -D Diffusion Theory	STP004 Burnup Diffusion Theory	STP006 1-D Diffusion Theory Perturbation
DELDATA (CSI007) Delayed Neutron Data Preparation				X
INHOMG (NUI001) Multigroup Cross Section Homog. Specifications	X	X	X	X
HOMOG (NUC001) Multigroup Cross Section Homog.	X	X	X	X
HHOMOG (NUC007) Material-to-Composition Cross Section Homogenization			X	
GNIP (NUI002) Neutronics Input Processor	X	X	X	X
DARC1D { DIF1D (NUC002) 1-D Diffusion Theory k_{eff} and Source Calculations	X	X	X	X

b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path (Cont'd.)

	Modules Utilized by Each Path	Paths Utilizing This Module			
		STP001 1-D Diffusion Theory	STP003 2-D Diffusion Theory	STP004 Burnup Diffusion Theory	STP006 1-D Diffusion Theory Perturbation
DARC1D	D1DSCH (NUC004) 1-D Diffusion Theory Criticality Search	X	X	X	X
DARC2D	DIF2D (NUC005) 2-D Diffusion Theory k_{eff} and Source Calculations		X	X	
DARC2D	D2DSCH (NUC004) 2-D Diffusion Theory Criticality Search		X	X	
	MODXSISO (NUI006) Run-time Microscopic Cross Section Modification	X	X		X
	PARC1D (AJC003) 1-D Diffusion Theory Perturbation				X

b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path (Cont'd.)

Modules Utilized by Each Path	Paths Utilizing This Module			
	STP001 1-D Diffusion Theory	STP003 2-D Diffusion Theory	STP004 Burnup Diffusion Theory	STP006 1-D Diffusion Theory Perturbation
OUTMAN1D (NUE001) 1-D Neutronics Output Manipulation	X	X		X
INVENT1D (AJC001) 1-D Neutron Inventory	X	X		
INVENT2D (AJC006) 2-D Neutron Inventory		X		
CPGEOM (AJC006) Geometry Data Set (GEOM) Copier				X
REBUS: INFUEL (FCI001) Fuel Cycle Specifications			X	
REBUS: FUELCY (FCC001) Fuel Cycle Analysis			X	

c. Data Sets Utilized by This Module

Upper-interface Data Sets	Records Read
A.DIF1D, One-dimensional Diffusion Theory Module-dependent BCD Input	All
BC, Boundary Condition Specifications	All
FA.D1, One-dimensional Adjoint Group Fluxes	All
or	
FR.D1, One-dimensional Real Group Fluxes	All
GEOM, Geometry Data	All except Region Volumes
SP.CICN, Module-independent Data	All
SP.CRIT, Criticality Search Specifications	All
XS.C.MIN, Macroscopic Composition Cross Sections	All
XS.ISO2, File 1, Problem-dependent Microscopic Group Cross sections File 1	File Size, Group Structure
or	
XS.ISO, File 1, Microscopic Group Cross Sections, File 1	File Size, Group Structure

Lower-interface Data Sets	
B.HOMOG, Binary Version of Homogenization Input	All
FA.D1, One-dimensional Adjoint Group Fluxes	All
and/or	
FR.D1, One-dimensional Real Group Fluxes	All
FR.PN, Power-normalized Real Regional Group Fluxes	All
FSA.D1, One-dimensional Adjoint Fission Source	All
and/or	
FSR.D1, One-dimensional Real Fission Source	All
GEOM, Geometry Data	All
XS.C.MIN, Macroscopic Composition Cross Sections	All

2. D1DSCH (NUC004) Description of Problem and Method of Solution

a. Introduction

This module adjusts certain parametric vectors, which are constrained to lie on a given straight line, in order to achieve a desired value of k_{eff} or of the inverse reactor period α. The parametric vectors include material volume fractions, mesh interval dimensions, geometric buckling, and α.

b. Problem Formulation

If $\overline{P} = \{P_\gamma\}$ denotes a parametric vector, then the desired vector is constrained to lie on a line given by

$$P_\gamma(\lambda) = P_\gamma(\lambda_0) + \lambda\delta P_\gamma \quad , \tag{b.1}$$

where δP_γ denotes the parameter modifiers (the direction numbers for the line), and $P_\gamma(\lambda_0)$ denotes the initial parameter value defined.

For any given value of λ, the parametric vector $\overline{P}(\lambda)$ generates the matrices $M(\lambda)$ and $F(\lambda)$. Hence, the corresponding value of $k_{eff} = k(\lambda)$ is obtained as a solution of the eigenvalue equation

$$M(\lambda)\phi = \frac{1}{k} F(\lambda)\phi \quad . \tag{b.2}$$

The object of any search for a desired k_{eff} is then to solve the equation

$$k(\lambda) = k_d \quad , \tag{b.3}$$

where k_d denotes the value of the desired k_{eff}.

A particular example of a search for a prescribed k_{eff} is the α-search. In this case, the parametric vector is a scalar; thus $\overline{P}(\lambda) = \alpha(\lambda) = \alpha(\lambda_0) = \lambda\delta\alpha$. Moreover, $M(\lambda) = M + \alpha(\lambda)$ and $F(\lambda) \equiv F$. From this we see that an α-search involves searching for the value of α satisfying the equation

$$(M + \alpha)\phi = \frac{1}{k_d} F\phi \quad . \tag{b.4}$$

Normally, $k_d = 1$.

It is also possible to search for a desired inverse period, α_d. Given any parametric vector $\overline{P}(\lambda)$, a search to achieve the desired α involves finding that value of λ for which the following equation has a positive solution vector ϕ:

$$[M(\lambda) + \alpha_d]\phi = F(\lambda)\phi \quad . \tag{b.5}$$

Since the equation

$$[M(\lambda) + \alpha_d]\phi = \frac{1}{k(\lambda)} F(\lambda)\phi \tag{b.6}$$

has a positive solution vector and eigenvalue $k(\lambda)$ for each λ, the solution to Eq. (b.5) is achieved by repeatedly solving Eq. (b.6), then searching for that value of λ for which $k(\lambda) = 1$.

$$\lambda_{n+1} = \lambda_n - \frac{[k_n - k_d]}{\left[\dfrac{k_n - k_{n-1}}{\lambda_n - \lambda_{n-1}}\right]}$$

$$\lambda_{n+1} = \lambda_n - (\lambda_n - \lambda_{n-1})\left[\frac{k_n - k_d}{k_n - k_{n-1}}\right] \quad .$$

c. Method of Solution

As shown in the previous section, all searches involve solving the equation (b.3), where $k(\lambda)$ denotes the value of k_{eff} corresponding to a given parametric vector $\overline{P}(\lambda)$. The value of $k(\lambda)$ is obtained from a k-effective calculation by the one-dimensional k-effective and source calculation module, DIF1D (NUC002). There are two possible search procedures. The first is a linear procedure which coincides with the method of regula falsi as soon as the desired value k_d has been bracketed. The second is based on parabolic interpolation, with linear interpolation being used to start the procedure. The parabolic method is used wherever feasible.

In the parabolic procedure, a parabola is passed through three points of the function $k(\lambda)$, and this curve is used to estimate the solution of equation (b.3). Again, the desired value of k_{eff} is bracketed. If the calculated parabola does not intersect k_d, or if the three points approximate a straight line, linear interpolation is used.

In the solution used in this module, the parameter values $P_\gamma(\lambda)$ are updated immediately on each pass n, so that instead of Eq. (b.1) the following equation is used:

$$P_\gamma(\lambda_n) = P_\gamma(\lambda_{n-1}) + \delta_n \delta P_\gamma \quad , \tag{c.1}$$

where

$$\delta_n = \lambda_n - \lambda_{n-1}$$

and

$$\lambda_n = \lambda_0 + \sum_{i=1}^{n} \delta_i \quad .$$

On the first pass,

$$\delta_1 = \lambda_1 \quad ,$$

from the input data set SP.CRIT. On the second pass,

$$\lambda_2 = \lambda_1 + \frac{k_d - k(\lambda_1)}{\left(\frac{dk}{d\lambda}\right)_1}$$

if $(dk/d\lambda)_1$ has a nonzero value in data set SP.CRIT; otherwise, λ_2 must be present in data set SP.CRIT. In either case,

$$\delta_2 = \lambda_2 - \lambda_1 \quad .$$

Translating the parametric vectors into the search types possible, we have the following.

(i) Dimension search (X)

P_γ represents the mesh increments, ΔX_i.

δP_γ represents the mesh modifiers, $\delta(\Delta X_i)$.

The equation thus becomes

$$(\Delta X_i)_n = (\Delta X_i)_{n-1} + \delta_n \cdot \delta(\Delta X_i)$$

for each modified interval i.

(ii) Buckling search (B^2)

P_γ represents the buckling by region and transverse direction, $B^2_{r,t}$; δP_γ represents the buckling modifiers $\delta B^2_{r,t}$. The search equation is

$$(B^2_{r,t})_n = (B^2_{r,t})_{n-1} + \delta_n(\delta B^2_{r,t}) \quad .$$

The change in buckling $B^2_{r,t}$ causes a change in the actual region half-height or radius $(H_L)_{r,t}$, while the extrapolation distance $E_{r,t}$ remains constant. This produces a change in the extrapolated half-height or radius $H_{r,t} = (H_L)_{r,t} + E_{r,t}$, as follows: for rectangular slab or cylindrical geometry, $H_{r,t} = (\pi/2B_{r,t})$; for cylindrical slab geometry, $H_{r,t} = (\alpha/B_{r,t})$, where α is the first zero of the Bessel function J_0. From this, the actual half-height or radius can be recomputed as

$$(H_L)_{r,t} = H_{r,t} - E_{r,t} \quad .$$

(iii) Alpha search (α)

Here, $P_\gamma = \alpha$ and $\delta P_\gamma = \delta\alpha$, the modifier. The search equation is

$$(\alpha)_n = (\alpha)_{n-1} + \delta_n\delta\alpha \quad .$$

(iv) Concentration search (σ_x)

This search operates on the volume fractions of the materials, modifying each cross section type. P_γ is σ_x, the cross section being modified; δP is σ_{xm}, the corresponding cross section of the modifying composition. If the cross section type being modified is σ_R, σ_f, $\nu\cdot\sigma_f$, or $\sigma^{g'g}$, the equation is simply

$$(\sigma_x)_n = (\sigma_x)_{n-1} + \delta_n(\sigma_{xm}) \quad .$$

For the diffusion coefficient, the transport cross section is in effect modified:

$$(D_c^g)_n = \frac{(D_c^g)_{n-1} (D_m^g)}{(D_m^g) + {}_n(D_c^g)_{n-1}} .$$

For the χ^g (vector) or $\chi^{g'g}$ (matrix) cross section, the modification is

$$(\chi_c^g)_n = \frac{(\chi_c^g)_{n-1} \cdot [(\nu\cdot\sigma_f)^g]_{c\,n-1} + \delta_n \cdot \chi_m^g \cdot (\nu\cdot\sigma_f)_m^g}{[(\nu\cdot\sigma_f)^g]_{c\,n}}$$

$$(\chi_c^{g'g})_n = \frac{\{(\chi_c^{g'g})_{n-1} \cdot [(\nu\cdot\sigma_f)^{g'}]_{c\,n-1} + \delta_n \cdot \chi_m^{g'g} \cdot (\nu\cdot\sigma_f)_m^{g'}\}}{[(\nu\cdot\sigma_f)^{g'}]_{c\,n}}$$

A nonfissionable composition can modify a fissionable one, but a fissionable composition cannot modify a nonfissionable one. If the latter is desired, the composition to be modified can be made to appear fissionable by adding to it a zero volume fraction of a fissionable material.

The scattering band width of the modifying composition must be less than or equal to that of the modified composition. This can be handled similarly by adding a zero volume fraction of a material having the greater band width to the modified composition.

d. Output Edits

The edits provided by this module can be divided into three categories: input, search pass history, k_{eff} calculation iteration history, and output.

(i) Input

The input data edits provided by this module are the same as those provided by the one-dimensional k-effective and source calculation module, DIF1D (NUC002), except that data read from

the criticality search specifications data set, SP.CRIT, are also displayed. The data printed include the prescribed k_{eff} or α to be searched to; the type of search desired (CONC, X-DIM, BSQ, or ALPHA); the maximum number of control changes; the first and second guesses at the control parameter X (λ); the initial (dk_{eff}/dx); the lower and upper bounds of X; and the convergence criterion. For a concentration search, the name of each composition being modified is printed, along with the name of its modifier; for an X-dimension search, the numbers of the intervals being modified are printed, along with the modifier $\delta\Delta X_i$ for each; and for a buckling search the buckling in each region and direction being modified is given, along with its modifier $\delta B^2_{r,t}$.

(ii) Search pass history

For each control pass after the first the value of the new control parameter $X(\lambda_n)$, of the k_{eff} found on the previous pass $[k(\lambda_{n-1})]$, and of the change in the control parameter $\overline{X}$ (δ_n) are displayed. In addition, the names of compositions modified and their modifiers are reprinted for a concentration search; the new values for interval lengths are given for an X-dimension search; and the modified values for the buckling and half-height and the extrapolation distance are shown for a buckling search.

When the search has converged, the geometry data (for an X-dimension or buckling search) or the composition data (for a concentration search) are displayed again, in the same format as in the input section. (The composition data are also displayed if any cross section values become negative during a concentration search pass; this condition also terminates the search.) In addition, the appropriate data set or sets (XS.C.MIN and B.HOMOG, or GEOM) are updated upon convergence.

(iii) k_{eff} calculation iteration history

This section, displayed for each control pass, is identical to that described for module DIF1D (NUC002).

(iv) Output

This section is also identical to that for the one-dimensional diffusion k-effective and source calculation module.

e. References

1. A. Hassitt, *CRAM - A One- and Two-Dimensional Multigroup Diffusion Theory Reactor Code*, TRG Report 229(r) (1962).

2. R. S. Varga, *Matrix Iterative Analysis*, (Prentice-Hall, Inc., New Jersey, 1962).

3. A. Ralston and H. S. Wilf, Eds., *Mathematical Methods for Digital Computers*, (John Wiley & Sones, Inc., 1960).

4. G. G. Bilodeau et al., *PDQ - An IBM-704 Code to Solve the Two-Dimensional Few-Group Neutron Diffusion Equations*, WAPD-TM (1957).

5. B. J. Toppel, Ed., *The Argonne Reactor Computation (ARC) System*, ANL-7332 (1967).

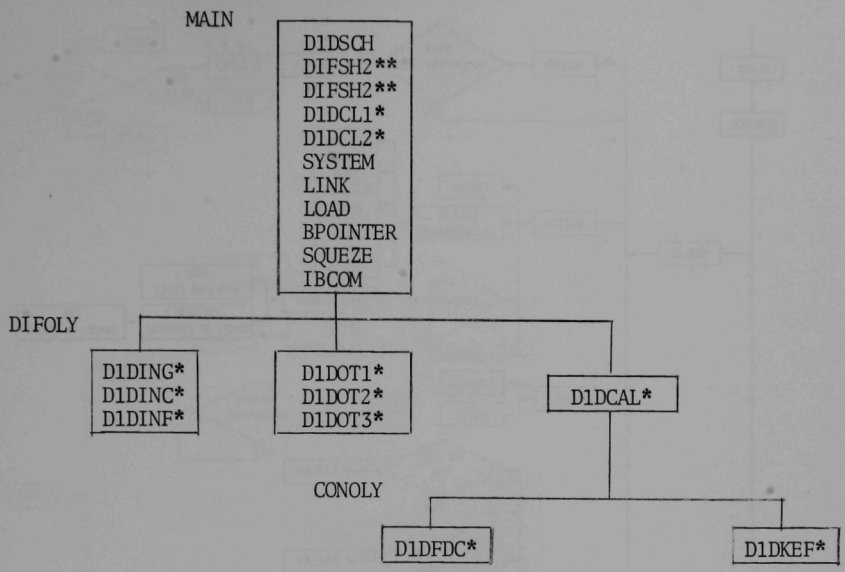
3. D1DSCH (NUC004) Description of Module Flow

This section defines the overlay structure of this module and the flow of logic through the subprograms which form the module.

Section a. consists of an overlay structure diagram, showing the ARC System segments and DIF1D and D1DSCH module segments required by this module and their positions in the overlay heirarchy. This is followed by a list of the subprograms located in each D1DSCH module segment, in alphabetical order.

Section b. illustrates the flow of logic through the subprograms of this module, and section c. describes this flow in terms of the module's functions. Calls to PAGHED, the subroutine which prints the page heading, and ERRMSG, the subroutine which prints error messages, are not indicated, nor are calls to subroutines or entry points which are part of the ARC System utility routines, such as POINTR and SNIFF.

a. D1DSCH (NUC004) Overlay Structure Diagram



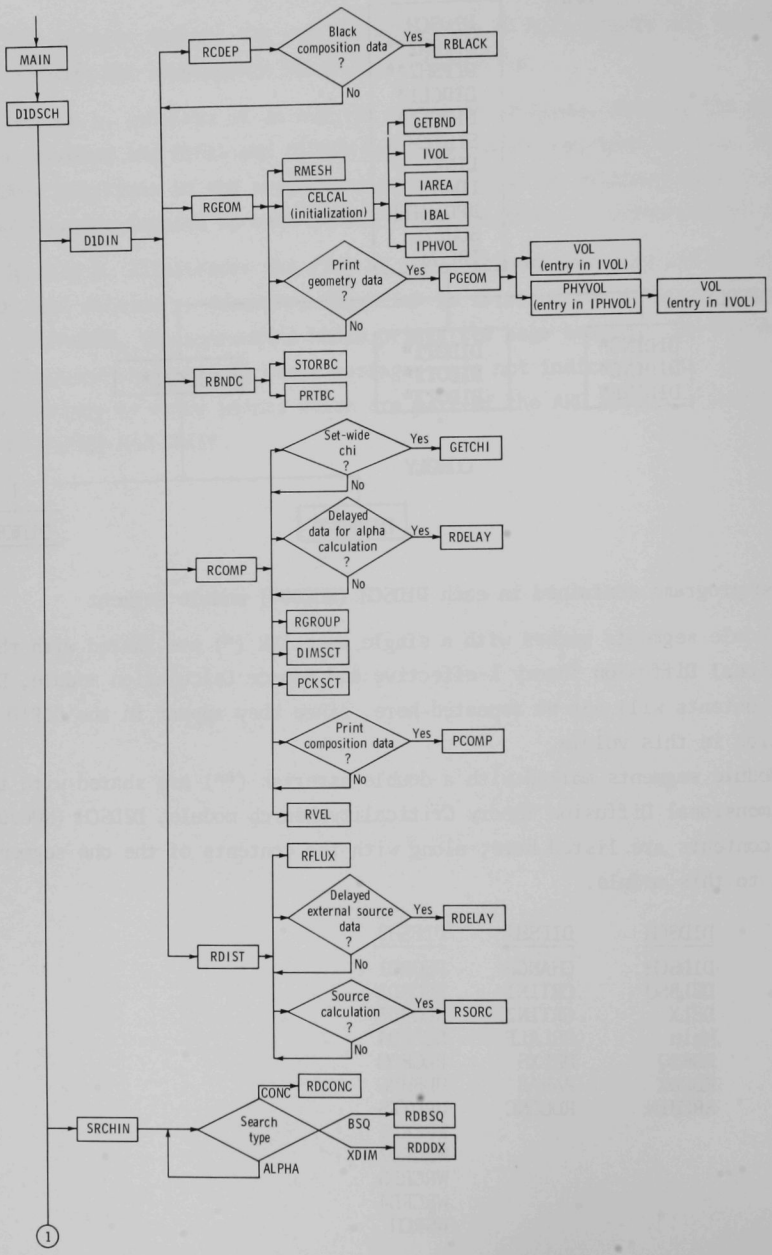
Subprograms contained in each D1DSCH (NUC004) module segment

Module segments marked with a single asterisk (*) are shared with the One-dimensional Diffusion Theory k-effective and Source Calculation module, DIF1D; their contents will not be repeated here, since they appear in the DIF1D documentation in this volume.

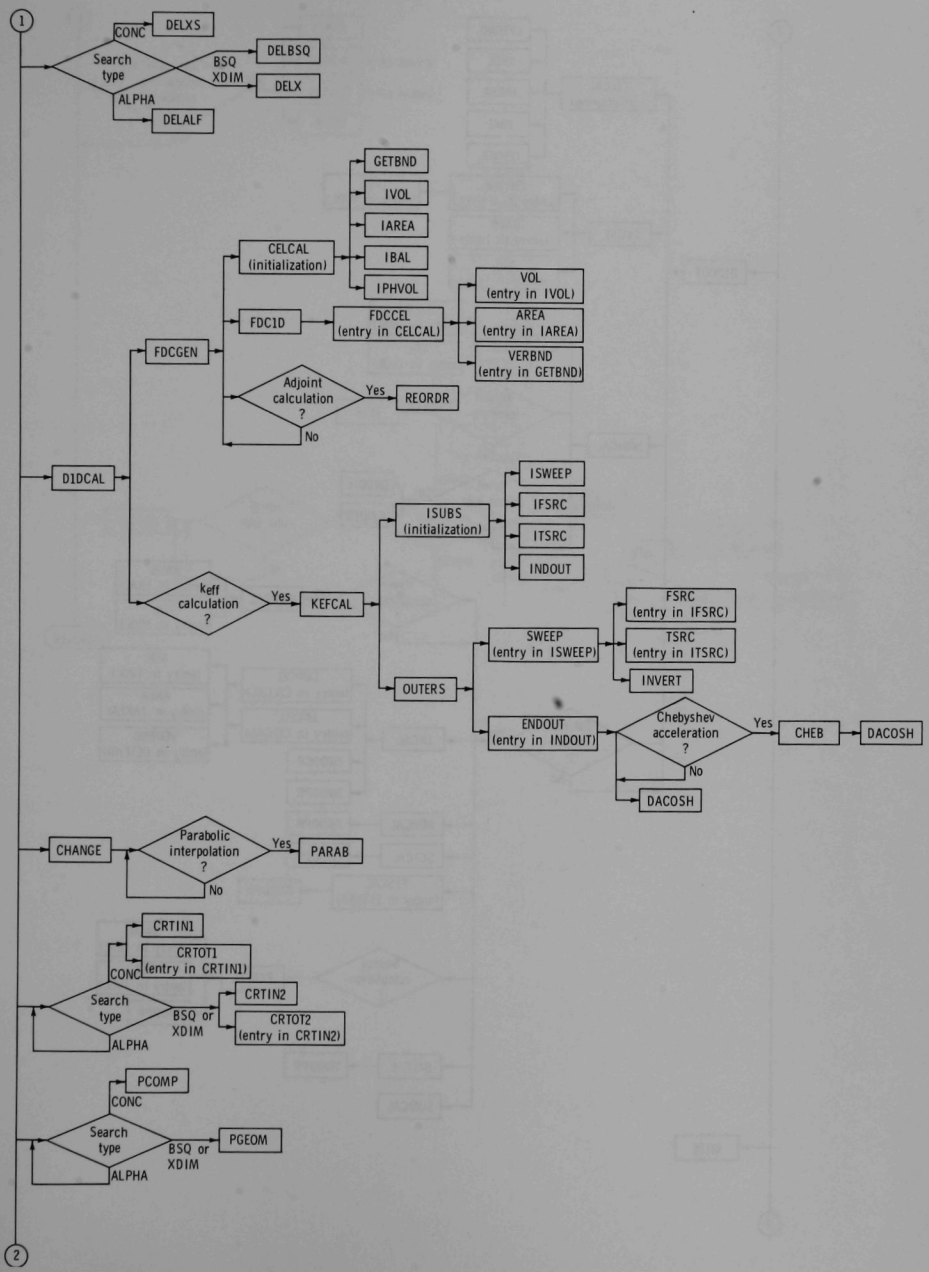
Module segments marked with a double asterisk (**) are shared with the Two-dimensional Diffusion Theory Criticality Search module, D2DSCH (NUC006). Their contents are listed here, along with the contents of the one segment unique to this module.

D1DSCH	D1FSH1	DIFSH2
D1DSCH	CHANGE	RCORD1
DELBSQ	CRTIN1	RCORD2
DELX	CRTIN2	RCORD3
Main	DELALF	RCORD4
RDBSQ	DELXS	RDGEOM
RDDDX	PARAB	UDBHMG
SRCHIN	RDCONC	UPDTXS
		WRCRD1
		WRCRD2
		WRCRD3
		WRCRD4
		WSRCH

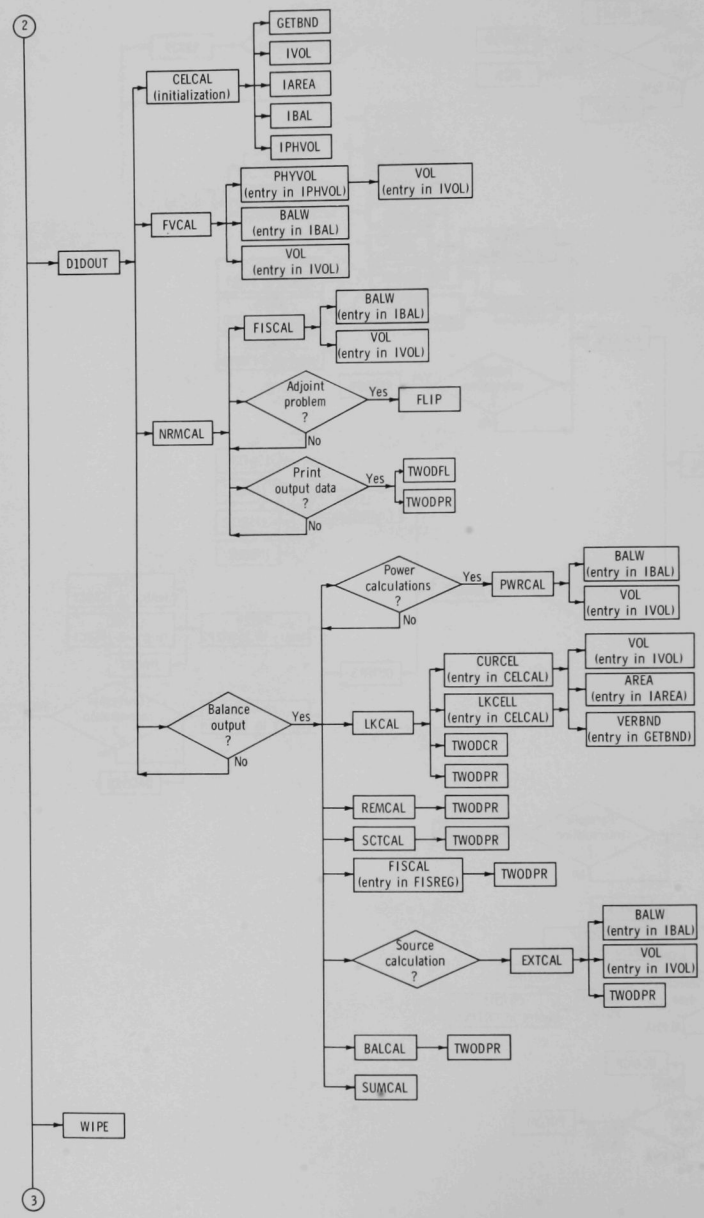
b. Diagrammatic Subprogram Logic Flow



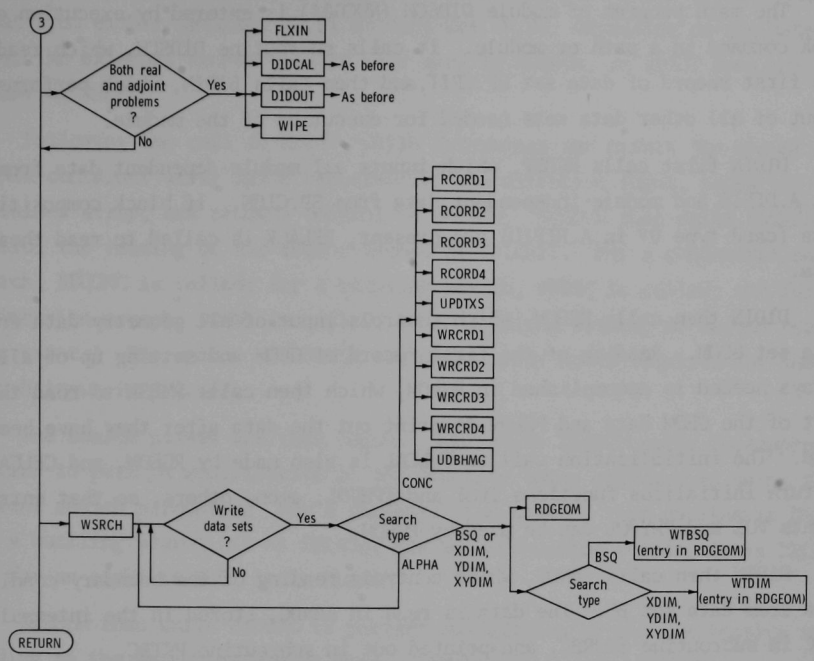
b. Diagrammatic Subprogram Logic Flow (Contd.)



b. Diagrammatic Subprogram Logic Flow (Contd.)



b. Diagrammatic Subprogram Logic Flow (Contd.)



c. Descriptive Subprogram Logic Flow

The main program of module D1DSCH (NUC004) is entered by execution of a LINK command in a path or module. It calls subroutine D1DSCH, which reads the first record of data set SP.CRIT, and then calls D1DIN, which performs input of all other data sets needed for execution of the module.

D1DIN first calls RCDEP, which inputs all module-dependent data from data set A.DIF1D and module-independent data from SP.CICN. If black composition data (card type 07 in A.DIF1D) are present, RBLACK is called to read these data.

D1DIN then calls RGEOM, which controls input of all geometry data from data set GEOM. Reading of the first record of GEOM and setting up of all arrays needed is accomplished in RGEOM, which then calls RMESH to read the rest of the GEOM data and PGEOM to print out the data after they have been read. The initialization call to CELCAL is also made by RGEOM, and CELCAL in turn initializes functions IVOL and IPHVOL, among others, so that entry points VOL and PHYVOL can be used by PGEOM.

D1DIN then calls RBNDC, which controls reading of the boundary condition data from data set BC. The data is read in RBNDC, stored in the internal array BNDC in subroutine STORBC, and printed out in subroutine PRTBC.

D1DIN then calls RCOMP to read composition data. RCOMP reads the first record of data set XS.C.MIN, sets up all arrays needed, and calls GETCHI to read the next record if there is a set-wide fission spectrum. It then calls RDELAY to read data set XS.DELAY* if delayed neutrons are to be taken into account for an alpha calculation, and then calls RGROUP once for each composition to read group-dependent data from XS.C.MIN. RCOMP then calls DIMSCT to set up the position of the scattering terms in the scattering array, PCKSCT to pack the scattering array by compositions, and PCOMP to print out the composition-related data. Its final call is to RVEL, which reads group-dependent neutron speeds from data set XS.ISO2 or XS.ISO for an alpha calculation.

---

*This data set has not been defined.

D1DIN then calls RDIST, which controls reading of distributed data. RDIST sets up all arrays needed for this data and calls RFLUX to read flux input from either data set FR.D1 or data set FA.D1, depending on the problem type. No external source data is ever used by NUC004, so RDIST does not call RDELAY or RSORC.

Following the call to RDIST, D1DIN determines and prints the amount of POINTR container array space required for the diffusion input, packs the container array, and returns control to D1DSCH. D1DSCH then calls SRCHIN to control the reading of the second record of SP.CRIT. For a concentration search, RDCONC is called; for a buckling search, RDBSQ is called; and for an X-dimension search, RDDDX is called. (For an alpha search, no second record is present.) The amount of POINTR container array space required for the search input is printed.

The search passes are then begun. Subroutine D1DSCH calls the appropriate routine to perform modification of the quantity being searched on using the current search parameter. For a concentration search, this routine is DELXS; for a buckling search, it is DELBSQ; for an X-dimension search, it is DELX; and for an alpha search, it is DELALF.

D1DSCH then calls D1DCAL to perform calculation of the k-effective corresponding to the search parameter used. D1DCAL first calls FDCGEN, which controls generation of the finite difference coefficients.

FDCGEN sets up the needed arrays and calls CELCAL to initialize routines to be used. It then calls FDC1D to control the calculation of the coefficients. FDC1D calls entry point FDCCEL of CELCAL for each interval and group to calculate the coefficients. It then completes the calculation of these and related terms, and re-orders these arrays by groups if an adjoint problem is to be done. Then, if this is an adjoint problem, FDCGEN calls REORDR to re-order other group-dependent arrays before returning control to D1DCAL.

D1DCAL then determines if the problem is a k-effective calculation or a source calculation; since for a search it is always a k-effective calculation, it calls KEFCAL.

KEFCAL controls k-effective calculations. It first sets up necessary arrays, determines and prints the amount of POINTR array space required for the calculation section, and calls ISUBS to initialize subroutines. It then transfers control to OUTERS. OUTERS controls the number of power iterations done and writes FR.D1 or FA.D1 after a specified number of passes if this restart option has been specified. For every iteration, it calls SWEEP (entry in ISWEEP) to perform the mesh sweep and ENDOUT (entry in INDOUT) to perform Chebyshev acceleration if desired, to compute monitoring information, and to test for convergence.

SWEEP calls FSRC (entry in IFSRC) to compute the fission source for every point and group. Then, for each group, it calls TSRC (entry in ITSRC) to compute the total source including scattering for each channel and INVERT to perform a Choleski inversion of the resulting tri-diagonal matrix. After this has been completed for all groups, SWEEP performs upscattering iterations, if any, in the same manner, calling TSRC and INVERT.

ENDOUT calculates and prints monitoring information and performs Chebyshev acceleration, if it is called for, by calling CHEB to obtain the Chebyshev coefficients and then by modifying the flux appropriately. It also tests for convergence and returns a flag to OUTERS, which returns control to D1DCAL and then to D1DSCH if the problem has converged or has exceeded the maximum number of outer iterations and continues to another iteration if not.

Subroutine D1DSCH then calls CHANGE to check for convergence of the search and to compute a new search parameter if it has not converged. CHANGE calls PARAB if parabolic interpolation is desired.

Following this, D1DSCH calls CRTIN1 for a concentration search or CRTIN2 for a buckling or X-dimension search to read the second record of SP.CRIT, and then updates the data set, using CRTOT1 (entry in CRTIN1) or CRTOT2 (entry in CRTIN2) to write the second record, if present.

If the search has converged or if the maximum number of search iterations has been exceeded D1DSCH calls PCOMP for a concentration search or PGEOM for a buckling or X-dimension search to print the search results. It then calls D1DOUT to control output of data sets and of printed output, if desired.

D1DOUT wipes out unneeded arrays, packs the POINTR container arrays, sets up necessary arrays, and calls CELCAL to initialize various routines used. It then calls FVCAL to compute regional flux integrals and then NRMCAL. NRMCAL computes regional flux averages and regional flux integrals normalized for BURNUP, if desired, normalizes fluxes and flux-related terms for a k-effective calculation, calls FISCAL to calculate fission density and fission source integrals by mesh interval and group, and then calls FLIP to re-order these quantities by group if the problem is adjoint. NRMCAL then outputs data sets: FR.D1 or FA.D1 for fluxes, FR.PN for a real problem for which power-normalization of fluxes for BURNUP is specified, and FSR.D1 or FSA.D1 if fission source integral output is requested. Then, if printing of these quantities has been specified, NRMCAL prints all flux-related and fission-related quantities thus far computed, utilizing printing routines TWODPR and TWODFL.

D1DOUT then returns control to subroutine D1DSCH if further output is not desired. If power-related terms are desired, D1DOUT calls PWRCAL to compute and print them. D1DOUT then calls LKCAL, which computes the leakage contribution to the balance equation (using entry LKCELL of CELCAL) and the current across region interfaces (using entry CURCEL of CELCAL), and prints them by calling TWODPR and TWODCR.

D1DOUT then calls REMCAL to compute the removal contribution to the balance equation and print it using TWODPR; SCTCAL to compute the scattering contribution to the balance equation and print it using TWODPR; FISREG (entry in FISCAL) to compute the fission source contribution to the balance calculation and print it using TWODPR; and BALCAL to calculate the balance term and print it using TWODPR. Finally, SUMCAL is called to sum all balance-related terms over regions, over groups, and over the reactor, and to print these quantities. Following this, control is returned to D1DSCH, where output arrays are wiped out of the POINTR container arrays and the container arrays are packed.

If the problem being solved was both real and adjoint, D1DCAL and D1DOUT are called again to perform an adjoint calculation based on the results of the real search and the wiping out and packing of the container array is repeated. Following this, for any problem type, WSRCH is called to output data sets whose contents are altered by the search, if this is desired.

For a concentration search, WSRCH sets up necessary arrays and calls RCORD1, RCORD2, RCORD3, and RCORD4 to read all data from data set XS.C.MIN. It then calls UPDTXS to update this data using the final value of the search parameter, and WRCRD1, WRCRD2, WRCRD3, and WRCRD4 to write the modified data set XS.C.MIN. The modified data thus replaces the original data. WSRCH then uses UDBHMG to update data set B.HOMOG. The maximum amount of POINTR container array space used is determined and printed.

For a buckling or dimension search, WSRCH again sets up needed arrays and calls RDGEOM to read data set GEOM. Then, to write the modified values on the data set, WSRCH calls WTBSQ (entry in RDGEOM) for a buckling search, or WTDIM (entry in RDGEOM) for a dimension search. Again, the amount of POINTR container array space used is given.

Following this, control is returned by subroutine D1DSCH to the main program of NUC004, which calls FREE to free storage and then returns control to the path which linked to the module.

4. D1DSCH (NUC004) Subprogram Descriptions

This section contains descriptions of all subprograms which are part of the D1DSCH module and which are not shared with the DIF1D module. The descriptions appear in alphabetical order by subprogram name.

Each subprogram description begins with a listing of the calling sequence of the subprogram, and of its entry points, if any. Subsection 1 is a list of the arguments of the subprogram and its entry points, in order of appearance, with the type of each argument, its dimensions in the subprogram if it is an array, and its definition or use. The four types of variables used are I*2, denoting an integer halfword (16 bits); I*4, denoting an integer fullword (32 bits); R*4, denoting a floating-point fullword; and R*8, denoting a floating-point doubleword (64 bits). Variables used as array dimensions are defined either as arguments or as members of common blocks.

When the character "*" appears as an argument, it indicates an optional return to a statement number specified in the calling subprogram. See the IBM FORTRAN manuals for further explanation.

Subsection 2 contains the names of all common blocks used by the subprogram. Each is followed by a list of the variables actually used from the common block within the subprogram, together with their types and definitions. The dimensions, if any, follow the variable name in parentheses.

Exceptions to this are the /BLKSTR/ and /SINGLE/ common blocks. For each, its presence is noted; for the /SINGLE/ block, the locations of variables used are listed, together with their local names and definitions.

Subsection 3 contains a listing of important variables which are local to the subprogram, in a format like that described for subsection 2.

Subsection 4 describes the functions performed by the subprogram in terms of the overall purpose of the module.

Subsection 5 lists all subprograms called by the subprogram or any of its entry points; subsection 6 lists the subprograms which call the subprogram or any of its entry points.

Subsection 7 describes error messages which can be generated by the subprogram, together with the action taken.

More detailed explanations of the contents of common blocks, and of the variable-dimensioned arrays stored in the large container array /BLKSTR/, can be found in sections 5 and 6.

Subroutine CHANGE (ICONV, *)

1. Arguments

Name	Type	Dimension	Definition
ICONV	I*4	--	Convergence flag 0-search has not converged 1-search has converged
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(2)	KEFF	Current value of k_{eff}, $k_{eff}^{(n)}$
FLT(52)	XNTH	Current value of search parameter, $X^{(n)}$
FLT(53)	XNM1	Previous value of search parameter, $X^{(n-1)}$
FLT(54)	XNM2	Previous value of search parameter, $X^{(n-2)}$
FLT(55)	DKDX	Slope of curve k_{eff} vs. X at the point $X^{(n)}$
FLT(56)	XL	Lower bound on search control parameter X
FLT(57)	XU	Upper bound on search control parameter X
FLT(58)	KZERO	Value of k_{eff} prescribed to be searched to, $k^{(0)}$
FLT(59)	EPSSRC	Convergence criterion ε_{SCH} on k_{eff} for search
FLT(60)	SKNTH	$k_{eff}^{(n)}$
FLT(61)	SKNM1	$k_{eff}^{(n-1)}$
FLT(62)	SKNM2	$k_{eff}^{(n-2)}$

Name	Local Name	Definition
FLT(65)	XBAR	Value of $\overline{X}^{(n)}$ used in actual search pass relation: $P^{(n)} = P^{(n-1)} + \overline{X}^{(n)} \delta P$
INT(82)	LORP	Interpolation type 1-linear 0 or 2-parabolic
INT(83)	NPASS	Number of current search pass

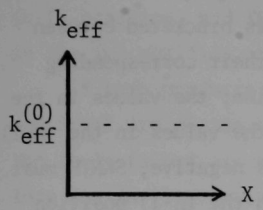
3. Local Variables

Name	Type	Definition
DKNTH	R*8	$k_{eff}^{(0)} - k_{eff}^{(n)}$
DKNM1	R*8	$k_{eff}^{(0)} - k_{eff}^{(n-1)}$
DKNM2	R*8	$k_{eff}^{(0)} - k_{eff}^{(n-2)}$
X(3)	R*8	Three values of search parameter X for parabolic interpolation
Y(3)	R*8	Three corresponding values of k_{eff} for parabolic interpolation
COEF(3)	R*8	Three values of parabolic interpolation coefficients, a, b, and c
DISCRM	R*8	$b^2 - 4\,ac$, the discriminant of the quadratic equation $ax^2+bx+c - k_{eff}^{(0)} = y$
ROOT1	R*8	First root of quadratic equation $ax^2+bx+c - k_{eff}^{(0)} = y$
ROOT2	R*8	Second root of quadratic equation $ax^2+bx+c - k_{eff}^{(0)} = y$
XNP1	R*8	New value of X, $X^{(n+1)}$, after parabolic interpolation
NPASS1	I*4	NPASS+1, number of next search pass

4. Functions and tasks performed by this subroutine

This subroutine computes the new value of the search control parameter X to be used on the next search pass. The task can be thought of as one of

plotting successive values of k_{eff} and X on a curve of k vs. X, attempting to solve for the value of X corresponding to $k_{eff}^{(0)}$, as shown in the diagram.



This is done by using input guesses to obtain the first two values of X, using the k-effective calculation to obtain the corresponding values of k_{eff}, and using linear interpolation on these points (X, k_{eff}) to obtain a third point. Hereafter, either linear extrapolation, using two points, or parabolic, using three points, can be used.

First, the current values of SKNTH, $k_{eff}^{(n)}$, and of DKNTH, $\delta k_{eff}^{(n)} = k_{eff}^{(0)} - k_{eff}^{(n)}$, are set up. If $\left| \frac{\delta k_{eff}^{(n)}}{k_{eff}^{(0)}} \right| \leq \varepsilon_{SCH}$, where ε_{SCH} is the search convergence criterion EPSSRC, the flag ICONV is set to 1, indicating convergence, and control is returned to the calling program. Otherwise, the value of X to be used on the next pass is sought.

If this is the first pass, this task is simple. First, XNTH becomes XNM1 (and correspondingly, SKNTH and DKNTH become SKNM1 and DKNM1). Then, if XNM2 was input, the new value of XNTH to be used becomes XNM2. Or, if DKDX, the slope of the curve k_{eff} vs. X at $X^{(0)}$, was input, the new value of XNTH is computed using linear interpolation, as $\frac{DKNM1}{DKDX}$ + XNM1.

On the second pass, linear interpolation is again used. XNM1, SKNM1, and DKNM1 become XNM2, SKNM2, and DKNM2; XNTH, SKNTH, and DKNTH become XNM1, SKNM1, and DKNM1; and the new value of XNTH is computed as $\frac{DKNM1}{DKDX}$ + XNM1, where DKDX is the ratio $\frac{(SKNM1 - SKNM2)}{(XNM1 - XNM2)}$, the slope of the curve between those points.

On any pass after the second, either linear or parabolic interpolation can be used, and in either case the decision must be made as to which of the old points (X, k_{eff}) should be saved. This decision is made in both cases using the values of DKNTH, DKNM1, and DKNM2, which by their signs tell on which side of the line through $k_{eff}^{(0)}$ the values of k_{eff} lie and by their magnitudes tell the distance of these values from the line through $k_{eff}^{(0)}$.

For linear interpolation, if the product DKNM2*DKNM1 is positive both SKNM2 and SKNM1 lie on the same side of the line through $k_{eff}^{(0)}$. The closer of them to the line through $k_{eff}^{(0)}$ is preserved in the (n-2) position, along

with the corresponding values of X and k_{eff}; the values in the (n) position are preserved in the (n-1) position; and the new value of XNTH is computed as above.

If the product DKNM2*DKNM1 is negative, the root is bracketed between them. Hence, if the product DKNM1*DKNTH is positive, their corresponding k_{eff}'s, SKNM1 and SKNTH, lie on the same side of the line; the values in the (n) position are preserved in the (n-1) position, and the values in the (n-2) position are maintained. If the product DKNM1*DKNTH is negative, SKNTH must be on the same side of the line as SKNM2; the values in the (n-1) position are preserved in (n-2) and the values in (n) are preserved in (n-1). In either case, having assigned these positions, the linear interpolation to locate XNTH is performed as above.

For parabolic interpolation, the values of the three existing X's are assigned to the X(3) array, with X(1) being the most recent, and the values of the three k_{eff}'s are similarly assigned to the Y(3) array. The subroutine PARAB is called to return the coefficients a, b, and c thus determined for the quadratic equation $ax^2+bx+c - k_{eff}^{(0)} = y$, if parabolic interpolation is feasible. If parabolic interpolation is not feasible because the points are too close, a message to that effect is printed, and linear interpolation is performed. Otherwise, the value of $k_{eff}^{(0)}$ is subtracted from the third coefficient c, to place the equation in standard quadratic form, and the discriminant DISCRM, b^2-4ac, is computed. If this is negative, again parabolic interpolation is not feasible, and linear is performed.

If the discriminant is zero or positive, the two roots are computed. If $b \leq 0$, ROOT1 is computed as $\frac{-b+\sqrt{b^2-4ac}}{2a}$; if $b > 0$, ROOT1 is $\frac{-b-\sqrt{b^2-4ac}}{2a}$. This ensures that accuracy will not be lost if b and the discriminant are nearly equal, by taking the larger of the two possibilities for the numerator of the fraction. If the resulting ROOT1 is zero, ROOT2 is of course equal; ROOT1 is preserved as XNP1. If ROOT1 is non-zero, ROOT2 is simply $\frac{c}{a \cdot \text{ROOT1}}$ from the quadratic equation; the closer of ROOT2 and ROOT1 to XNTH is preserved as XNP1.

The decision must then be made as to which two of the old points to preserve. This is done on the basis of two products, DKNTH*DKNM1, and DKNM1*DKNM2. If both are positive, the root is not bracketed; the latest two values are preserved, (n-1) in the (n-2) position, and (n) in the (n-1) position, and XNP1 becomes XNTH. If the first product is

positive and the second negative, the root is bracketed and SKNTH and SKNM1 are on the same side of the line through $k_{eff}^{(0)}$. If SKNTH is closer to the line, the values in (n-2) are maintained, the (n) values are moved into the (n-1) position, and XNTH becomes XNM1; if SKNM1 is closer, only the values in the (n) position are changed, and XNP1 is moved into XNTH.

If both products are negative, the root is again bracketed, and SKNTH and SKNM2 are on the same side of the line through $k_{eff}^{(0)}$. If SKNTH is closer, the (n-1) values are moved into the (n-2) position, the (n) values are moved into the (n-1) position, and XNTH becomes XNP1. If SKNM2 is closer, the (n-1) and (n-2) values are preserved in their locations, and XNTH is again set to XNP1.

Finally, if the first product is negative and the second positive, the root is bracketed with SKNM1 and SKNM2 on the same side of the line through $k_{eff}^{(0)}$. The closer of the two to the line is saved in the (n-2) position, the (n) values are moved into the (n-1) position, and XNTH becomes XNP1.

On any pass, after the new value of $X^{(n)}$ has been calculated, the value of the search parameter actually used in the search relation $P^{(n)} = P^{(n-1)} + \overline{X}^{(n)}\delta P$, which is $\overline{X}^{(n)} = X^{(n)} - X^{(n-1)}$, is computed, and the new data is printed. The subroutine then checks to ensure that the new $X^{(n)}$ is between the defined limits, XL $\leq$ XNTH $\leq$ XU, and if it is, returns control to the calling program.

5. Subprograms called by this subroutine

PAGHED
PARAB

6. Subprograms calling this subroutine

D1DSCH

7. Error messages generated by this subroutine

Message: *****BOUNDS ON CONTROL PARAMETER EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE CHANGE

Significance: The search control parameter computed for the next pass is not within the bounds set for the control parameter; the condition XL $\leq$ XNTH $\leq$ XU is not met.

Action taken: Execution is terminated.

Message: PARABOLIC EXTRAPOLATION NOT FEASIBLE--LINEAR EXTRAPOLATION USED

Significance: This is merely an information message. The user has specified parabolic extrapolation, but it is not feasible, either because the denominator used in solving for the parabolic fit by means of Cramer's rule is too small ($\leq 1.0 \cdot 10^{-8}$), indicating that accuracy might be better with linear extrapolation; because the quadratic coefficient a is near zero and should not be used as a divisor; or because the discriminant b^2-4ac of the quadratic equation $ax^2+bx+c - k_{eff}^{(0)} = y$ is less than zero.

Action taken: Linear interpolation is used.

Subroutine CRTIN1 (NUMMY, NLGTH, NCRIT)

Entry Points

Entry CRTOT1

1. Arguments

Name	Type	Dimension	Definition
NUMMY	I*4	NLGTH	Array containing second record of SP.CRIT for concentration search
NLGTH	I*4	--	Length of second record of SP.CRIT
NCRIT	I*4	--	Data set reference number for SP.CRIT

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine reads the second record of data set SP.CRIT into a dummy array for a concentration search when CRTIN1 is called, and rewrites the record from the array onto the data set when CRTOT1 is called.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

CRTIN1 is called by D1DSCH
CRTOT1 is called by D1DSCH

7. Error messages generated by this subroutine - None

Subroutine CRTIN2 (DUMMY, NL, NCRIT)

Entry Points

Entry CRTOT2

1. Arguments

Name	Type	Dimension	Definition
DUMMY	R*8	NL	Array containing second record of SP.CRIT for dimension or buckling search
NL	I*4	--	Length of array in second record of SP.CRIT
NCRIT	I*4	--	Data set reference number for SP.CRIT

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
NUM1	I*4	First integer in second record of SP.CRIT
NUM2	I*4	Second integer in second record of SP.CRIT

4. Functions and tasks performed by this subroutine

This subroutine reads the second record of data set SP.CRIT into two integer locations and a dummy floating-point array for a dimension or buckling

search when CRTIN2 is called, and rewrites the record from these locations onto the data set when CRTOT2 is called.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

CRTIN2 is called by D1DSCH
CRTOT2 is called by D1DSCH

7. Error messages generated by this subroutine - None

Subroutine DELALF

1. Arguments - None

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(3)	ALPHA	Inverse reactor time period α
FLT(64)	DALPHA	Alpha modifier $\delta\alpha$
FLT(65)	XNTH	Current value of search parameter $\overline{X}^{(n)}$ (XBAR in SINGLE)

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine modifies the value of the inverse reactor time period α using the current value of the search parameter $\overline{X}^{(n)}$ for an alpha search. α is modified according to the formula $\alpha^{(n)} = \alpha^{(n-1)} + \overline{X}^{(n)}\delta\alpha$ and the resulting value is printed.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

D1DSCH

7. Error messages generated by this subroutine - None

Subroutine DELBSQ (BSQ, DBSQ, REGHHT, EXTDIS, REGNAM, RMAX, *)

1. Arguments

Name	Type	Dimension	Definition
BSQ	R*8	RMAX*2	Buckling B^2 by region and direction
DBSQ	R*8	RMAX*NIL	Buckling modifiers δB^2 by region and direction
REGHHT	R*8	RMAX*2	Actual region half-heights $(H_L)_r$ by direction
EXTDIS	R*8	RMAX*2	Extrapolation distance $(H_r - (H_L)_r)$ by region and direction
REGNAM	R*8	RMAX	Region names
RMAX	I*4	--	Number of regions defined
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(31)	GEOM	Geometry type	'RECTSLAB' 'CYLSLAB' 'CYLINDER' 'SPHERE'

Name	Local Name	Definition
FLT(65)	XNTH	Current value of search parameter $\overline{X}^{(n)}$ (XBAR in SINGLE)
INT(89)	NIL	Number of finite directions defined

3. Local Variables

Name	Type	Definition
CSLAB	R*8	'CYLSLAB' for geometry type
DUMMY	R*8	Dummy variable used for checking for POINTR errors
BUCK2	R*8	$2 \cdot \sqrt{B^2}$ for recomputing actual half-height H_{Lr}

4. Functions and tasks performed by this subroutine

This subroutine modifies the buckling B^2 and the corresponding actual region half-heights $(H_L)_r$ using the current value of the search parameter $\overline{X}^{(n)}$ for a buckling search.

For any region r and direction n, if the buckling is zero (an original buckling of zero indicates that the reactor is infinite for that region and direction), no modification is performed. For a non-zero buckling, a modified value is computed according to the equation $B^{2(n)}_{r,n} = B^{2(n-1)}_{r,n} + \overline{X}^{(n)}\delta B^2_{r,n}$.

If the modified value of B^2 is zero, the region-half-height is set to .5 and the extrapolation distance to 0., according to the program conventions. This region will not be modified on successive passes.

If the modified value of B^2 is not zero, the corresponding value of the actual half-height of the reactor in that region and direction is computed. For a rectangular slab or a cylinder, the new actual half-height is

$(H_L)^{(n)}_{r,n} = \frac{\pi}{2B^{(n)}_{r,n}} - \left(H^{(0)}_{r,n} - (H_L)^{(0)}_{r,n}\right)$, where $B^{(n)}_{r,n} = \sqrt{B^{2(n)}_{r,n}}$, $\frac{\pi}{2H^{(n)}_{r,n}}$ is the new extrapolated half-height, and $\left(H^{(0)}_{r,n} - (H_L)^{(0)}_{r,n}\right)$ is the extrapolation distance, which remains constant throughout the problem. For a cylindrical slab, similarly, $(H_L)^{(n)}_r = \frac{\alpha}{B^{(n)}_r} - \left(H^{(0)}_r - (H_L)^{(0)}_r\right)$ where α is the first zero of the Bessel function J_0, $\alpha = 2.40482556$.

The modified values of the buckling and actual region half-height, along with the constant extrapolation distance, are then printed, along with the region name and the direction number, where the first direction is transverse and the second is axial.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)
DUMP (entry in POINTR)
ERRMSG

6. Subprograms calling this subroutine

D1DSCH

7. Error messages generated by this subroutine

Message: *****A NEGATIVE VALUE HAS BEEN COMPUTED FOR BUCKLING
FOR REGION aaaaaa DIRECTION iii
BUCKLING = -x.xxxxxxExxx

ERROR NUMBER 1 IN SUBROUTINE DELBSQ

Significance: The negative buckling value printed has been generated in the region named, where the direction number is 1 for the transverse direction and 2 for the axial direction.

Action taken: Execution is terminated.

Subroutine DELX (X, DX, DDX, IMAX, *)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	IMAX+1	Mesh line locations X
DX	R*8	IMAX	Mesh interval lengths ΔX
DDX	R*8	IMAX	Mesh interval length modifiers $\delta\Delta X$
IMAX	I*4	--	Number of mesh intervals defined
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(65)	XNTH	Current value of search parameter $\overline{X}^{(n)}$ (XBAR in SINGLE)

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy variable used for checking for POINTR errors
IMP1	I*4	IMAX+1 for number of mesh lines

4. Functions and tasks performed by this subroutine

This subroutine modifies the length of the X-dimension mesh intervals and the corresponding locations of mesh lines using the current value of the search parameter $\overline{X}^{(n)}$ for an X-dimension search. The intervals are modified according to the formula $\Delta X_i^{(n)} = \Delta X_i^{(n-1)} + \overline{X}^{(n)} \delta \Delta X_i$, and mesh line locations then become $X_{i+1}^{(n)} = X_i^{(n)} + \Delta X_i^{(n)}$. The routine then prints mesh line locations after modification.

5. Subprograms called by this subroutine

IPTERR (entry in POINTR)
ERRMSG

6. Subprograms calling this subroutine

D1DSCH

7. Error messages generated by this subroutine

Message: *****THE LENGTH OF THE iiiiTH INTERVAL IS NEGATIVE
INTERVAL LENGTH = -x.xxxxxxExxx

ERROR NUMBER 1 IN SUBROUTINE DELX

Significance: The negative interval length value printed has been generated in the interval whose number is given.

Action taken: Execution is terminated.

Subroutine DELXS (MODXS, FSCMP, DC, REM, NUFIS, CHI, SCTLIM, SCTLOC, SCTXS, FISXS, CMPNAM, CMAX, FSCMAX, CHIDIM, SCTDIM, GMAX, IERR, *)

1. Arguments

Name	Type	Dimension	Definition
MODXS	I*4	CMAX	Number of modifier composition for each composition
FSCMP	I*2	CMAX	Indices of fissionable compositions in arrays dimensioned FSCMAX
DC	R*8	CMAX*GMAX	Diffusion coefficients D_c^g
REM	R*8	CMAX*GMAX	Removal cross section σ_R
NUFIS	R*8	FSCMAX*GMAX	Nu times fission cross section $\nu \cdot \sigma_f$
CHI	R*8	CHIDIM*GMAX	Fission spectrum $\chi_c^{g'g}$
SCTLIM	I*2	2*CMAX	Maximum number of groups of upscattering and of downscattering for each composition
SCTLOC	I*2	CMAX*GMAX	Location of scattering band for each composition and group in scattering array SCTXS
SCTXS	R*8	SCTDIM*GMAX	Scattering cross section $\sigma_c^{g'g}$
FISXS	R*8	FSCMAX*GMAX	Fission cross section σ_f
CMPNAM	R*8	CMAX	Composition names
CMAX	I*4	--	Number of compositions defined
FSCMAX	I*4	--	Number of fissionable compositions defined
CHIDIM	I*4	--	First dimension of fission spectrum array CHI

Name	Type	Dimension	Definition
SCTDIM	I*4	--	First dimension of scattering array SCTXS
GMAX	I*4	--	Number of groups defined
IERR	I*4	--	Error flag 0-no error 1-6-a value or values of specified cross section became negative 1-diffusion coefficient 2-removal cross section 3-fission cross section 4-$\nu \cdot \sigma_f$ cross section 5-fission spectrum 6-scattering
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container; FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(65)	XNTH	Current value of search parameter $\overline{X}^{(n)}$ (XBAR in SINGLE)
INT(39)	MTRCHI	Fission spectrum array type 0-χ-vector 1-χ-matrix

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy variable used for checking for POINTR errors
C	I*4	Index on original compositions

Name	Type	Definition
MODC	I*4	Modifier composition for given composition C
NFSCMP	I*4	Position of composition C in fissionable cross section arrays
LIM1	I*4	Lower limit of scattering band for composition C, group G
LIM2	I*4	Upper limit of scattering band for composition C, group G
LIM1M	I*4	Lower limit of scattering band for composition MODC, group G
LIM2M	I*4	Upper limit of scattering band for composition MODC, group G

4. Functions and tasks performed by this subroutine

This subroutine modifies all cross sections of compositions being modified, using the current value of the search parameter $\overline{X}^{(n)}$, for a concentration search. If MODXS(C) is zero, no modification is performed for composition C.

For the removal cross section, the modification is performed simply as

$$\left((\sigma_R)^g_c\right)^{(n)} = \left((\sigma_R)^g_c\right)^{(n-1)} + \overline{X}^{(n)} (\sigma_R)^g_{mc}$$

for each group g, where c is the original composition and mc the modifier composition.

For the diffusion coefficient, the transport cross section is in effect modified:

$$\left(D^g_c\right)^{(n)} = \frac{\left(D^g_c\right)^{(n-1)} \cdot D^g_{mc}}{D^g_{mc} + \overline{X}\left(D^g_c\right)^{(n-1)}}$$

If the modifier composition is non-fissionable, no modification of fission-related cross sections is attempted. If the modifier is fissionable and the original composition is not, an error is flagged. For the

case where both are fissionable, modification of the fission cross section and of nu times the fission cross section is performed as follows:

$$\left((\sigma_f)_c^g\right)^{(n)} = \left((\sigma_f)_c^g\right)^{(n-1)} + \overline{X}^{(n)} (\sigma_f)_{mc}^g$$

$$\left((\nu\cdot\sigma_f)_c^g\right)^{(n)} = \left((\nu\cdot\sigma_f)_c^g\right)^{(n-1)} + \overline{X}^{(n)} (\nu\cdot\sigma_f)_{mc}^g \ .$$

Modification of the fission spectrum for fissionable compositions is somewhat more involved. For a χ-matrix $(\chi^{g'g})$, modification is performed over all groups g for a fixed g':

$$\left(\chi_c^{g'g}\right)^{(n)} = \sum_{g=1}^{G} \frac{\left(\chi_c^{g'g}\right)^{(n-1)} \cdot \left[\left((\nu\cdot\sigma_f)_c^{g'}\right)^{(n)} - \overline{X}^{(n)}\left((\nu\cdot\sigma_f)_c^{g'}\right)^{(n)}\right] + \overline{X}^{(n)}\left(\chi_{mc}^{g'g}\right)(\nu\cdot\sigma_f)_{mc}^{g'}}{\left((\nu\cdot\sigma_f)_c^{g'}\right)^{(n)}}$$

Here, the term in brackets is clearly the $\nu\cdot\sigma_f$ cross section before modification on this pass, $\left((\nu\cdot\sigma_f)_c^g\right)^{(n-1)}$. For a χ-vector, the calculation is similarly performed over all groups g for a fixed g':

$$\left(\chi_c^{g}\right)^{(n)} = \sum_{g=1}^{G} \frac{\left(\chi_c^{g}\right)^{(n-1)} \cdot \left[\left((\nu\cdot\sigma_f)_c^{g'}\right)^{(n)} - \overline{X}^{(n)}\left((\nu\cdot\sigma_f)_c^{g'}\right)^{(n)}\right] + \overline{X}^{(n)}\left(\chi_{mc}^{g}\right)(\nu\cdot\sigma_f)_{mc}^{g'}}{\left((\nu\cdot\sigma_f)_c^{g'}\right)^{(n)}}$$

Finally, for all modified compositions the scattering cross section is modified within the band width specified for each composition and group. The maximum number of groups of downscattering and of upscattering for the original composition must be equal to or greater than the corresponding numbers for the modifier composition, or the modification cannot be performed. The equation for the modification is simply

$$\left(\sigma_c^{g'g}\right)^{(n)} = \left(\sigma_c^{g'g}\right)^{(n-1)} + \overline{X}^{(n)}\left(\sigma_{mc}^{g'g}\right) ,$$

where g' runs along the band width of the modifier composition.

If any value or values of any cross section type become negative, the IERR flag is set to the appropriate value. If more than one cross section type becomes negative, the IERR flag is set to the number of the highest-numbered type. A non-zero value of the IERR flag will cause the composition data to be printed and the search to be terminated.

5. Subprograms called by this subroutine

ERRMSG

6. Subprograms calling this subroutine

D1DSCH

7. Error messages generated by this subroutine

Message: *****NON-FISSIONABLE COMPOSITION CANNOT BE MODIFIED BY A FISSIONABLE COMPOSITION
NON-FISSIONABLE COMPOSITION CAN BE MADE TO APPEAR FISSIONABLE
BY ADDING ZERO VOLUME-FRACTION OF A FISSIONABLE MATERIAL
ON TYPE 14 CARD OF DATA SET A.NIP

ERROR NUMBER 1 IN SUBROUTINE DELXS

Significance: Self-explanatory. Storage space for fission-related cross-sections is not allocated unless the composition appears fissionable.

Action taken: Execution is terminated.

Message: *****NUMBERS OF GROUPS OF DOWN SCATTERING AND OF UP SCATTERING FOR ORIGINAL COMPOSITION
MUST BE EQUAL TO OR GREATER THAN THOSE FOR MODIFIER COMPOSITION
THIS CAN BE ACCOMPLISHED BY ADDING ZERO VOLUME-FRACTION OF
MODIFIER COMPOSITION TO ORIGINAL ON TYPE 14 CARD OF DATA SET A.NIP

ERROR NUMBER 2 IN SUBROUTINE DELXS

Significance: Self-explanatory. Storage space for scattering for a given composition and group is only as wide as the band width.

Action taken: Execution is terminated.

Subroutine D1DSCH (IQUIT)

1. Arguments

Name	Type	Dimension	Definition	
IQUIT	I*4	--	Flag returned to path level	0-execution of NUC004 successful 1-execution of NUC004 terminated abnormally

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition		
FLT(1)	PROB	Problem type		'REAL' 'ADJOINT'' 'BOTH'
FLT(51)	SEARCH	Search type	'CONC' 'XDIM' 'YDIM'	'BSQ' 'ALPHA' 'XYDIM'
FLT(52)	XNTH	Current value of search control parameter, $X^{(n)}$		
FLT(53)	XNM1	Previous value of search control parameter, $X^{(n-1)}$		

Name	Local Name	Definition
FLT(54)	XNM2	Previous value of search control parameter, $X^{(n-2)}$
FLT(55)	DKDX	Slope of curve k_{eff} vs. X at $X^{(n)}$
FLT(56)	XL	Lower bound on search control parameter X
FLT(57)	XU	Upper bound on search control parameter X
FLT(58)	KZERO	Value of k_{eff} prescribed to be searched to, $k_{eff}^{(0)}$
FLT(59)	EPSSRC	Convergence criterion ε_{SCH} on k_{eff} for search
FLT(66)	DMOD	Modifier for search when only one is necessary
INT(7)	KOUT	Output print flag 0-don't print search results > 0-print search results
INT(11)	NMAX	Number of outer iterations in k-effective calculation
INT(19)	MAXBLK	Size of POINTR container array in bulk storage
INT(31)	RMAX	Number of regions defined
INT(32)	IMAX	Number of mesh intervals defined
INT(35)	CMAX	Number of compositions defined
INT(36)	FSCMAX	Number of fissionable compositions defined
INT(37)	GMAX	Number of groups defined
INT(40)	CHIDIM	First dimension of fission spectrum array CHI

Name	Local Name	Definition	
INT(41)	SCTDIM	First dimension of scattering array SCTXS	
INT(58)	NRST	Number of first outer iteration at which fluxes are to be written on FR.D1 or FA.D1 in k-effective calculation	
INT(81)	NPMAX	Maximum number of search passes	
INT(82)	INTERP	Interpolation type	1-linear 0 or 2-parabolic
INT(83)	NPASS	Number of current search pass	
INT(84)	KSRCH	Search type number	1-concentration 2-X-dimension 3-Y-dimension 4-buckling 5-α 6-X- and Y-dimension
INT(88)	NFIN	Number of finite geometry type	
INT(89)	NIL	Number of finite directions defined	
INT(94)	KFLAG	Type of variable prescribed to be searched to	0-k_{eff} 1-α

3. Local Variables

Name	Type	Definition
SPCRIT	R*8	'SP.CRIT' for SNIFF call
XDIM	R*8	'XDIM' for search type
CONC	R*8	'CONC' for search type
BSQ	R*8	'BSQ' for search type
ADJ	R*8	'ADJOINT' for problem type
REAL	R*8	'REAL' for problem type
BOTH	R*8	'BOTH' for problem type
TIMETL	R*4	Time remaining at beginning of module; total time spent in module in minutes
DUMMY	R*4	Dummy argument to TLEFT

Name	Type	Definition	
IPROB	I*4	Problem type flag	0-'REAL' or 'ADJOINT' 1-'BOTH'
NCRIT	I*4	Data set reference number for SP.CRIT	
N1	I*4	First data item in SP.CRIT (same as KSRCH)	
N2	I*4	Second data item in SP.CRIT (data set convergence flag)	0-not converged 1-converged
LSPR	I*4	Length of specification record in SP.CRIT	
IERR	I*4	Error flag returned from DELXS	0-no error > 0-a value or values of some cross section became negative
IX	I*4	Convergence flag returned from CHANGE	0-not converged 1-converged
N3	I*4	Third through sixteenth data items in SP.CRIT	
N4	I*4		
N5	I*4		
N6	I*4		
D7	R*8		
D8	R*8		
D9	R*8		
D10	R*8		
D11	R*8		
D12	R*8		
D13	R*8		
D14	R*8		
D15	R*8		
D16	R*8		

Name	Type	Definition
LAST	I*4	Last location used in POINTR core container array
LASTB	I*4	Last location used in POINTR bulk container array

4. Functions and tasks performed by this subroutine

This subroutine is the driver for the ARC System one-dimensional diffusion theory search calculation module, NUC004.

It first reads the first record of data set SP.CRIT and then calls D1DIN to perform the input of all other data sets required by the module. (The input overlay is shared with the one-dimensional diffusion theory k-effective and source calculation module, NUC002.) It then uses IPROB to save the problem type if it is 'BOTH', and resets PROB so that to the calculation and output sections the type appears only as 'REAL' or 'ADJOINT'.

D1DSCH then calls SRCHIN to modify for use by this module the data from the first record of SP.CRIT and to read the modifier data from the second record of SP.CRIT, and sets up a dummy storage array (NUMMY for a concentration search, DUMMY for a buckling or dimension search) to be used in copying the second record of SP.CRIT after each search pass. It determines and prints the amount of POINTR container array space used.

The search pass execution is then begun. For each pass, the appropriate modifying subroutine is called: DELX for an X-dimension search, DELXS for a concentration search, DELBSQ for a buckling search, or DELALF for an alpha search. (For a concentration search, if any value or values of any cross section type or types become negative, PCOMP is called immediately and execution is terminated.)

D1DCAL, the k-effective calculation driver (shared with NUC002), is then called to perform a k-effective calculation using the modified data. The search monitoring routine, CHANGE, is then called to compute monitoring information and the new search modifier X and to test for convergence.

The data set SP.CRIT is then updated; the only values altered are N2, N5, D11, D12, D13, and D14, which are replaced by IX, NPASS, XNTH, XNM1, XNM2, and DKDX, respectively.

When the search has converged, or when the maximum number of search iterations has been exceeded, the final version of modified data is printed

if KOUT > 0 by calling PGEOM for a buckling or dimension search or PCOMP for a concentration search, and the output section, D1DOUT (also shared with NUC002) is called. WIPE is called to remove the output arrays from the POINTR containers, and PURGE and PURGEB are called to pack the containers.

If the original problem type was 'BOTH', FLXIN is called to read a user-supplied adjoint flux guess. If no adjoint flux guess is provided, the latest real flux is used as the initial guess for the adjoint calculation. D1DCAL and D1DOUT are then called again to perform an adjoint calculation. In any case, WSRCH (shared with the two-dimensional diffusion theory search module, NUC006) is then called to rewrite the data sets modified by the search if desired: XS.C.MIN and B.HOMOG for a concentration search, or GEOM for a buckling or dimension search.

5. Subprograms called by this subroutine

TLEFT (IBM routine)	CHANGE
SNIFF (ARC System module)	CRTIN1
PUTPNT (entry in POINTR)	CRTOT1 (entry in CRTIN1)
IGET (entry in POINTR)	CRTIN2
WIPOUT (entry in POINTR)	CRTOT2 (entry in CRTIN2)
D1DIN	PCOMP
SRCHIN	PGEOM
ILAST (entry in POINTR)	D1DOUT
ILASTB (entry in POINTR)	WIPE
DELX	PURGE (entry in POINTR)
DELXS	PURGEB (entry in POINTR)
DELBSQ	WSRCH
DELALF	ERRMSG
D1DCAL	FLXIN

6. Programs calling this subroutine

Main program for module NUC004

7. Error messages generated by this subroutine

Message: *****DATA SET SP.CRIT NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE D1DSCH

Significance: This data set could not be found by the ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****MAXIMUM NUMBER OF CONTROL PASSES EXCEEDED DURING SEARCH

ERROR NUMBER 2 IN SUBROUTINE D1DSCH

Significance: The number of search passes NPMAX defined in data set SP.CRIT has been exceeded, and k_{eff} has not converged according to the search criterion.

Action taken: Execution is continued as through problem had converged.

Message: *****ERROR ENCOUNTERED IN WRITING DATA SETS MODIFIED BY SEARCH

ERROR NUMBER 2 IN SUBROUTINE D1DSCH

Significance: An error was encountered in updating either XS.C.MIN and B.HOMOG or GEOM, depending on the type of search, after the search.

Action taken: Execution is terminated.

Message: *****EXECUTION OF MODULE NUC004 HAS BEEN TERMINATED

Significance: An error has been encountered in the execution of the module which made successful execution of the module impossible.

Action taken: The error flag IQUIT is set to 1 and control is returned to the main program NUC004.

Main Program for NUC004

1. Arguments - None

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this program

This program is the dummy driver for the ARC System one-dimensional diffusion theory search calculation module, NUC004. It calls subroutine D1DSCH, which is the actual driver for the module, and FREE, an entry point in POINTR which frees storage space allocated in bulk (LCS) and main storage by the module. It also serves to pass the error flag from D1DSCH to the path level.

5. Subprograms called by this program

D1DSCH
FREE (entry in POINTR)

6. Programs calling this program

This program is linked to by standard path STP001 and by any other paths which use the ARC System one-dimensional diffusion theory search capability. The statement used to link to this program is CALL LINK('NUC004', IQUIT).

7. Error messages generated by this program

No error messages are actually generated by this program. However, the error flag IQUIT is passed through this routine to the path from the driver subroutine D1DSCH. IQUIT has a value of 0 if execution of the search was successful, and a value of 1 if execution of the module was terminated due to some error condition. The path may test this flag before continuing to execute other modules dependent on the result of this module.

Subroutine PARAB (X, Y, COEF, *)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	3	Three values of X
Y	R*8	3	Three corresponding values of k_{eff}
COEF	R*8	3	Three quadratic coefficients returned
*	--	--	Return used if parabolic interpolation is not feasible

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
DENOM	R*8	Denominator determinant for Cramer's rule
A	R*8	COEF(1)
B	R*8	COEF(2)
C	R*8	COEF(3)

4. Functions and tasks performed by this subroutine

This subroutine computes the coefficients a, b, and c of the quadratic equation $y = ax^2+bx+c$ to give a parabolic fit for the three points (x_1,y_1), (x_2,y_2), and (x_3,y_3), using Cramer's rule.

The determinant forming the denominator of the solutions by Cramer's rule is first calculated:

$$\text{DENOM} = \begin{vmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{vmatrix} .$$

If the value of this determinant is less than 1.0×10^{-8}, a parabolic fit is not calculated, and the first return is used. If this denominator is within range, computation of the coefficients a, b, and c is made according to the equations

$$a = \frac{\begin{vmatrix} y_1 & x_1 & 1 \\ y_2 & x_2 & 1 \\ y_3 & x_3 & 1 \end{vmatrix}}{\text{DENOM}}, \quad b = \frac{\begin{vmatrix} x_1^2 & y_1 & 1 \\ x_2^2 & y_2 & 1 \\ x_3^2 & y_3 & 1 \end{vmatrix}}{\text{DENOM}}, \quad \text{and } c = \frac{\begin{vmatrix} x_1^2 & x_1 & y_1 \\ x_2^2 & x_2 & y_2 \\ x_3^2 & x_3 & y_3 \end{vmatrix}}{\text{DENOM}} .$$

Again, if the value of the first coefficient a is less than 1.0×10^{-8}, the first return is used, since this would involve dividing by a small number in the calculation of the roots of the resulting quadratic equation.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

CHANGE

7. Error messages generated by this subroutine - None

Subroutine RCORD1 (CNAME, N1, *)

1. Arguments

Name	Type	Dimension	Definition
CNAME	R*8	NCMP	Composition names
N1	I*4	--	Data set reference number for XS.C.MIN
*	--	--	Error return

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
N3	I*4	Number of compositions defined, NCMP (first data item on record 1 of XS.C.MIN)
N4	I*4	Second through seventh data items on record 1 of data set XS.C.MIN
N5	I*4	
N6	I*4	
N7	I*4	
N8	I*4	
N9	I*4	

4. Functions and tasks performed by this subroutine

This subroutine reads the composition names from the first record of

XS.C.MIN. The other data items have already been preserved in the subroutine WSRCH.

5. Subprograms called by this subroutine

ERRMSG

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine

Message: *****ERROR IN READING RECORD 1 OF XS.C.MIN ON DSRN iii

ERROR NUMBER 1 IN SUBROUTINE RCORD1

Significance: An error was encountered in attempting to read the first record of this data set.

Action taken: Execution is terminated.

Subroutine RCORD2 (CHI, ISCHI, NGROUP, N1, *)

1. Arguments

Name	Type	Dimension	Definition
CHI	R*8	ISCHI*NGROUP	Set-wide fission spectrum $\chi^{g'g}$
ISCHI	I*4	--	Set-wide fission spectrum flag 0-no set-wide χ 1-set-wide χ-vector NGROUP-set-wide χ-matrix
NGROUP	I*4	--	Number of groups defined
N1	I*4	--	Data set reference number for XS.C.MIN
*	--	--	Error return

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine reads the set-wide fission spectrum $\chi^{g'g}$, if one is present, from record 2 of data set XS.C.MIN.

5. Subprograms called by this subroutine

ERRMSG

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine

Message: *****ERROR IN READING RECORD 2 OF XS.C.MIN ON DSRN iii

ERROR NUMBER 1 IN SUBROUTINE RCORD2

Significance: An error was encountered in attempting to read the second record of this data set.

Action taken: Execution is terminated.

Subroutine RCORD3 (NUP, NDN, ICHI, NCMP, NGROUP, C, F, N1, *)

1. Arguments

Name	Type	Dimension	Definition
NUP	I*4	NCMP*NGROUP	Number of groups of upscattering for each composition and group
NDN	I*4	NCMP*NGROUP	Number of groups of downscattering for each composition and group
ICHI	I*4	NCMP	Composition fission spectrum flag -1-uses set-wide χ 0-non-fissionable 1-uses own χ-vector NGROUP-uses own χ-matrix

Name	Type	Dimension	Definition
NCMP	I*4	--	Number of compositions defined
C	I*4	--	Composition number
F	I*4	--	Fissionable composition index for composition
N1	I*4	--	Data set reference number for XS.C.MIN
*	--	--	Error return

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

For each composition C, this subroutine reads the corresponding record type 3 of XS.C.MIN, containing the composition fission spectrum flag and the number of groups of upscattering and of downscattering for each group for this composition. If ICHI is non-zero, indicating that the composition is fissionable, F is updated to F+1.

5. Subprograms called by this subroutine

ERRMSG

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine

Message: *****ERROR IN READING RECORD 3 OF XS.C.MIN FOR COMPOSITION iii
ON DSRN iii

ERROR NUMBER 1 IN SUBROUTINE RCORD3

Significance: An error was encountered in attempting to read record type 3 of this data set for the composition whose number is given.

Action taken: Execution is terminated.

Subroutine RCORD4 (XA, XTOT, REM, XTR, FISXS, NUFIS, CCHI, SCTXS, PXNF, DNCHI, NUP, NDN, ICHI, CHI, NCMP, NGROUP, NFCMP, C, G, F, N1, ISCHI)

1. Arguments

Name	Type	Dimension	Definition
XA	R*8	NCMP*NGROUP	Absorption cross section
XTOT	R*8	NCMP*NGROUP	Total cross section
REM	R*8	NCMP*NGROUP	Removal cross section
XTR	R*8	NCMP*NGROUP	Transport cross section
FISXS	R*8	NFCMP*NGROUP	Fission cross section
NUFIS	R*8	NFCMP*NGROUP	Nu times fission cross section
CCHI	R*8	NFCMP*NGROUP* NGROUP	Fission spectrum, $\chi^{g'g}$
SCTXS	R*8	NCMP*NGROUP* MAXSCT	Total scattering into given group
PXNF	R*8	NFCMP*NGROUP if NFAM > 0 0 if NFAM = 0	Number of prompt neutrons per fission times fission cross section
DNCHI	R*8	NFCMP*NGROUP* NFAM	Delayed neutron fission spectrum
NUP	I*4	NCMP*NGROUP	Number of groups of upscattering for each composition and group
NDN	I*4	NCMP*NGROUP	Number of groups of downscattering for each composition and group
ICHI	I*4	NCMP	Composition fission spectrum flag -1-uses set-wide χ 0-non-fissionable 1-uses own χ-vector NGROUP-uses own χ-matrix
CHI	R*8	ISCHI*NGROUP	Set-wide fission spectrum $\chi^{g'g}$, if present
NCMP	I*4	--	Number of compositions defined

Name	Type	Dimension	Definition
NGROUP	I*4	--	Number of groups defined
NFCMP	I*4	--	Number of fissionable compositions defined
C	I*4	--	Composition number
G	I*4	--	Group number
F	I*4	--	Fissionable composition index for composition C
N1	I*4	--	Data set reference number for XS.C.MIN
ISCHI	I*4	--	Set-wide fission spectrum flag: 0-no set-wide χ; 1-set-wide χ-vector; NGROUP-set-wide χ-matrix
*	--	--	Error return

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
ICHC	I*4	ICHI for given composition C
NSCAT	I*4	Scattering band width for given composition C

4. Functions and tasks performed by this subroutine

For a given group G and composition C, this subroutine reads the group record, record type 4 of data set XS.C.MIN, according to the proper format. Formats vary according to whether the composition is non-fissionable or fissionable, whether it uses the set-wide fission spectrum or is assigned its own, and whether prompt and delayed neutron data is present. All data items are read in, and all will be modified for modified compositions. For a composition using the set-wide fission spectrum, the corresponding locations in the CCHI array are set equal to the set-wide fission spectrum CHI.

5. Subprograms called by this subroutine

ERRMSG

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine

Message: *****iiiTH GROUP RECORD FOR COMPOSITION NO iiii IS OUT OF ORDER
GROUP NUMBER ON RECORD IS iiii

ERROR NUMBER 1 IN SUBROUTINE RCORD4

Significance: In reading the group records for the composition whose number is given, an out-of-order record was encountered. The first number is that of the group which was expected; the last number is the group number appearing on the record.

Action taken: Execution is terminated.

Message: *****ERROR IN READING RECORD 4 OF XS.C.MIN FOR COMPOSITION iii
AND GROUP iiii ON DSRN iii

ERROR NUMBER 2 IN SUBROUTINE RCORD4

Significance: An error was encountered in attempting to read record type 4 of this data set for the composition and group whose numbers are given.

Action taken: Execution is terminated.

Message: *****COMPOSITION NO iii USES A SET-WIDE CHI
BUT NO SET-WIDE CHI IS GIVEN

ERROR NUMBER 4 IN SUBROUTINE RCORD4

Significance: No set-wide fission spectrum is defined on data set XS.C.MIN, but the composition whose number is given expects one.

Action taken: Execution is terminated.

Subroutine RDBSQ (DBSQ, RMAX, NCRIT, REGNAM, *)

1. Arguments

Name	Type	Dimension	Definition
DBSQ	R*8	RMAX*NIL	Buckling modifiers δB^2
RMAX	I*4	--	Number of regions defined
NCRIT	I*4	--	Data set reference number for SP.CRIT
REGNAM	R*8	RMAX	Region names
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
INT(89)	NIL	Number of finite directions defined

3. Local Variables

Name	Type	Definition
RMX1	I*4	Number of regions defined in SP.CRIT
NIL1	I*4	Number of finite directions defined in SP.CRIT

4. Functions and tasks performed by this subroutine

This subroutine reads the second record of data set SP.CRIT, defining the modifiers DBSQ of the buckling by region and direction BSQ, for a buckling search (record type 4 of SP.CRIT) and prints all modifiers, along with the names of the regions modified.

5. Subprograms called by this subroutine

ERRMSG

6. Subprograms calling this subroutine

SRCHIN

7. Error messages generated by this subroutine

Message: *****SP.CRIT B**2 MODIFIER DATA RECORD IS WRONG LENGTH
RMX1 = iiii NIL1 = iiii

ERROR NUMBER 1 IN SUBROUTINE RDBSQ

Significance: One or both of the numbers determining the length of the buckling modifier data record does not agree with the corresponding number defined for the problem in data set GEOM; RMX1 does not agree with RMAX, and/or NIL1 does not agree with NIL.

Action taken: Execution is terminated.

Subroutine RDCONC (MODXS, CMPNAM, CMAX, NCRIT, *)

1. Arguments

Name	Type	Dimension	Definition
MODXS	I*4	CMAX	Number of modifier composition for each composition
CMPNAM	R*8	CMAX	Composition names
CMAX	I*4	--	Number of compositions defined
NCRIT	I*4	--	Data set reference number for SP.CRIT
*	--	--	Error return (not used)

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine reads the second record of data set SP.CRIT, containing the numbers MODXS of the compositions modifying each composition, for

a concentration search (record type 2 of SP.CRIT) and prints the names of all compositions modified, along with the names of the modifying compositions. For any composition C, if MODXS(C) is zero, the composition C is not modified in this problem.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

   SRCHIN

7. Error messages generated by this subroutine - None

Subroutine RDDDX (DDX, IMAX, NCRIT, *)

1. Arguments

Name	Type	Dimension	Definition
DDX	R*8	IMAX	Mesh interval modifiers, $\delta\Delta X$
IMAX	I*4	--	Number of mesh intervals defined
NCRIT	I*4	--	Data set reference number for SP.CRIT
*	--	--	Error return

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition	
FLT(31)	GEOM	Geometry type	'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'

3. Local Variables

Name	Type	Definition
RSLAB	R*8	'RECTSLAB' for geometry type
CYL	R*8	'CYLINDER' for geometry type
SPH	R*8	'SPHERE' for geometry type
CSLAB	R*8	'CYLSLAB' for geometry type
IMX1	I*4	Number of mesh intervals defined in SP.CRIT
NDUM	I*4	Dummy variable used to read number of j-direction mesh intervals

4. Functions and tasks performed by this subroutine

This subroutine reads the second record of data set SP.CRIT, containing the modifiers DDX of the mesh intervals DX, for an X-dimension search (record type 3 of SP.CRIT) and prints all non-zero modifiers, along with the numbers of the intervals to be modified.

5. Subprograms called by this subroutine

ERRMSG

6. Subprograms calling this subroutine

SRCHIN

7. Error messages generated by this subroutine

Message: *****SP.CRIT DX MODIFIER DATA RECORD IS WRONG LENGTH
IMX1=iiii

ERROR NUMBER 1 IN SUBROUTINE RDDDX

Significance: The number of mesh intervals IMX1 defined in data set SP.CRIT does not agree with the number IMAX defined for the problem in data set GEOM.

Action taken: Execution is terminated.

Subroutine RDGEOM (XMESH, YMESH, MR, MC, NC, NRA, NSB, B, H, EX, RNAME, CNAME, AREA, ZONE, NPTI, NPTJ, NINTI, NINTJ, NREG, NCMP, NAREA, LAREA, NZONE, LZONE, NIL, N3)

Entry Points

Entry WTBSQ (B, H, EX)

Entry WTDIM (XMESH, YMESH)

1. Arguments

Name	Type	Dimension	Definition
XMESH	R*8	NPTI	I-direction mesh line locations (X in BLK array for WTDIM entry for X-dimension or XY-dimension search)
YMESH	R*8	NPTJ	J-direction mesh line locations (Y in BLK array for WTDIM entry for Y-dimension or XY-dimension search)
MR	I*4	NINTI*NINTJ	Region-to-mesh correspondence
MC	I*4	NINTI*NINTJ	Composition-to-mesh correspondence
NC	I*4	NREG	Composition-to-region correspondence
NRA	I*4	LAREA	Entire contents of record type 6 (area definition)
NSB	I*4	LZONE	Entire contents of record type 7 (zone definition)
B	R*8	NREG*NIL	Buckling B^2 by region and direction (BSQ in BLK array for WTBSQ entry for buckling search)
H	R*8	NREG*NIL	Actual half-height $(H_L)_r$ by region and direction (REGHHT in BLK array for WTBSQ entry for buckling search)
EX	R*8	NREG*NIL	Extrapolated half-height H_r by region and direction (EXTDIS in BLK array for WTBSQ entry for buckling search)
RNAME	R*8	NREG	Region names

Name	Type	Dimension	Definition
CNAME	R*8	NCMP	Composition names
AREA	R*8	NAREA	Area names
ZONE	R*8	NZONE	Zone names
NPTI	I*4	--	Number of I-direction mesh lines defined
NPTJ	I*4	--	Number of J-direction mesh lines defined
NINTI	I*4	--	Number of I-direction mesh intervals defined
NINTJ	I*4	--	Number of J-direction mesh intervals defined
NREG	I*4	--	Number of regions defined
NCMP	I*4	--	Number of compositions defined
NAREA	I*4	--	Number of areas defined
LAREA	I*4	--	Length of area record (type 6)
NZONE	I*4	--	Number of zones defined
LZONE	I*4	--	Length of zone record (type 7)
NIL	I*4	--	Number of finite directions defined
N3	I*4	--	Data set reference number for GEOM

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

RDGEOM reads whatever record types 2-9 are present from data set GEOM for an X-dimension, Y-dimension, XY-dimension, or buckling search.

Entry WTBSQ is used only after a buckling search. Its arguments are the modified buckling data produced by the program. The extrapolated half-height used by the data set is set up as the sum of the modified actual half-height and the original extrapolation distance used by the program, and the data set is rewritten.

Entry WTDIM is used after a dimension search. For an X-dimension search, the contents of XMESH are the values modified by the program; for a Y-dimension search, YMESH is replaced by the corresponding program array; and for an XY-dimension search, both sets of mesh lines contain the new values. The data set is rewritten using these values.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

RDGEOM is called by WSRCH
WTBSQ is called by WSRCH
WTDIM is called by WSRCH

7. Error messages generated by this subroutine - None

Subroutine SRCHIN (NCRIT, *)

1. Arguments

Name	Type	Dimension	Definition
NCRIT	I*4	--	Data set reference number for SP.CRIT
*	--	--	Error return

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(3)	ALPHA	Inverse reactor time period α
FLT(51)	SEARCH	Search type 'CONC' 'BSQ' 'XDIM' 'ALPHA' 'YDIM' 'XYDIM'
FLT(52)	XNTH	Current value of seach control parameter $X^{(n)}$
FLT(54)	XNM2	Previous value of search control parameter, $X^{(n-2)}$
FLT(55)	DKDX	Slope of curve k_{eff} vs. X at $X^{(0)}$
FLT(56)	XL	Lower bound on search control parameter X
FLT(57)	XU	Upper bound on search control parameter X
FLT(58)	KZERO	Value of k_{eff} prescribed to be searched to, $k_{eff}^{(0)}$
FLT(59)	EPSSRC	Convergence criterion ε_{SCH} on k_{eff} for search
FLT(64)	DALPHA	Modifier for alpha search, $\delta\alpha$
FLT(65)	XBAR	Value of $\overline{X}^{(n)}$ used in previous search pass: $P^{(n)} = P(n-1) + \overline{X}^{(n)} \cdot \delta P$
FLT(66)	DMOD	Modifier for search when only one is necessary
INT(2)	KINOPT	Kinetics option 0 - no kinetics 2 - delayed data for alpha calculation

Name	Local Name	Definition
INT(31)	RMAX	Number of regions defined
INT(32)	IMAX	Number of mesh intervals defined
INT(35)	CMAX	Number of compositions defined
INT(81)	NPMAX	Maximum number of search passes
INT(82)	INTERP	Interpolation type 1 - linear 0 or 2 - parabolic
INT(83)	NPASS	Number of current search pass
INT(84)	KSRCH	Number of search type 1 - concentration 2 - X-dimension 3 - Y-dimension 4 - buckling 5 - α 6 - X- and Y-dimension
INT(89)	NIL	Number of directions of finite geometry defined
INT(94)	KFLAG	Type of variable prescribed to be searched to 0 - k_{eff} 1 - α

3. Local Variables

Name	Type	Definition
SCHTYP(6)	R*8	Six search types, corresponding to values of KSRCH (6A8)
XDIM	R*8	'XDIM' for search type
YDIM	R*8	'YDIM' for search type
XYDIM	R*8	'XYDIM' for search type
CONC	R*8	'CONC' for search type
BSQ	R*8	'BSQ' for search type

Name	Type	Definition	
ALF	R*8	'ALPHA' for search type	
PRES(2)	R*8	Name of variable prescribed to be searched to	'K-EFF' 'ALPHA'

4. Functions and tasks performed by this subroutine

This subroutine controls the modification of data in the first record of data set SP.CRIT to the form in which it is used by the one-dimensional diffusion theory search calculation, and the reading of the modifier data from the second record of SP.CRIT, if present.

SRCHIN first sets up SEARCH as SCHTYP(KSRCH) and initializes XBAR as XNTH. It then prints the type of variable prescribed, k_{eff} or α. If α is prescribed, ALPHA is set to the value of α prescribed, and KZERO, the value of k_{eff} to be searched to, is set to 1.0. Following this, the remainder of the description of the problem contained in the first record of SP.CRIT is printed.

For an X-dimension search, SRCHIN then calls RDDDX to read the DX modifiers, δDX, from the second record of SP.CRIT; for a concentration search, RDCONC is called to read the number of the modifier composition for each composition; and for a buckling search, RDBSQ is called to read the B^2 modifiers, δB^2. For an alpha search, the alpha modifier $\delta\alpha$ is set up from the value of DMOD in the first record of SP.CRIT.

5. Subprograms called by this subroutine

PAGHED
PUTPNT (entry in POINTR)
IGET (entry in POINTR)
RDDDX
RDCONC
RDBSQ
ERRMSG

6. Subprograms calling this subroutine

DIDSCH

7. Error messages generated by this subroutine

Message: *****ALPHA SEARCH TYPE AND SEARCH TO PRESCRIBED ALPHA HAVE BEEN SPECIFIED

ERROR NUMBER 1 IN SUBROUTINE SRCHIN

Significance: A prescribed value of alpha has been input, and the search quantity to be modified is also alpha.

Action taken: Execution is terminated.

Message: *****SEARCH TYPE NOT RECOGNIZED. TYPE=iiii

ERROR NUMBER 2 IN SUBROUTINE SRCHIN

Significance: A search type KSRCH not between 1 and 6, inclusive, has been input. KSRCH is printed.

Action taken: Execution is terminated.

Message: *****TWO-DIMENSIONAL SEARCHES NOT ALLOWED. TYPE=aaaaaaaa

ERROR NUMBER 3 IN SUBROUTINE SRCHIN

Significance: A Y-dimension search or an X- and Y-dimension search (KSRCH = 3 or 6) has been specified.

Action taken: Execution is terminated.

Subroutine UDBHMG (N1, V1, N2, V2, MC, MX, DM, DC, DIMIX, ATMDEN, MSPEC, DITBL, DMTBL, MCNEW, NB, IEOF)

1. Arguments

Name	Type	Dimension	Definition
N1	R*8	NCMP*NMAT	Work space for setting up new DMMIX table for update of B.HOMOG
V1	R*8	NCMP*NMAT	Work space for setting up new VOLFRC table for update of B.HOMOG

Name	Type	Dimension	Definition
N2	R*8	MCSLTH	Material mix table from B.HOMOG (DMMIX in BLK array)
V2	R*8	MCSLTH	Volume fraction table from B.HOMOG (VOLFRC in BLK array)
MC	I*4	NCMP	Number of materials in each composition, from B.HOMOG (MCSPEC in BLK array)
MX	I*4	CMAX	Number of modifier composition for each composition, from SP.CRIT (MODXS in BLK array)
DM	R*8	NMAT	List of all materials used from B.HOMOG (DMTBL in BLK array)
DC	R*8	NCMP	List of all compositions used, from B.HOMOG (DCTBL in BLK array)
DIMIX	R*8	MSLTH	Isotope mix table, from B.HOMOG
ATMDEN	R*8	MSLTH	Atomic density table, from B.HOMOG
MSPEC	I*4	NMAT	Number of isotopes in each material, from B.HOMOG
DITBL	R*8	NISO2	List of all isotopes used, from B.HOMOG
DMTBL	R*8	NMAT	List of all materials used, from B.HOMOG
MCNEW	R*8	MCMP	Work space for setting up new MCSPEC table after concentration search
NB	I*4	--	Data set reference number for B.HOMOG
IEOF	I*4	--	Flag for number of records in B.HOMOG: 0 - four records; 1 - two records

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(52)	XNTH	Search control parameter $X^{(n)}$

3. Local Variables

Name	Type	Definition
BLANK	R*8	A word containing eight blanks, used for wiping out duplicate names and for comparisons
MSLTH	I*4	Length of isotope mix tables
NMAT	I*4	Number of materials
NISO2	I*4	Number of isotopes used
MCSLTH	I*4	Length of material mix tables
NCMP	I*4	Number of compositions
L1	I*4	Location of first material assigned to composition C in N2 and V2 during packing; in N1 and V1 during updating
L2	I*4	Location of last material assigned to composition C in N2 and V2 during packing; in N1 and V1 during updating
L3	I*4	Number of position in N2 and V2 into which a given member of N2 and V2 is moved during packing; location of first material assigned to composition C in N2 and V2 during updating
LL1	I*4	L2-1 during packing; L1-L3, and location of first material assigned to composition MMX in N2 and V2, during updating

Name	Type	Definition
LL	I*4	Location in N2 of next name after a name found to be non-blank during packing; original number of materials assigned to composition C during updating
CM1	I*4	Number of materials assigned to composition C, MC(C), during packing; MMX-1 during updating
L4	I*4	Location of last material assigned to composition C in N2 and V2 during updating
LL3	I*4	Same as L4
L5	I*4	Location in N1 and V1 for element being moved from N2 and V2
MMX	I*4	Number of composition modifying composition C, MX(C), during search (if MMX=0, C is not modified)
KK	I*4	Number of materials assigned to modifier composition MMX
LL5	I*4	Location of last material assigned to modifier composition MMX in N2 and V2 during updating
NN2	R*8	Name of material assigned to modifier composition MMX which is being compared to names of materials assigned to C, from N2

4. Functions and tasks performed by this subroutine

This subroutine updates data set B.HOMOG after a concentration search. It first reads the four records of B.HOMOG (or two records if only two are present, indicating possible direct isotope-to-composition assignment).

UDBHMG then "packs" the existing material mix table N2 and volume fraction table V2, to eliminate duplicate material names. For each composition, the subroutine scans the material mix table. When a non-blank name is found, it scans the remaining names for duplicates. If any duplicate

names are found, it adds the volume fraction of the duplicate into the location of the first occurrence, and blanks the duplicate name. After doing this for all compositions, it packs the tables by eliminating all positions in N2 and V2 corresponding to blank names and updates MC, the number of materials per composition, to reflect the change.

The subroutine then proceeds to update the data into the corresponding working arrays N1, V1, and MCNEW. After setting up the appropriate indices for each composition C, it checks whether C was modified during the search. If not, the original material mix table and volume fraction table are copied into the appropriate locations in N1 and V1 as they are. If C is modified by some modifier composition MMX, the subroutine scans the table of materials present in MMX. For each, it scans the table of materials present in C. When a match is found, the volume fraction of the material in C is updated by adding it to the product of the search parameter XNTH and the volume fraction of the material in MMC. If no match is found, the material has not previously been present in C; its name is added to the updated mix table, with a corresponding volume fraction of XNTH times the volume fraction of the material in MMC, and the updated number of materials assigned to C, MCNDW(C), is incremented by one. After each composition has been updated the length of the material mix tables, MCSLTH, is also updated appropriately.

After all compositions have been updated in the working mix tables N1, V1, and MCNEW, the entire B.HOMOG data set is rewritten to include the changes. A message to this effect is printed, and the updated mix tables for each composition modified are also displayed.

5. Subprograms called by this subroutine

SQUEZE (ARC System module)
PAGHED

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine - None

Subroutine UPDTXS (XA, XTOT, REM, XTR, FISXS, NUFIS, CCHI, SCTXS, PXNF, DNCHI, MODXS, FSCMP, ICHI, NCMP, NGROUP, NFCMP, MAXSCT, ISCHI)

1. Arguments

Name	Type	Dimension	Definition
XA	R*8	NCMP*NGROUP	Absorption cross section
XTOT	R*8	NCMP*NGROUP	Total cross section
REM	R*8	NCMP*NGROUP	Removal cross section
XTR	R*8	NCMP*NGROUP	Transport cross section
FISXS	R*8	NFCMP*NGROUP	Fission cross section
NUFIS	R*8	NFCMP*NGROUP	Nu times fission cross section
CCHI	R*8	NFCMP*NGROUP* NGROUP	Fission spectrum $\chi^{g'g}$
SCTXS	R*8	NCMP*NGROUP* MAXSCT	Total scattering into given group
PXNF	R*8	NFCMP*NGROUP if NFAM > 0 0 if NFAM = 0	Number of prompt neutrons per fission times fission cross section
DNCHI	R*8	NFCMP*NGROUP* NFAM	Delayed neutron fission spectrum
MODXS	I*4	NCMP	Number of modifier composition for each composition
FSCMP	I*2	NCMP	Indices of fissionable compositions in arrays dimensioned NFCMP
ICHI	I*4	NCMP	Composition fission spectrum flag -1 - uses set-wide χ 0 - non-fissionable 1 - uses own χ-vector NGROUP - uses own χ-matrix
NCMP	I*4	--	Number of compositions defined

Name	Type	Dimension	Definition
NGROUP	I*4	--	Number of groups defined
NFCMP	I*4	--	Number of fissionable compositions defined
MAXSCT	I*4	--	Maximum scattering band width over all compositions
ISCHI	I*4	--	Set-wide fission spectrum 0 - no set-wide χ 1 - set-wide χ-vector NGROUP - set-wide χ-matrix

2. Common Blocks

/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4 words

Variables used from this common block

Name	Local Name	Definition
FLT(52)	XNTH	Final value of search parameter $X^{(n)}$
INT(44)	NFAM	Number of families defined

3. Local Variables

Name	Type	Definition
MODC	I*4	Number of modifier composition corresponding to composition C
NF	I*4	Fissionable composition number corresponding to composition C
ICHC	I*4	ICHI for composition C
MODF	I*4	Fissionable composition number corresponding to composition MODC

4. Functions and tasks performed by this subroutine

This subroutine updates all of the data in composition blocks of data set XS.C.MIN for modified compositions following a concentration search. For the absorption cross section, total cross section, transport cross section, and scattering cross section, and for the fission cross section and the nu times fission cross section, if present, the modification is simply

$$\left(XS_c^g\right)^{(n)} = \left(XS_c^g\right)^{(0)} + X^{(n)} \cdot XS_{mc}^g \quad ,$$

where XS_c^g represents the cross section for the composition c being modified and the group g, and XS_{mc}^g the corresponding value for the modifier composition.

For a fissionable composition, the fission spectrum is modified as follows for the case of a χ-matrix, computing $\chi^{g'g}$ over all g for a given g':

$$\left(\chi_c^{g'g}\right)^{(n)} = \sum_{g=1}^{G} \frac{\left(\chi^{g'g}\right)^{(0)}\left(\left(\nu\cdot\sigma_f\right)_c^{g'}\right)^{(0)} + X^{(n)}\left(\chi_{mc}^{g'g}\right)\left(\nu\cdot\sigma_f\right)_{mc}^{g'}}{\left(\left(\nu\cdot\sigma_f\right)_c^{g'}\right)^{(n)}} \quad .$$

For a χ-vector, the modification is similar:

$$\left(\chi_c^g\right)^{(n)} = \sum_{g=1}^{G} \frac{\chi_c^{g\,(0)}\left(\left(\nu\cdot\sigma_f\right)_c^{g'}\right)^{(0)} + X^{(n)}\left(\chi_{mc}^g\right)\left(\nu\cdot\sigma_f\right)_{mc}^{g'}}{\left(\left(\nu\cdot\sigma_f\right)_c^{g'}\right)^{(n)}} \quad .$$

If prompt and delayed neutron data are present, the number of prompt neutrons times fission cross section is modified simply as

$$\left(\left(\nu_p\cdot\sigma_f\right)_c^g\right)^{(n)} = \left(\left(\nu_p\cdot\sigma_f\right)_c^g\right)^{(0)} + X^{(n)}\left(\nu_p\cdot\sigma_f\right)_{mc}^g \quad .$$

The delayed neutron fission spectrum is modified analogously to the fission spectrum χ:

$$\left(\chi^{g}_{c,f}\right)^{(n)} = \frac{\left(\chi^{g}_{c,f}\right)^{(0)} \cdot \left(\left(\nu_p \cdot \sigma_f\right)^{g}_{c}\right)^{(0)} + \chi^{(n)}\left(\chi^{g}_{mc,f}\right) \cdot \left(\nu_p \cdot \sigma_f\right)^{g}_{mc}}{\left(\nu_p \cdot \sigma_f\right)^{g}_{c}} .$$

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

   WSRCH

7. Error messages generated by this subroutine - None

Subroutine WRCRD1 (CNAME, N3, N4, N5, N6, N7, N8, N9, N2)

1. Arguments

Name	Type	Dimension	Definition
CNAME	R*8	NCMP	Composition names
N3	I*4	--	Number of compositions defined, NCMP (first data item on record 1 of XS.C.MIN)
N4	I*4	--	Second through seventh data items on record 1 of data set XS.C.MIN
N5	I*4	--	
N6	I*4	--	
N7	I*4	--	
N8	I*4	--	
N9	I*4	--	
N2	I*4	--	Data set reference number for XS.C.MI1

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine writes record 1 of data set XS.C.MIN on data set XS.C.MI1 following a concentration search. The contents of this record are not modified during the search.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine - None

Subroutine WRCRD2 (CHI, ISCHI, NGROUP, N2)

1. Arguments

Name	Type	Dimension	Definition
CHI	R*8	ISCHI*NGROUP	Set-wide fission spectrum $\chi^{g'g}$
ISCHI	I*4	--	Set-wide fission spectrum flag 0 - no set-wide χ 1 - set-wide χ-vector NGROUP - set-wide χ-matrix
NGROUP	I*4	--	Number of groups defined
N2	I*4	--	Data set reference number for XS.C.MI1

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This subroutine rewrites record 2 of data set XS.C.MIN, the set-wide fission spectrum $\chi^{g'g}$, if present, on data set XS.C.MI1 following a concentration search. The set-wide fission spectrum is not modified during the search; if it was used by any composition modified, that composition will be written with an individual fission spectrum.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine - None

7. Error messages generated by this subroutine - None

Subroutine WRCRD3 (NUP, NDN, ICHI, NCMP, NGROUP, C, F, N2)

1. Arguments

Name	Type	Dimension	Definition
NUP	I*4	NCMP*NGROUP	Number of groups of upscattering for each composition and group
NDN	I*4	NCMP*NGROUP	Number of groups of downscattering for each composition and group
ICHI	I*4	NCMP	Composition fission spectrum flag -1 - uses set-wide χ 0 - non-fissionable 1 - uses own χ-vector NGROUP - uses own χ-matrix
NCMP	I*4	--	Number of compositions defined
NGROUP	I*4	--	Number of groups defined

Name	Type	Dimension	Definition
C	I*4	--	Composition number
F	I*4	--	Fissionable composition index for composition C
N2	I*4	--	Data set reference number for XS.C.MI1

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

For each composition C, this subroutine rewrites the type 3 record of data set XS.C.MIN on data set XS.C.MI1 following a concentration search. The only data item on this record which may have been modified after the search is the fission spectrum flag ICHI; a modified composition which originally used the set-wide fission spectrum (ICHI = -1) is now assigned its own fission spectrum (ICHI = 1 or ICHI = NGROUP).

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine - None

Subroutine WRCRD4 (XA, XTOT, REM, XTR, FISXS, NUFIS, CCHI, SCTXS, PXNF, DNCHI, NUP, NDN, ICHI, NCMP, NGROUP, NFCMP, C, G, F, N2)

1. Arguments

Name	Type	Dimension	Definition
XA	R*8	NCMP*NGROUP	Absorption cross section
XTOT	R*8	NCMP*NGROUP	Total cross section
REM	R*8	NCMP*NGROUP	Removal cross section
XTR	R*8	NCMP*NGROUP	Transport cross section
FISXS	R*8	NFCMP*NGROUP	Fission cross section
NUFIS	R*8	NFCMP*NGROUP	Nu times fission cross section
CCHI	R*8	NFCMP*NGROUP* NGROUP	Fission spectrum $\chi^{g'g}$
SCTXS	R*8	NCMP*NGROUP* MAXSCT	Total scattering into given group
PXNF	R*8	NFCMP*NGROUP if NFAM > 0 0 if NFAM = 0	Number of prompt neutrons per fission times fission cross section
DNCHI	R*8	NFCMP*NGROUP* NFAM	Delayed neutron fission spectrum
NUP	I*4	NCMP*NGROUP	Number of groups of upscattering for each composition and group
NDN	I*4	NCMP*NGROUP	Number of groups of downscattering for each composition and group
ICHI	I*4	NCMP	Composition fission spectrum -1 - uses set-wide 0 - non-fissionabl 1 - uses own χ-vector NGROUP - uses own χ-matrix
NCMP	I*4	--	Number of compositions defined
NGROUP	I*4	--	Number of groups defined

Name	Type	Dimension	Definition
NFCMP	I*4	--	Number of fissionable compositions defined
C	I*4	--	Composition number
G	I*4	--	Group number
F	I*4	--	Fissionable composition index for composition C
N2	I*4	--	Data set reference number for XS.C.MI1 (output composition data set)

2. Common Blocks

/SINGLE/FLT(100), INT(100) — Single variable container: FLT contains 100 R*8 words, and INT contains 100 I*4

Variables used from this common block

Name	Local Name	Definition
INT(44)	NFAM	Number of families defined

3. Local Variables

Name	Type	Definition
ICHC	I*4	ICHI for given composition
NSCAT	I*4	Maximum scattering band width for given composition

4. Functions and tasks performed by this subroutine

For a given group G and composition C, this subroutine writes the group record, record type 4 of data set XS.C.MIN, on data set XS.C.MI1 according to the proper format. Formats vary according to whether the composition is non-fissionable or fissionable, whether it uses the set-wide fission spectrum or is assigned its own, and whether prompt and delayed

neutron data is present. If composition C was modified during a concentration search, all data written out represents the modified values.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

WSRCH

7. Error messages generated by this subroutine - None

Subroutine WSRCH (IERR)

1. Arguments

Name	Type	Dimension	Definition	
IERR	I*4	--	Error flag returned	0 - no error in writing data set 1 - error in writing data set

2. Common Blocks

/BLKSTR/BLK(1)	POINTR dynamic storage container array
/SINGLE/FLT(100), INT(100)	Single variable container: FLT contains 100 R*8 words, and INT contains I*4 words

Variables used from this common block

Name	Local Name		Definition	
FLT(51)	SEARCH	Search type	'CONC' 'XDIM' 'YDIM'	'BSQ' 'ALPHA' 'XYDIM'

Name	Local Name	Definition
INT(17)	SCHOPT	Data set writing option 0 - update data set 1 - don't update data set
INT(19)	MAXBLK	Size of POINTR container array in bulk storage
INT(44)	NFAM	Number of families of delayed data

3. Local Variables

Name	Type	Definition
XSC	R*8	'XS.C.MIN' for SNIFF call
XSCSH	R*8	'XS.C.MI1' for SNIFF call
CONC	R*8	'CONC' for search type
ALF	R*8	'ALPHA' for search type
GGEOM	R*8	'GEOM' for SNIFF call
BSQ	R*8	'BSQ' for search type
YDIM	R*8	'YDIM' for search type
XDIM	R*8	'XDIM' for search type
N1	I*4	Data set reference number for XS.C.MIN (input composition data set)
N2	I*4	Data set reference number for XS.C.MI1 (output composition data set)
NCMP	I*4	Number of compositions defined
NGROUP	I*4	Number of groups defined
ISCHI	I*4	Fission spectrum flag 0 - no set-wide χ 1 - set-wide χ-vector NGROUP - set-wide χ-matrix
NFCMP	I*4	Number of fissionable compositions

Name	Type	Definition
MAXUP	I*4	Maximum number of groups of upscattering over all compositions
MAXDN	I*4	Maximum number of groups of downscattering over all compositions
MAXSCT	I*4	Maximum scattering band width over all compositions
F	I*4	Number of fissionable compositions after each composition block has been processed
LAST1	I*4	Last location used in POINTR core container array for update of XS.C.MIN or GEOM; also maximum of LAST1 and LAST2 for concentration search
LASTB1	I*4	Last location used in POINTR bulk container array for update of XS.C.MIN or GEOM; also maximum of LASTB1 and LASTB2 for concentration search
NB	I*4	Data set reference number for B.HOMOG
MSLTH	I*4	Length of material mix table in record type 2 of B.HOMOG
NMAT	I*4	Number of materials, from records 1 and 3 of B.HOMOG
NISO2	I*4	Number of isotopes in list in record type 2 of B.HOMOG
D	R*8	Dummy variable for skipping record type 2
IEOF	I*4	Flag for number of records in B.HOMOG: 0 - four records; 1 - two records
MCSLTH	I*4	Length of composition mix table in record type 4 or 2 of B.HOMOG
LAST2	I*4	Last location used in POINTR core container array for update of B.HOMOG
LASTB2	I*4	Last location used in POINTR bulk container array for update of B.HOMOG
N3	I*4	Data set reference number for GEOM

Name	Type	Definition
NDIM	I*4	Number of dimensions given in geometry
NGEOM	I*4	Geometry type number
NPTI	I*4	Number of I-direction mesh lines defined
NPTJ	I*4	Number of J-direction mesh lines defined
NINTI	I*4	Number of I-direction mesh intervals defined
NINTJ	I*4	Number of J-direction mesh intervals defined
NREG	I*4	Number of regions defined
NAREA	I*4	Number of areas defined
LAREA	I*4	Length of area record in GEOM
NZONE	I*4	Number of zones defined
LZONE	I*4	Length of zone record in GEOM
NFIN	I*4	Number of finite directions defined
NIL	I*4	Number of finite geometry type

4. Functions and tasks performed by this subroutine

This subroutine controls the modifying of the data sets whose original contents are altered during a diffusion search following completion of the search. If the search option flag SCHOPT (in data set A.DIF1D or A.DIF2D) is non-zero, or if the search type was alpha, no updating of data sets is performed. Otherwise, for a concentration search, data sets XS.C.MIN and B.HOMOG are updated; for a buckling or dimension search, GEOM is updated. In both cases, the data set as modified by the search replaces the old data set.

For a rewriting of XS.C.MIN, the entire old data set is first read in, using arrays set up for that purpose. The first record is read within this subroutine, and storage is set up for the remainder of the data set. The data set is rewound, and RCORD1 is then called to read the remainder of the

first record, the composition names. If a set-wide fission spectrum is present, RCORD2 is then called to read it from the second record.

For each composition, then, RCORD3 and RCORD4 are called to read the data on record types 3 and 4. Then UPDTXS is called to update this data using the results of the search; for each composition modified, each cross section is altered using the final value of the search parameter X and the appropriate modifier composition. The resulting data is then written out by calling WRCRD1, WRCRD2 (for a set-wide fission spectrum), and, for each composition, WRCRD3 and WRCRD4. The amount of POINTR container array space needed for the XS.C.MIN update is saved, unneeded arrays are wiped out of the container arrays, and the container arrays are packed.

The material-to-composition assignments and the volume fractions in data set B.HOMOG are also updated for a concentration search. The first record of B.HOMOG is read, and storage arrays are set up for the contents of the second record. An attempt is then made to read the third record. If this attempt is successful, storage is set up for the contents of the fourth record. If an end-of-file is encountered, B.HOMOG is rewound and the first record is re-read in place of the third record. Storage for the contents of the second record are set up as storage for the fourth record above. WSRCH then calls UDBHMG to perform the actual updating of the data set. Following this, the amount of POINTR container array space needed for the B.HOMOG update is determined, and the maximum used for both updates is printed.

If data set GEOM is to be updated, the first record is similarly read within this subroutine, and storage for the remainder of the original data is set up. The amount of POINTR container array space needed for the GEOM update is determined and printed. RDGEOM is then called to read the rest of the data set. Record 1 is then rewritten. Following this, for a buckling search, entry WTBSQ of RDGEOM is called, and the updated buckling data from the search replaces the old buckling data. For a dimension search, entry WTDIM is similarly called, replacing the appropriate original arrays with the corresponding results. In either case, the entry writes the remainder of the data set.

5. Subprograms called by this subroutine

SNIFF (ARC System Module)
ERRMSG
PUTPNT (entry in POINTR)
REDEF (entry in POINTR)
IGET (entry in POINTR)
RCORD1
RCORD2
RCORD3
RCORD4
UPDTXS
ILAST (entry in POINTR)
ILASTB (entry in POINTR)
WRCRD1
WRCRD2
WRCRD3
WRCRD4
PURGE (entry in POINTR)
PURGEB (entry in POINTR)
UDBHMG
RDGEOM
WTBSQ (entry in RDGEOM)
WTDIM (entry in RDGEOM)

6. Subprograms calling this subroutine

D1DSCH

7. Error messages generated by this subroutine

Message: *****DATA SET XS.C.MIN COULD NOT BE UPDATED
DSRN FOR OLD=iiii DSRN FOR NEW=iiii

ERROR NUMBER 1 IN SUBROUTINE WSRCH

Significance: One of the two data sets could not be found by ARC System routine SNIFF. This implies either that the old one was not written or that the new one was not set up. In practice, currently, both data sets have the same data set reference number, since the old is written over the new.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: *****ERROR IN READING RECORD 1 OF XS.C.MIN ON DSRN iii

ERROR NUMBER 2 IN SUBROUTINE WSRCH

Significance: An error was encountered in attempting to read the first record of this data set.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: ERROR NUMBER 3 IN SUBROUTINE WSRCH

Significance: An error was encountered in attempting to read some record of data set XS.C.MIN.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: *****DATA SET GEOM NOT FOUND

ERROR NUMBER 4 IN SUBROUTINE WSRCH

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: ERROR NUMBER 5 IN SUBROUTINE WSRCH

Significance: An end-of-file was encountered after the second record of B.HOMOG. The data set was re-wound, and successive attempts to re-read record 1 resulted in an end-of-file being read. (This is highly unlikely.)

Action taken: The IERR flag is set to 1 and execution is terminated.

5. D1DSCH (NUC004) Common Block Descriptions

This section provides more detailed information about the /SINGLE/ common block as it is used in the D1DSCH module. All other common blocks, except the /BLKSTR/ common block described in section 6, are the same as for the DIF1D module.

For the /SINGLE/ block, all members used are listed, together with their local names, definitions, and lists of the subprograms in which the value of the member is set or altered.

/SINGLE/FLT(100),INT(100)

Name	Local Name	Subprograms where values are set or modified	Definition	
FLT(1)	PROB	RCDEP	Problem type, determined from SP.CICN data	'REAL' 'ADJOINT' 'BOTH'
		D1DSCH	Reset to	'REAL' 'ADJOINT'
FLT(2)	KEFF	RFLUX ENDOUT (entry in INDOUT)	k_{eff} in k-effective calculations, read from data set FR.D1 or FA.D1 and modified during iterations	
FLT(3)	ALPHA	RCDEP	Inverse reactor time period α, from data set SP.CICN	
		SRCHIN	Desired α for search to a prescribed α, from data set SP.CRIT	
		DELALF	Modified during α-search to a prescribed k_{eff}	
FLT(4)	SCTEPS	RCDEP	Convergence criterion ε_S for upscatter iterations in k-effective calculation, from data set A.DIF1D $\max_G \dfrac{GMAX + \sum_{I=1}^{IMAX} \lvert \phi_{I,G}^{(\ell)} - \phi_{I,G}^{(\ell-1)} \rvert}{\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(\ell-1)}} \leq \varepsilon_S$	
FLT(5)	PROBT	RCDEP	Calculation type, 'K CALC', determined from SP.CICN data	
FLT(6)	MU	RCDEP	1. /k_{fixed} for source calculations, where k_{fixed} is read from data set SP.CICN	
FLT(11)	EPOFLX	RCDEP	Convergence criterion ε_f on sum of absolute value of flux differences, from data set A.DIF1D $\dfrac{\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \lvert \phi_{I,G}^{(n)} - \phi_{I,G}^{(n-1)} \rvert}{\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(n)}} \leq \varepsilon_f$ for k-effective calculation	

/SINGLE/FLT(100),INT(100)(Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
FLT(12)	EPSLAM	RCDEP	Convergence criterion ε_λ on difference of flux bounds, from data set A.DIF1D $(\bar{\lambda}^{(n)} - \underline{\lambda}^{(n)}) \leq \varepsilon_\lambda$, where $\bar{\lambda}^{(n)} = \max_{I,G} \left(\frac{\phi_{I,G}^{(n)}}{\phi_{I,G}^{(n-1)}} \right)$ and $\underline{\lambda}^{(n)} = \min_{I,G} \left(\frac{\phi_{I,G}^{(n)}}{\phi_{I,G}^{(n-1)}} \right)$
FLT(13)	EPSKEF	RCDEP	Convergence criterion ε_k on difference in k_{eff} from data set A.DIF1D $(k^{(n)} - k^{(n-1)}) \leq \varepsilon_k$
FLT(14)	AOUTER	RCDEP	Interval shift factor a when applying Chebyshev extrapolation, from data set A.DIF1D
FLT(15)	FLXMON	RCDEP	Flux monitoring ratio, used in computing FLXNGL (FLT(73)), from data set A.DIF1D
FLT(31)	GEOM	RGEOM	Geometry type 'CYLSLAB' 'RECTSLAB' 'CYLINDER' 'SPHERE'
FLT(32)	XL	RMESH	Left reactor boundary, X(1), from data set GEOM
FLT(33)	XR	RMESH	Right reactor boundary, X(IMAX+1), from data set GEOM
FLT(51)	SEARCH	SRCHIN	Search type, set up from SP.CRIT data 'CONC' 'XDIM' 'YDIM' 'BSQ' 'ALPHA' 'XYDIM'

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
FLT(52)	XNTH	D1DSCH CHANGE	Current value of search control parameter, $X^{(n)}$ (initially read from data set SP.CRIT)
FLT(53)	XNM1	D1DSCH CHANGE	Previous value of search control parameter, $X^{(n-1)}$ (initially read data set SP.CRIT)
FLT(54)	XNM2	D1DSCH CHANGE	Previous value of search control parameter, $X^{(n-2)}$ (initially read from data set SP.CRIT)
FLT(55)	DKDX	D1DSCH CHANGE	Slope of the curve k_{eff} vs. X at the point $X^{(n)}$ for search (initially from data set SP.CRIT)
FLT(56)	XL	D1DSCH	Lower bound on search control parameter X, from data set SP.CRIT
FLT(57)	XU	D1DSCH	Upper bound on search control parameter X, from data set SP.CRIT
FLT(58)	KZERO	D1DSCH	Desired k_{eff}, $k_{eff}^{(0)}$, for search to a prescribed k_{eff}, from data set SP.CRIT
FLT(59)	EPSSRC	D1DSCH	Convergence criterion ε_{SCH} on k_{eff} for search $\lvert k^{(0)} - k^{(n)} \rvert \leq \varepsilon_{SCH}$
FLT(60)	SKNTH	CHANGE	Current value of k_{eff}, $k_{eff}^{(n)}$, during search
FLT(61)	SKNM1	CHANGE	Previous value of k_{eff}, $k_{eff}^{(n-1)}$, during search
FLT(62)	SKNM2	CHANGE	Previous value of k_{eff}, $k_{eff}^{(n-2)}$, during search
FLT(64)	DALPHA	SRCHIN	Modifier $\delta\alpha$ for α search, from data set SP.CRIT
FLT(65)	XBAR	SRCHIN CHANGE	Value of $\bar{X}^{(n)}$ used in previous search pass relation $P^{(n)} = P^{(n-1)} + \bar{X}^{(n)} \delta P$ (initially, XBAR=XNTH)

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
FLT966)	DMOD	D1DSCH	Modifier for search when only one is necessary, from data set SP.CRIT
FLT(71)	POWIN	RCDEP	Power normalization factor P_0 for BURNUP calculation, used in writing data set FR.PN, from data set A.DIF1D. No power normalization performed if POWIN=0.
FLT(72)	FLXCVG	RFLUX ENDOUT (entry in INDOUT)	Sum of fluxes used in checking for convergence $\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(n)}$ for k-effective calculation $\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} \phi_{I,G}^{(\ell-1)}$ for k-effective calculation upscatter iteration
FLT(73)	FLXNGL	RFLUX ENDOUT (entry in INDOUT)	Factor below which fluxes are neglected in computation of flux bounds $\bar{\lambda}^{(n)}$ and $\underline{\lambda}^{(n)}$ $FLXNGL=FLXMON*\sqrt{\frac{\sum_{G=1}^{GMAX} \sum_{I=1}^{IMAX} [\phi_{I,G}^{(n-1)}]^2}{GMAX*IMAX}}$
INT(1)	MKRTHY	RCDEP	Problem type, from data set SP.CICN 0 - real problem 1 - adjoint problem 2 - both real and adjoint problems
INT(2)	KINOPT	RCDEP	Kinetics option, from data set A.DIF1D 0 - no kinetics 2 - delayed data for alpha calculation
INT(3)	LMAX	RCDEP	Maximum number of upscatter iterations, from data set A.DIF1D
INT(4)	MAXSIZ	RCDEP	Maximum size of POINTR container array in core, from data set A.DIF1D

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition	
INT(5)	IPRINT	RCDEP	POINTR debugging flag, from data set A.DIF1D	0 - no print out 1 - standard print out 2 - debugging dump print out 3 - full print out
INT(7)	KOUT	RCDEP	Diffusion output flag, from data set A.DIF1D	0 - problem description and iteration history print out 1 - above plus geometry and composition input data print out 2 - above plus standard balance output 3 - above plus power calculation
INT(9)	FSOPT	RCDEP	Fission source flag, from data set A.DIF1D	0 - no fission source by intervals output 1 - fission source by intervals printed and written on FSR.D1 or FSA.D1
INT(10)	XTROUT	RCDEP	Extrapolation type, from data set A.DIF1D	0 - no extrapolation 1 - Chebyshev extrapolation

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(11)	NMAX	RCDEP D1DSCH	Maximum number of outer iterations, from data set A.DIF1D Changed to NMAX+10 if adjoint problem is to be done following search
INT(12)	NSTART	RCDEP	Number of unextrapolated iterations to be done before beginning to extrapolate from data set A.DIF1D
INT(13)	NTEST	RCDEP	Number of outer iterations to be done in extrapolation sequence before testing for payoff, from data set A.DIF1D
INT(14)	NEXT	RCDEP	Maximum number of extrapolated iterations to be done in an extrapolation sequence, from data set A.DIF1D
INT(15)	NUNEXT	RCDEP	Number of unextrapolated iterations to be done following one extrapolation sequence before beginning another, from data set A.DIF1D
INT(16)	P	OUTERS (entry in INDOUT)	Extrapolation iteration counter (degree of Chebyshev polynomial)
INT(17)	SCHOPT	RCDEP	Search data set output flag, from data set A.DIF1D 0 - XS.C.MIN and B.HOMOG or GEOM written after search 1 - XS.C.MIN and B.HOMOG or GEOM not written after search
INT(19)	MAXBLK	RCDEP	Maximum size of POINTR container array in bulk (LCS) storage, from data set A.DIF1D
INT(31)	RMAX	RGEOM	Number of regions defined in problem, from data set GEOM

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(32)	IMAX	RGEOM	Number of mesh intervals defined in problem, from data set GEOM
INT(34)	BCMAX NBCDEF	RBNDC	Number of boundary conditions and non-continuity interface conditions defined in problem, from data set BC
INT(35)	CMAX	RGEOM RCOMP	Number of compositions defined in problem, read from data set GEOM and then from data set XS.C.MIN
INT(36)	FSCMAX	RCOMP	Number of fissionable compositions defined in problem, from data set XS.C.MIN
INT(37)	GMAX	RCOMP	Number of energy groups defined in problem, from data set XS.C.MIN
INT(38)	GMAXUP	RCOMP RGROUP	Number of lowest-numbered group from which upscattering occurs in problem (GMAXUP=GMAX if there is no upscattering)
INT(39)	MTRCHI	RCOMP RGROUP	Fission spectrum array type 0 - χ-vector 1 - χ-matrix
INT(40)	CHIDIM	RCOMP	First dimension of fission spectrum array CHI, CHI(CHIDIM,GMAX) CHIDIM=FSCMAX*GMAX**MTRCHI
INT(41)	SCTDIM	RCOMP DIMSCT	First dimension of packed scattering cross section array SCTXS, SCTXS(SCTDIM,GMAX) SCTDIM is the sum over compositions of the maximum possible band width for each composition $\text{SCTDIM} = \sum_{C=1}^{CMAX} \text{SCTLIM}(1,C)+\text{SCTLIM}(2,C)+1$
INT(44)	NFAM	RCOMP WSRCH	Number of families of delayed data, from data set XS.C.MIN Re-read for update of XS.C.MIN following search

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(49)	NGEOM	RGEOM	Number of geometry type, read from data set GEOM as three values and modified to four 1 - cylindrical slab 2 - rectangular slab 3 - cylinder 4 - sphere
INT(50)	NBLACK	RCDEP	Number of black composition group intervals defined, from data set A.DIF1D
INT(51)	NTH LTH	RCDEP OUTERS SWEEP (entry in ISWEEP)	Position in flux array of nth generation of fluxes; $\phi_{i,g}^{(n)}$ is in FLX(I,G,NTH) Also position of ℓth generation of fluxes in an upscattering problem
INT(52)	NM1	RCDEP OUTERS	Position in flux array of (n-1)th generation of fluxes; $\phi_{i,g}^{(n-1)}$ is in FLX(I,G,NM1)
INT(53)	NM2	RCDEP OUTERS	Position in flux array of (n-2)th generation of fluxes; $\phi_{i,g}^{(n-2)}$ is in FLX(I,G,NM2)
INT(54)	NDUM LM1	RCDEP SWEEP (entry in ISWEEP)	Position in flux array of (ℓ-1)th generation of fluxes for upscattering problem; $\phi_{i,g}^{(\ell-1)}$ is in FLX(I,G,LM1)
INT(57)	ITER	OUTERS	Iteration number at which flux data set FR.D1 or FA.D1 is written out
INT(58)	NRST	RCDEP	First iteration number at which FR.D1 or FA.D1 will be written out for restart option, from data set A.DIF1D
		D1DSCH	Changed to NRST+10 if adjoint problem is to be done following search and if NRST was originally set to zero (option not desired)

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition
INT(59)	NRSTI	RCDEP	Increment in NRST at which succeeding intervals FR.D1 or FA.D1 will be written again for restart option, from data set A.DIF1D
INT(81)	NPMAX	D1DSCH	Maximum number of control passes during search
INT(82)	INTERP	D1DSCH	Interpolation type used during search, from data set SP.CRIT 1 - linear interpolation 0 or 2 - parabolic interpolation
INT(83)	NPASS	D1DSCH	Control pass number during search, read initially from data set SP.CRIT as zero
INT(84)	KSRCH	D1DSCH	Search type, from data set SP.CRIT 1 - concentration 2 - X-dimension 3 - Y-dimension 4 - buckling 5 - α 6 - X- and Y-dimension
INT(88)	NFIN	RGEOM	Number of finite geometry type, from data set GEOM
INT(89)	NIL	RGEOM	Number of finite directions defined per region, from data set GEOM
INT(93)	IPROB	RCDEP	Calculation type, from data set SP.CICN 1 - source calculation 2 - k_{eff} calculation 3 - concentration search 4 - buckling search 5 - alpha search 6 - dimension search

/SINGLE/FLT(100),INT(100) (Contd.)

Name	Local Name	Subprograms where values are set or modified	Definition	
INT(94)	KFLAG	D1DSCH	Type of variable prescribed to search to, from data set SP.CRIT	0 - k_{eff} 1 - α

6. D1DSCH (NUC004) Dynamic Storage Array Allocation

This section describes all arrays which may be defined by this module in common block /BLKSTR/BLK, using the facilities of the dynamic storage allocation module POINTR (see Volume I). The arrays are listed in alphabetical order by the name given them in the POINTR name table. For each array, the following information is given:

1. type
2. dimension, in terms of module variables
3. names of subprograms where action is taken that affects the status of the array in the container, followed in parentheses by the action that is taken in the subprogram (Here, significant POINTR calls are generally PUTPNT, which allocates space in the container for the array; REDEF, which sets aside space if the array is not present or redefines its space allocation if it is; and WIPOUT, which removes the array from the name table, thus making it inaccessible. The meaning of any other POINTR calls listed can be found in the POINTR documentation.)
4. names of subprograms in which the value of elements of the array are set or altered
5. the definition of the array contents.

If the meaning of a variable used as the dimension of an array is not known, it can be found in the subprogram in which a call to PUTPNT or REDEF is made for that array.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
A	R*8	IMAX*GMAX	FDCGEN(REDEF) D1DOUT(PUTPNT)	FDC1D	Finite difference coefficients $a_{i,g}$ across the interface of cell i with cell i+1 for group g
AREA	R*8	NAREA	WSRCH(PUTPNT)	RDGEOM	Names of areas, from record type 9 of data set GEOM, for update of GEOM after search
ATMDEN	R*8	MSLTH	WSRCH(PUTPNT)	UDBHMG	Atomic density table, from record type 2 of data set B.HOMOG
B	R*8	NREG*NIL	WSRCH(PUTPNT)	RDGEOM	Original buckling data by region and direction, from record type 8 of data set GEOM, for update of GEOM after search
BLKCMP	R*8	NBLACK	RCDEP(PUTPNT) RCOMP(WIPOUT)	RBLACK	Names of compositions defined as black compositions in data set A.DIF1D, if any
BLKGRP	I*4	NBLACK*2	RCDEP(PUTPNT) RCOMP(WIPOUT)	RBLACK RGROUP	Numbers of initial and final energy groups between which a composition named in BLKCMP is defined as being black in data set A.DIF1D. For a composition BLKCMP(N), BLKGRP(N,1) is the initial energy group, and BLKGRP(N,2) is the final energy group. If BLKGRP(N,1)=0, the condition is applied over all energy groups in the composition. If BLKGRP(N,2)=0 and BLKGRP(N,1)≠0, the condition applies only to the energy group indicated by BLKGRP(N,1).

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
BLNC	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	BALCAL	Output balance by regions and groups. BLNC(R,G)=-LEAK(R,G) -RMVL(R,G)+SCAT(R,G)+FISS(R,G)/KEFF for a k-effective calculation.
BLNCG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Output balance by groups $BLNCG(G) = \sum_{R=1}^{RMAX} BLNC(R,G)$
BLNCR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Output balance by regions $BLNCR(R) = \sum_{G=1}^{GMAX} BLNC(R,G)$
{BNDC, JBNDC}	{R*4, I*4}	8*NBCDEF	RBNDC(PUTPNT)	STORBC	For each boundary condition NBC (NBC=1,...,NBCDEF) defined, eight terms as follows: JBNDC(1,NBC) - mesh line numbers, positive if the condition applies to the right side of the line, negative if it applies to the left. JBNDC(2,NBC) - boundary condition type 1 - $C_1\phi' + C_2\phi = C_3$ 2 - $C_1D\phi' + C_2\phi = C_3D$ where D is the diffusion coefficient 3 - $\phi_{i+1} = C_1\phi_i + C_2\phi_i'$ $\phi_{i+1} = C_3\phi_i + C_4\phi_i'$ JBNDC(3,NBC) - number of initial (higher) energy group to which condition applies

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
					BNDC(5,NBC) = C_1 BNDC(6,NBC) = C_2 BNDC(7,NBC) = C_3 BNDC(8,NBC) = C_4
BSQ	R*8	RMAX*2	RGEOM(PUTPNT)	RMESH	Buckling values from data set GEOM, where BSQ(R,1) is B^2 in the transverse direction and BSQ(R,2) is B^2 in the axial direction for region R. If B^2 is not defined for a given region R in a given direction N, BSQ(R,N)=0.
				DELBSQ	Modified during buckling search.
C	R*8	IMAX*GMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	Finite difference coefficients $c_{i,g}$ across the interface of cell i with cell i-1 for group g
CCHI	R*8	NFCMP*NGROUP *NGROUP	WSRCH(PUTPNT) WSRCH(WIPOUT)	RCORD4 UPDTXS	Fission spectrum by fissionable composition and group, from data set XS.C.MIN, for update of XS.C.MIN after search
CHI	R*8	CHIDIM*GMAX	RCOMP(PUTPNT)	RGROUP DELXS	Fission spectrum fractions stored as vectors (χ_c^g) or matrices ($\chi_c^{g'g}$) for each fissionable composition.
		0 for no set-wide χ ISCHI*GMAX for set-wide χ	WSRCH(PUTPNT) WSRCH(REDEF) WSRCH(WIPOUT)	RCORD2	Modified during concentration search. Set-wide fission spectrum $\chi^{g'g}$ if one is present in data set XS.C.MIN, for update of XS.C.MIN after search

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
CMPMSH	I*2	IMAX	RGEOM(PUTPNT)	RMESH	Composition-to-mesh correspondence, defined in data set GEOM. For any mesh interval (I,CMPMSH(I) is the number of the composition defined in I.
CMPNAM	R*8	CMAX	RGEOM(PUTPNT)	RMESH	Composition names, defined in data set GEOM. CMPNAM(C) is the name of the composition whose number is C.
CMPREG	I*2	RMAX	RGEOM(PUTPNT)	RMESH	Composition-to-region correspondence, defined in data set GEOM. For any region R, CMPREG(R) is the number of the composition defined in R.
CNAME	R*8	NCMP	WSRCH(PUTPNT) WSRCH(WIPOUT)	RDGEOM	Composition names, from record type 9 of data set GEOM, for update of GEOM following search
CRDUM	I*4	RMAX	RGEOM(PUTPNT) RGEOM(WIPOUT)	RMESH	Temporary storage for composition-to-region correspondence array as read from data set GEOM (stored in I*2 array CMPREG)
CURR	R*8	(RMAX+1) *GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	LKCAL	Output currents at region interfaces by group, in the direction of the positive x-axis

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
DC	R*8	CMAX*GMAX	RCOMP(PUTPNT)	RGROUP	Diffusion coefficients by composition and group. DC(C,G)=0. for a black composition and group. Diffusion coefficients are formed from the transport cross section read into this array from XS.C.MIN.
				DELXS	Modified during concentration search
DBSQ	R*8	RMAX*NIL	SRCHIN(PUTPNT)	RDBSQ	BSQ modifier values by region and direction for buckling search, from data set SP.CRIT
DCCHI	R*8	NFAM*GMAX for kinetics option 1 or 2 0 for kinetics option 0	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Delayed neutron-emitter-yield spectrum $\chi^{g'i}$ by family and group, from data set XS.DELAY
DCTBL	R*8	NCMP	WSRCH(PUTPNT)	UDBHMG	Labels of compositions, from record type 4 of data set B.HOMOG
DDX	R*8	IMAX	SRCHIN(PUTPNT)	RDDDX	DX modifier values $\delta\Delta X$ by interval for X-dimension search, from data set SP.CRIT

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
DECCON	R*8	NFAM for kinetics option 1 or 2 0 for kinetics option 0	RCOMP(PUTPNT) RCOMP(WTPOUT) RDIST(REDEF)	RDELAY	Decay constants by family for delayed neutrons, from data set XS.DELAY
DICHI	R*8	NFAM*GMAX for kinetics option 2 0 for kinetics option 0 or 1	RCOMP(PUTPNT) RCOMP(WTPOUT) RDIST(REDEF)	RDELAY	Delayed neutron spectrum χ_i^g by family and group, from data set XS.DELAY
DIMIX	R*8	MSLTH	WSRCH(PUTPNT)	UDBHMG	Isotope mix table, from record type 2 of data set B.HOMOG
DITBL	R*8	NISO2	WSRCH(PUTPNT)	UDBHMG	List of all isotopes used, from record type 2 of data set B.HOMOG
DMMIX	R*8	MCSLTH	WSRCH(PUTPNT)	UDBHMG	Material mix table, from record type 4 of data set B.HOMOG
DMTBL	R*8	NMAT	WSRCH(PUTPNT)	UDBHMG	List of all materials used, from record type 4 of data set B.HOMOG
DNCHI	R*8	NFCMP*NGROUP *NFAM	WSRCH(PUTPNT)	RCORD4 UPDTXS	Delayed neutron fission spectrum from data set XS.C.MIN, for update of XS.C.MIN after search

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
DNSCT	I*4	GMAX	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Number of groups of downscattering given for each group in a given composition block in data set XS.C.MIN
DUMNAM	R*8	NCMP*NMAT	WSRCH(PUTPNT)	UDBHMG	Work space for setting up new DMMIX table after concentration search
DUMVOL	R*8	NCMP*NMAT	WSRCH(PUTPNT)	UDBHMG	Work space for setting up new VOLFRC table after concentration search
DX	R*8	IMAX	RGEOM(PUTPNT)	RMESH DELX	Mesh interval lengths ΔX DX(I)=X(I+1)-X(I) Modified during X-dimension search
E	R*8	IMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	Central finite difference coefficient for cell i, including removal $D \cdot B^2$ terms, and boundary condition terms, if present
EX	R*8	NREG*NIL	WSRCH(PUTPNT)	RDGEOM	Original extrapolated half-height H_r by region and direction, from record type 8 of data set GEOM, for update of GEOM after search
EXTBC	R*8	0 for k-effective calculation	FDCGEN(REDEF)	FDC1D	Source due to inhomogeneous interface or boundary conditions for each mesh interval and group

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
EXTDIS	R*8	RMAX*2	RGEOM(PUTPNT)	RMESH	Extrapolation distance $[H_r-(H_L)_r]$ by region and direction from data set GEOM, where the extrapolated half-height H_r is first read into the array, and the actual half-height is then subtracted from it. EXTDIS(R,1) is the extrapolated half-height in the transverse direction for region R and EXTDIS(R,2) the extrapolated half-height in the axial direction. If buckling is not defined for a given region R in a given direction N, EXTDIS(R,N)=0.
				WTBSQ	Modified extrapolated half-height for update of data set GEOM after buckling search
EXTS	R*8	0 for k-effective calculation	D1DOUT(PUTPNT) WIPE(WIPOUT)	EXTCAL	External source by region and group for output edit, including external source due to inhomogeneous interface condition and to delayed data, if present
EXTSG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	External source by groups for output edit EXTSG(G)=0 for any G for a k-effective calculation
EXTSR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	External source by regions for output edit EXTSR(R)=0 for any R for a k-effective calculation

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
FISAVG	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FISREG (entry in FISCAL)	Region-averaged fission source integrals by group, for output edit
FISDEN	R*8	IMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FISCAL	Fission source density by mesh interval and group, for output edit
FISS	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FISREG (entry in FISCAL)	Fission source integrals by region and group for output edit
FISSG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Fission source integrals by groups $FISSG(G) = \sum_{R=1}^{RMAX} FISS(R,G)$
FISSOR	R*8	IMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FISCAL	Fission source integrals by mesh interval and group, for data set FSR.D1 or FSA.D1, printed and written only if FSOPT (INT(9)) $\neq$ 0
FISSR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Fission source integrals by regions $FISSR(R) = \sum_{G=1}^{GMAX} FISS(R,G)$
FISSRC	R*8	IMAX*GMAX	KEFCAL(REDEF) D1DOUT(WIPOUT)	FSRC (entry in IFSRC)	Fission source by mesh interval and group during iterations

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
FISXS	R*8	FSCMAX*GMAX	RCOMP(PUTPNT)	RGROUP	Fission cross section σ_f by composition and group from XS.C.MIN. Data is stored only for fissionable compositions.
				DELXS	Modified during concentration search
				RCORD4	Fission cross section σ_f for update
				UPDTXS	of XS.C.MIN after search
FIXSRC	R*8	0 for k-effective calculation	RDIST(PUTPNT) RDIST(REDEF)	RSORC	External fixed source by interval and group from data set ES.D1D, weighted using fission spectrum weighting over groups, if desired
FLX	R*8	IMAX*GMAX*4 for problem with upscattering IMAX*GMAX*3 otherwise	RDIST(PUTPNT)	RFLX	Three generations of fluxes, $\phi_{i,g}^{(n)}$, $\phi_{i,g}^{(n-1)}$, and $\phi_{i,g}^{(n-2)}$, during iterations, referenced by NTH (INT51)), NM1 (INT(52)), and NM2 (INT(53)). One additional generation, $\phi_{i,g}^{(\ell-1)}$, for upscatter iterations, referenced by LM1(INT(54)).
				NRMCAL FLIP	Fluxes (NTH generation only) for output edits, normalized for k-effective calculation, not normalized for source calculation, and written on data set FR.D1 or FA.D1
FLXVOL	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FVCAL	Regional flux integrals by group for output edit, normalized for k-effective calculations, not normalized for source calculations

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
FSCMP	I*2	CMAX	RCOMP(PUTPNT)	RGROUP	Indices corresponding to fissionable compositions. For any composition C, if FSCMP(C)=0, C is not a fissionable composition; if FSCMP(C) is some non-zero integer, it is the index of that composition in arrays dimensioned with FSCMAX.
GAMMA	R*8	IMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	Off-diagonal elements γ for a given group of upper matrix U which results from splitting tri-diagonal matrix A, whose elements are the finite difference coefficients
H	R*8	NREG*NIL	WSRCH(PUTPNT)	RDGEOM	Original actual half-height $(H_L)_r$ by region and direction, from record type 8 of data set GEOM, for update of GEOM after search
ICHI	I*4	NCMP	WSRCH(PUTPNT) WSRCH(WIPOUT)	RCORD3 UPDTXS	Flag from data set XS.C.MIN for determining fission spectrum type for each composition, for update of XS.C.MIN after search 0 - composition not fissionable -1- fissionable composition using set wide fission spectrum χ 1 - fissionable composition with individual χ-vector GMAX - fissionable composition with individual χ-matrix
LEAK	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	LKCAL	Leakage by region and group for output edits, including $D \cdot B^2$ terms if present

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
LEAKG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Leakage by groups $LEAKG(G) = \sum_{R=1}^{RMAX} LEAK(R,G)$
LEAKR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Leakage by regions $LEAKR(R) = \sum_{G=1}^{GMAX} LEAK(R,G)$
MC	I*4	NINTI*NINTJ	WSRCH(PUTPNT)	RDGEOM	Composition-to-mesh correspondence, from record type 4 of data set GEOM, for update of GEOM after search
MCNEW	I*4	NCMP	WSRCH(PUTPNT)	UDBHMG	Work space for setting up new MCSPEC table after concentration search
MCSPEC	I*4	NCMP	WSRCH(PUTPNT)	UDBHMG	Number of materials contained in each composition, from record type 4 of data set B.HOMOG
MODXS	I*4	CMAX	SRCHIN(PUTPNT)	RDCONC	Modifier composition numbers, from data set SP.CRIT, for concentration search. For any composition C, if MODXS(C)=0, C is not modified in this series; if MODXS(C)≠0, it is the number of the modifying composition.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
MR	I*4	NINTI*NINTJ	WSRCH(PUTPNT)	RDGEOM	Region-to-mesh correspondence, from record type 3 of data set GEOM, for update of GEOM after search
MSHCMP	I*4	IMAX	RGEOM(PUTPNT) RGEOM(WIPOUT)	RMESH	Temporary storage for composition-to-mesh correspondence as read from data set GEOM (stored in I*2 array CMPMSH)
MSHREG	I*4	IMAX	RGEOM(PUTPNT) RGEOM(WIPOUT)	RMESH	Temporary storage for region-to-mesh correspondence as read from data set GEOM (stored in I*2 array REGMSH)
MSPEC	I*4	NMAT			Number of isotopes contained in each material, from record type 2 of data set B.HOMOG
NC	I*4	MREG	WSRCH(PUTPNT)	RDGEOM	Composition-to-region correspondence, from record type 5 of data set GEOM, for update of GEOM after search
NDN	I*4	NCMP*NGROUP	WSRCH(PUTPNT) WSRCH(WIPOUT)	RCORD3	Number of groups of downscattering given for each group in a given composition block in data set XS.C.MIN, for update of XS.C.MIN after search
NRA	I*4	LAREA	WSRCH(PUTPNT)	RDGEOM	Complete contents of record type 6 of data set GEOM, if present, for update of GEOM after search

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
NSB	I*4	LZONE	WSRCH(PUTPNT)	RDGEOM	Complete contents of record type 7 of data set GEOM, if present, for update of GEOM after search
NUFIS	R*8	FSCMAX*GMAX	RCOMP(PUTPNT)	RGROUP	Nu times fission cross section $\nu \cdot \sigma_f$ by composition and group, where ν is the number of neutrons released per fission for that group, from data set XS.C.MIN.
				DELXS	Modified during concentration search
				RCORD4 UPDTXS	Nu times fission cross section $\nu \cdot \sigma_f$ for update of XS.C.MIN after search
NUP	I*4	NCMP*NGROUP	WSRCH(PUTPNT) WSRCH(WIPCUT)	RCORD3	Number of groups of upscattering given for each group in a given composition block in XS.C.MIN, for update of XS.C.MIN after search
OMEGA	R*8	IMAX	SRCCAL(PUTPNT) KEFCAL(REDEF) D1DOUT(WIPOUT)	INVERT	The solution, $L\omega = F$, of the down sweep in the Choleski inversion of the tri-diagonal matrix $A = LU$, used in the up sweep $Ux = \omega$. This is the method used to solve a system of equations $Ax = F$.
PRECON	R*8	0 for k-effective calculation	RDIST(REDEF) D1DOUT(WIPOUT)	RSORC	Precursor concentrations ξ_g by family and group, from data set XS.DELAY

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
PWRAVG	R*8	RMAX if KOUT=3 0 if KOUT<3	D1DOUT(PUTPNT) WIPE(WIPOUT)	PWRCAL	Average power by region for output edit
PWRDEN	R*8	IMAX if KOUT=3 0 if KOUT<3	D1DOUT(PUTPNT) WIPE(WIPOUT)	PWRCAL	Average power by interval for output edits
PWRREG	R*8	RMAX if KOUT=3 0 if KOUT<3	D1DOUT(PUTPNT) WIPE(WIPOUT)	PWRCAL	Power by region for output edits
PXNF	R*8	NFCMP*NGROUP if NFAM≠0 0 if NFAM=0	WSRCH(PUTPNT)	RCORD4 UPDTXS	Prompt $\nu \cdot \sigma_f$ from data set XS.C.MIN, if present, for update of XS.C.MIN after search
R	R*8	IMAX*GMAX	FDCGEN(REDEF) D1DOUT(WIPOUT)	FDC1D	The terms whose reciprocals are the diagonal elements of the lower triangular matrix L of the tri-diagonal matrix A=LU for the Choleski inversion of A
RBOUND	R*8	(RMAX+1)*2	D1DOUT(PUTPNT) WIPE(WIPOUT)	LKCAL	Pairs of region names delineating region interfaces for printing currents in output edits. The first name in each pair is that of the region on the left of the interface; the second name is that of the region on the right.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
REGBRN	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	NRMCAL	Region-averaged power-normalized flux integrals for burnup, computed and written on data set FR.PN if input power factor from A.DIF1D is non-zero
REGFLX	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	NRMCAL	Region-averaged flux integrals by group, for output edit, normalized for k-effective calculations, not normalized for source calculations
REGHHT	R*8	RMAX*2	RGEOM(PUTPNT)	RMESH DELBSQ	Actual region half-height $(H_L)_r$ from data set GEOM, where REGHHT(R,1) is the half-height in the transverse direction and REGHHT(R,2) the half-height in the axial direction for region R. If a region half-height is not defined for a given region R and direction N, or if BSQ(R,N)=0., REGHHT(R,N)=.5. Modified during buckling search.
REGMSH	I*2	IMAX	RGEOM(PUTPNT)	RMESH	Region-to-mesh correspondence, defined in data set GEOM. For any mesh interval I, REGMSH(I) is the number of the region to which I belongs.
REGNAM	R*8	RMAX	RGEOM(PUTPNT)	RMESH	Region names, defined in data set GEOM. REGNAM(R) is the name of the region whose number is R.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
REGVOL	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	FVCAL	Physical region volume for $B^2 \neq 0$, or volume/unit height for $B^2=0$ (from PHYVOL function)
REM	R*8	CMAX*GMAX	RCOMP(PUTPNT)	RGROUP	Removal cross section σ_R by composition and group, from data set XS.C.MIN
				DELXS	Modified during concentration search
				RCORD4 UPDTXS	Removal cross section σ_R for update of XS.C.MIN after search
RMVL	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	REMCAL	Removal contribution to balance by region and group for output edit, including α/v terms
RMVLG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Removal by groups $RMVLG(G) = \sum_{R=1}^{RMAX} RMVL(R,G)$
RMVLR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Removal by regions $RMVLR(R) = \sum_{G=1}^{GMAX} RMVL(R,G)$
RNAME	R*8	NREG	WSRCH(PUTPNT)	RDGEOM	Region names, from record type 9 of data set GEOM, for update of GEOM after search
SCAT	R*8	RMAX*GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SCTCAL	Scattering contribution to balance by region and group for output edit

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
SCATG	R*8	GMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Scattering by groups $SCATG(G) = \sum_{R=1}^{RMAX} SCAT(R,G)$
SCATR	R*8	RMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	SUMCAL	Scattering by regions $SCATR(R) = \sum_{G=1}^{GMAX} SCAT(R,G)$
SCTDUM	R*8	CMAX*GMAX *GMAX	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Array used to read in full scattering cross section before packing into SCTXS from XS.C.MIN
		NCMP*NGROUP *MAXSCT	WSRCH(PUTPNT) WSRCH(WIPOUT)	RCORD4 UPDTXS	Scattering cross section $\sigma_c^{g'g}$ for update of XS.C.MIN after search
SCTLIM	I*2	2*CMAX	RCOMP(PUTPNT)	RGROUP	Scattering array band widths for each composition C. SCTLIM(1,C) is the maximum width of the downscattering band for C, and SCTLIM(2,C) is the maximum width of the upscattering band for C. SCTLIM(1,C) +SCTLIM(2,C)+1 is the maximum band width.
SCTLOC	I*2	CMAX*GMAX	RCOMP(PUTPNT)	DIMSCT PCKSCT	Location of specified scattering band in SCTXS. For a given composition C and group G, SCTLOC(C,G) gives the location of the band of elements $\sigma_c^{g'g}$.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
SCTXS	R*8	SCTDIM*GMAX	RCOMP(PUTPNT)	PCKSCT	Scattering cross section $\sigma_c^{g'g}$ by composition, band width, and group, packed from SCTDUM
				DELXS	Modified during concentration search
SETCHI	R*8	ISCHI*GMAX	RCOMP(PUTPNT)	GETCHI	Set-wide fission spectrum $\chi^{g'g}$, if one is defined from data set XS.C.MIN. Values are transferred into the CHI array for compositions using the set-wide $\chi^{g'g}$.
TOTSRC	R*8	IMAX	KEFCAL(REDEF) D1DOUT(WIPOUT)	TSRC (entry in ITSRC)	Total source by mesh interval for one group, including fission source and scattering source during iterations
UPSCT	4	GMAX	RCOMP(PUTPNT) RCOMP(WIPOUT)	RGROUP	Number of groups of upscattering given for each group in a given composition block in XS.C.MIN
VEL	R*8	GMAX	RCOMP(PUTPNT)	RVEL	Neutron speeds for kinetics option 2, read from data set XS.ISO2 or XS.ISO. VEL(G) = 1 for all groups if data is not needed or not present.
VOLFRC	R*8	MCSLTH	WSRCH(PUTPNT)	UDBHMG	Volume fraction table, from record type 4 of data set B.HOMOG
X	R*8	IMAX+1	RGEOM(PUTPNT)	RMESH	Mesh line locations, from data set GEOM
XA	R*8	NCMP*NGROUP	WSRCH(PUTPNT) WSRCH(WIPOUT)	RCROD4 UPDTXS	Absorption cross section σ_A, for update of data set XS.C.MIN after search

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
XBAR	R*8	IMAX	D1DOUT(PUTPNT) WIPE(WIPOUT)	NRMCAL	Mesh interval centers for output prints XBAR(I) = X(I) + .5*DX(I)
XMESH	R*8	NPTI	WSRCH(PUTPNT)	RDGEOM	Original mesh line locations in X-direction, from record type 2 of data set GEOM, for update of GEOM after search
XTOT	R*8	NCMP*NGROUP	WSRCH(PUTPNT) WSRCH(WIPOUT)	RCORD4 UPDTXS	Total cross section σ_{TOT}, for update of data set XS.C.MIN after search
XTR	R*8	NPTI	WSRCH(PUTPNT) WSRCH(WIPOUT)	RCORD4 UPDTXS	Transport cross section σ_{TR}, for update of data set XS.C.MIN after search
YMESH	R*8	NPTJ	WSRCH(PUTPNT)	RDGEOM	Original mesh line locations in Y-direction, from record type 2 of data seg GEOM, for update of GEOM after search
ZONE	R*8	NZONE	WSRCH(PUTPNT)	RDGEOM	Zone names from record type 9 of data set GEOM, for update of GEOM after search

7. D1DSCH (NUC004) Data Set Usage

This section contains descriptions of all data sets used by this module. For each data set, the following information is given:

1. the name of the data set, if it is an ARC System data set, or a description of its contents and function if it is a scratch data set used only within the module;
2. the record structure of the data set, if it is a scratch data set;
3. names of subprograms in which the data set is read and/or written, along with a list of the record types which are read and/or written and the name of the variable containing the data set reference number.

A complete description of the record structure and contents of each ARC System data set can be found in Volume I of this series.

Name or description of data set - A.DIF1D, One-dimensional Diffusion Theory Module-dependent BCD Input

Record structure - ARC System data set

Read in subprograms	Card types	Reference variable
RCDEP	One of Problem Title (first type 01), one of Problem Options (second type 01), one of Upscatter Iteration and Normalization Control (type 02), one of Iteration and Array Control (type 03), and one of Convergence Criteria (type 04)	N
RBLACK	As many of Black Region Control (type 07) as are present	N

Written in subprograms - None

Name or description of data set - BC, Boundary Condition Specifications

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RBNDC	One of Specifications (type 1) and all of Boundary Conditions (type 2) for vertical boundaries	NTAPE

Written in subprograms - None

Name or description of data set - FA.D1, One-dimensional Adjoint Group Fluxes

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RFLUX	One of Specifications (type 1) and one of One-dimensional Adjoint Flux (type 2)	NTAPE
FLXIN	Same as above	N1

Written in subprograms	Record types	Reference variable
OUTERS	One of Specifications (type 1) and one of One-dimensional Adjoint Flux (type 2)	N1
NRMCAL	Same as above	NFAD1

Name or description of data set - FR.D1, One-dimensional Real Group Fluxes

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RFLUX	One of Specifications (type 1) and one of One-dimensional Real Flux (type 2)	NTAPE

Written in subprograms	Record types	Reference variable
OUTERS	One of Specifications (type 1) and one of One-dimensional Real Flux (type 2)	N1
NRMCAL	Same as above	NFRD1

Name or description of data set - FR.PN, Power-normalized Real Regional Group Fluxes

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
NRMCAL	One of Specifications (type 1) and one of Power-normalized Regional Group Flux (type 2)	NFRPN

Name or description of data set - FSA.D1, One-dimensional Adjoint Fission Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
NRMCAL	One of Specifications (type 1) and one of One-dimensional Adjoint Fission Source (type 2)	NFSAD1

Name or description of data set - FSR.D1, One-dimensional Real Fission Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
NRMCAL	One of Specifications (type 1) and one of One-dimensional Real Fission Source (type 2)	NFSRD1

Name or description of data set - GEOM, Geometry Data

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RGEOM	One of Specifications (type 1)	NTAPE
RMESH	One of Boundaries (type 2), one of Region-Interval Correspondence (type 3), one of Composition-Interval Correspondence (type 4), one of Composition-Region Correspondence (type 5), one of Area Definitions (type 6) skipped if present, one of Zone Definitions (type 7) skipped if present, one of Finite Geometry Data (type 8) if present, and one of Labels (type 9)	NTAPE

Read in subprograms	Record types	Reference variable
WSRCH	One of type 1	N3
RDGEOM	One each of whichever of types 2 - 9 are present	N3

Written in subprograms	Record types	Reference variable
WSRCH	One of type 1	N3
WTBSQ WTDIM (entries in RDGEOM)	One each of whichever of types 2 - 9 are present, and one of Region Volumes (type 10)	N3

Name or description of data set - SP.CICN, Module-independent Data

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RCDEP	One of Specifications (type 1)	NS

Written in subprograms - None

Name or description of data set - SP.CRIT, Criticality Search Specifications

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
D1DSCH	One of Specifications (type 1)	NCRIT
CRTIN1	One of Concentration Search Modifiers (type 2)	NCRIT
CRTIN2	One of Dimension Search Modifiers (type 3) or Buckling Search Modifiers (type 4)	NCRIT
RDCONC	One of Concentration Search Modifiers (type 2)	NCRIT

Read in subprograms	Record types	Reference variable
RDDDX	One of Dimension Search Modifiers (type 3)	NCRIT
RDBSQ	One of Buckling Search Modifiers (type 4)	NCRIT

Written in subprograms	Record types	Reference variable
D1DSCH	One of Specifications (type 1)	NCRIT
CRTOT1 (entry in CRTIN1)	One of Concentration Search Modifiers (type 2)	NCRIT
CRTOT2 (entry in CRTIN2)	One of Dimension Search Modifiers (type 3) or one of Buckling Search Modifiers (type 4)	NCRIT

Name or description of data set - XS.C.MIN, Macroscopic Composition Cross Sections

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RCOMP	One of Specifications (type 1 of file 1)	NTAPE
GETCHI	One of Set Chi (type 2 of file 2) if present	NTAPE
RGROUP	One composition block, consisting of one record of Composition Specifications (type 3 of file 2) and GMAX records of Composition Macroscopic Group Cross Sections (type 4 of file 2)	NTAPE
WSRCH	One of Specifications (type 1)	N1
RCORD1	One of Specifications (type 1)	N1
RCORD2	One of Set Chi (type 2) if present	N1

Read in subprograms	Record types	Reference variable
RCORD3	One of Composition Specifications (type 3)	N1
RCORD4	One of Composition Macroscopic Group Cross Sections (type 4)	N1

Written in subprograms	Record types	Reference variable
WRCRD1	One of Specifications (type 1)	N2
WRCRD2	One of Set Chi (type 2) if present	N2
WRCRD3	One of Composition Specifications (type 3)	N2
WRCRD4	One of Composition Macroscopic Group Cross Sections (type 4)	N2
		(Note: N2=N1)

Name or description of data set - XS.ISO2, Problem-dependent Microscopic Group Cross Sections, or XS.ISO, Microscopic Group Cross Sections

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
RVEL	One of File Size (type 1 of file 1), one of Isotope Names (type 2 of file 1) skipped, and one of Group Structure (type 3 of file 1)	NTAPE

Written in subprograms - None

---

8. D1DSCH (NUC004) Error Messages

This section lists the error messages generated by module D1DSCH in alphabetical order by subprogram or entry name. For each error message, the significance of the message is given, along with the action taken by the subprogram.

In addition to the errors listed, execution of the module can be terminated for an error in POINTR dynamic storage array allocation. The names of the subprograms in which such an error is tested for, and from which an error return can occur, are

DELBSQ
DELX
DIMSCT
FDC1D
ISUBS
PCKSCT
PCOMP
RBNDC
RDELAY
REORDR
RFLUX
RGROUP
RMESH
RVEL

No message except the one that execution has been terminated will appear for an error in dynamic storage array allocation.

CELCAL

Message: *****GENERAL INTERNAL INTERFACE CONDITIONS SPECIFIED FOR I=iiii, G=iiii

ERROR NUMBER $\left\{\begin{matrix}1\\2\end{matrix}\right\}$ IN SUBROUTINE CELCAL

Significance: An interface condition (ITYPER or ITYPEL) not equal to 0, 1, or 2 was returned to CELCAL from VERBND (entry in GETBND). This is assumed to indicate that an interface condition of 3, indicating general internal interface conditions, was specified in data set BC, and hence by implication that the internal interface condition type 2 was specified in A.NIP (card type 10, cols. 19-24), for this I and G. An error number 1 indicates that the error was encountered on the right of cell i,g; an error number 2, that it was encountered on the left.

Action taken: The IERR flag is set to 1 and control is returned to the calling program. No calculations are performed.

CHANGE

Message: *****BOUNDS ON CONTROL PARAMETER EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE CHANGE

Significance: The search control parameter computed for the next pass is not within the bounds set for the control parameter; the condition XL≤XNTH≤XU is not met.

Action taken: Execution is terminated.

Message: PARABOLIC EXTRAPOLATION NOT FEASIBLE--LINEAR EXTRAPOLATION USED

Significance: This is merely an information message. The user has specified parabolic extrapolation, but it is not feasible, either because the denominator used in solving for the parabolic fit by means of Cramer's rule is too small ($\leq 1.0 \cdot 10^{-8}$), indicating that accuracy might be better with linear extrapolation; because the quadratic coefficient a is near zero and should not be used as a divisor; or because the discriminant b^2-4ac of the quadratic equation $ax^2+bx+c - k_{eff}^{(0)} = y$ is less than zero.

Action taken: Linear interpolation is used.

D1DSCH

Message: *****DATA SET SP.CRIT NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE D1DSCH

Significance: This data set could not be found by the ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****MAXIMUM NUMBER OF CONTROL PASSES EXCEEDED DURING SEARCH

ERROR NUMBER 2 IN SUBROUTINE D1DSCH

Significance: The number of search passes NPMAX defined in data set SP.CRIT has been exceeded, and k_{eff} has not converged according to the search criterion.

Action taken: Execution is continued as though problem had converged.

Message: *****ERROR ENCOUNTERED IN WRITING DATA SETS MODIFIED BY SEARCH

ERROR NUMBER 3 IN SUBROUTINE D1DSCH

Significance: An error was encountered in updating either XS.C.MIN and B.HOMOG or GEOM, depending on the type of search, after the search.

Action taken: Execution is terminated.

Message: *****EXECUTION OF MODULE NUC004 HAS BEEN TERMINATED

Significance: An error has been encountered in the execution of the module which made successful execution of the module impossible.

Action taken: The error flag IQUIT is set to 1 and control is returned to the main program of NUC004.

DELBSQ

Message: *****A NEGATIVE VALUE HAS BEEN COMPUTED FOR BUCKLING FOR REGION aaaaaa DIRECTION iii

BUCKLING =-x.xxxxxxExxx

ERROR NUMBER 1 IN SUBROUTINE DELBSQ

Significance: The negative buckling value printed has been generated in the region named, where the direction number is 1 for the transverse direction and 2 for the axial direction.

Action taken: Execution is terminated.

DELX

Message: *****THE LENGTH OF THE iiiiTH INTERVAL IS NEGATIVE
INTERVAL LENGTH = -x.xxxxxxExxx

ERROR NUMBER 1 IN SUBROUTINE DELX

Significance: The negative interval length value printed has been generated in the interval whose number is given.

Action taken: Execution is terminated.

DELXS

Message: *****NON-FISSIONABLE COMPOSITION CANNOT BE MODIFIED BY A FISSIONABLE COMPOSITION
NON-FISSIONABLE COMPOSITION CAN BE MADE TO APPEAR FISSIONABLE
BY ADDING ZERO VOLUME-FRACTION OF A FISSIONABLE MATERIAL
ON TYPE 14 CARD OF DATA SET A.NIP

ERROR NUMBER 1 IN SUBROUTINE DELXS

Significance: Self-explanatory. Storage space for fission-related cross sections is not allocated unless the composition appears fissionable.

Action taken: Execution is terminated.

Message: *****NUMBERS OF GROUPS OF DOWN SCATTERING AND OF UP SCATTERING FOR ORIGINAL COMPOSITION
MUST BE EQUAL TO OR GREATER THAN THOSE FOR FOR MODIFIER COMPOSITION
THIS CAN BE ACCOMPLISHED BY ADDING ZERO VOLUME-FRACTION OF
MODIFIER COMPOSITION TO ORIGINAL ON TYPE 14 CARD OF DATA SET A.NIP

ERROR NUMBER 2 IN SUBROUTINE DELXS

Significance: Self-explanatory. Storage space for scattering for a given composition and group is only as wide as the band width.

Action taken: Execution is terminated.

ENDOUT (entry in INDOUT)

Message: *****FLUX = -x.xxxxxxxExx FOR I=iiii G=iiii
ABSOLUTE VALUE OF FLUX WILL BE USED

ERROR NUMBER 1 IN SUBROUTINE ENDOUT

Significance: The flux for the specified mesh interval and group has gone negative following Chebyshev acceleration. It is assumed that the flux is very close to zero for this to have occurred.

Action taken: Flux for this mesh interval and group is set to the absolute value of the flux.

Message: *****OUTER ITERATIONS HAVE DIVERGED TWELVE TIMES IN SUCCESSION

ERROR NUMBER 2 IN SUBROUTINE ENDOUT

Significance: Following Chebyshev extrapolation, the iterations have diverged on twelve successive passes.

Action taken: IRETRN is set to -1, and execution is terminated.

FLXIN

Message: ERROR NUMBER 1 IN SUBROUTINE FLXIN

Significance: The data set FA.D1 could not be found.

Action taken: Execution is terminated.

Message: *****ERROR IN READING FLUX GUESS******
IMAX=iiii IMX1=iiii
GMAX=iiii GMX1=iiii

ERROR NUMBER 2 IN SUBROUTINE FLXIN

Significance: The number of energy groups (GMX1) and/or the number of mesh intervals (IMX1) defined in the data set FA.D1 do not correspond to the values (GMAX,IMAX) for the current problem.

Action taken: Execution is terminated.

OUTERS

Message: *****OUTER ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE OUTERS

Significance: The maximum number of outer iterations, NMAX, has been exceeded, and the problem has not converged.

Action taken: Execution of the remainder of the program, consisting of the output section, is performed as if the problem had converged.

PCOMP

Message: *****SEARCH CALCULATED A NEGATIVE aaaaaaaa VALUE
COMPOSITION SEARCH TERMINATED

ERROR NUMBER 1 IN SUBROUTINE PCOMP

where aaaaaaaa is a cross-section type:

'DIF COEF', 'REMOVAL', 'FISSION', 'NU FISS', 'CHI MTRX', or 'SCATTER'

Significance: In adding the modifier composition to the original composition on a search pass, the search calculated a negative value or values of some cross-section type or types (only the most recently calculated type being indicated).

Action taken: The concentration search is terminated, without output of data sets.

PRTBC

Message: *****CALCULATION TYPE IS aaaaaaaa
BOUNDARY CONDITIONS INDICATE CALCULATION SHOULD BE INHOMOGENEOUS

ERROR NUMBER 1 IN SUBROUTINE PRTBC

where aaaaaaaa is the calculation type in PROBT

Significance: One or more inhomogeneous boundary or interface conditions have been defined, and the calculation type is not source. Either the boundary conditions or the calculation type is in error.

Action taken: Execution is terminated.

RBNDC

Message: *****DATA SET BC NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RBNDC

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****LESS THAN TWO BOUNDARY CONDITIONS DEFINED. NBCDEF=iii

ERROR NUMBER 2 IN SUBROUTINE RBNDC

Significance: Both external boundary conditions, at least, must be defined, and two boundary conditions are not defined in data set BC. NBCDEF is the number of boundary conditions defined.

Action taken: Execution is terminated.

RCDEP

Message: *****DATA SET A.DIF1D NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RCDEP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****DATA SET SP.CICN NOT FOUND

ERROR NUMBER 2 IN SUBROUTINE RCDEP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

RCOMP

Message: *****DATA SET XS.C.MIN NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RCOMP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NO END-OF-FILE FOUND FOR FIRST FILE OF XS.C.MIN

ERROR NUMBER 2 IN SUBROUTINE RCOMP

Significance: The end of the first file of this data set was not found, and the second file thus could not be read.

Action taken: Execution is terminated.

Message: *****DATA SET XS.DELAY NOT FOUND

ERROR NUMBER 3 IN SUBROUTINE RCOMP

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****NUMBER OF FAMILIES DEFINED FOR DELAYED DATA IS ZERO

ERROR NUMBER 4 IN SUBROUTINE RCOMP

Significance: The number of families NFAM defined in XS.C.MIN was zero, and the kinetics option indicates that delayed data is desired for an α-calculation.

Action taken: Execution is terminated.

RCORD1

Message: *****ERROR IN READING RECORD 1 OF XS.C.MIN ON DSRN iii

ERROR NUMBER 1 IN SUBROUTINE RCORD1

Significance: An error was encountered in attempting to read the first record of this data set.

Action taken: Execution is terminated.

RCORD2

Message: *****ERROR IN READING RECORD 2 OF XS.C.MIN ON DSRN iii

ERROR NUMBER 1 IN SUBROUTINE RCORD2

Significance: An error was encountered in attempting to read the second record of this data set.

Action taken: Execution is terminated.

RCORD3

Message: *****ERROR IN READING RECORD 3 OF XS.C.MIN FOR COMPOSITION iii ON DSRN iii

ERROR NUMBER 1 IN SUBROUTINE RCORD3

Significance: An error was encountered in attempting to read record type 3 of this data set for the composition whose number is given.

Action taken: Execution is terminated.

RCORD4

Message: *****iiiTH GROUP RECORD FOR COMPOSITION NO iiii IS OUT OF ORDER
GROUP NUMBER ON RECORD IS iiii

ERROR NUMBER 1 IN SUBROUTINE RCORD4

Significance: In reading the group records for the composition whose number is given, an out-of-order record was encountered. The first number is that of the group which was expected; the last number is the group number appearing on the record.

Action taken: Execution is terminated.

Message: *****ERROR IN READING RECORD 4 OF XS.C.MIN FOR COMPOSITION iii
AND GROUP iiii ON DSRN iii

ERROR NUMBER 2 IN SUBROUTINE RCORD4

Significance: An error was encountered in attempting to read record type 4 of this data set for the composition and group whose numbers are given.

Action taken: Execution is terminated.

Message: *****COMPOSITION NO iii USES A SET-WIDE CHI
BUT NO SET-WIDE CHI IS GIVEN

ERROR NUMBER 4 IN SUBROUTINE RCORD4

Significance: No set-wide fission spectrum is defined on data set XS.C.MIN, but the composition whose number is given expects one.

Action taken: Execution is terminated.

RDBSQ

Message: *****SP.CRIT B**2 MODIFIER DATA RECORD IS WRONG LENGTH
RMX1=iiii NIL1=iiii

ERROR NUMBER 1 IN SUBROUTINE RDBSQ

Significance: One or both of the numbers determining the length of the buckling modifier data record does not agree with the corresponding number defined for the problem in data set GEOM; RMX1 does not agree with RMAX, and/or NIL1 does not agree with NIL.

Action taken: Execution is terminated.

RDDDX

Message: *****SP.CRIT DX MODIFIER DATA RECORD IS WRONG LENGTH
IMX1=iiii

ERROR NUMBER 1 IN SUBROUTINE RDDDX

Significance: The number of mesh intervals IMX1 defined in data set SP.CRIT does not agree with the number IMAX defined for the problem in data set GEOM.

Action taken: Execution is terminated.

RDIST

Message: *****DATA SET XS.DELAY NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RDIST

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written. (This message will not appear for a search, since it applies only to source calculations.)

Action taken: Execution is terminated.

Message: *****NUMBER OF FAMILIES DEFINED FOR DELAYED DATA IS ZERO

ERROR NUMBER 2 IN SUBROUTINE RDIST

Significance: The number of families, NFAM defined in XS.C.MIN was zero, and the kinetics option indicates delayed neutron source data is required. (This message will not appear for a search, since it applies only to source calculations.)

Action taken: Execution is terminated.

RFLUX

Message: *****DATA SET FR.D1 NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RFLUX

or

*****DATA SET FA.D1 NOT FOUND

ERROR NUMBER 2 IN SUBROUTINE RFLUX

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****FLUXES DIMENSIONED IMPROPERLY ON FR.D1 OR FA.D1
IMX1=iiii GMX1=iiii

ERROR NUMBER 3 IN SUBROUTINE RFLUX

Significance: The number of mesh intervals (IMX1) and/or of groups (GMX1) defined on this data set are not the same as the corresponding numbers defined on data sets GEOM and XS.C.MIN.

Action taken: Execution is terminated.

RGEOM

Message: *****DATA SET GEOM NOT FOUND

ERROR NUMBER 1 IN SUBROUTINE RGEOM

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: Execution is terminated.

Message: *****GEOMETRY TYPE CANNOT BE IDENTIFIED. NGEOM=iiii

ERROR NUMBER 2 IN SUBROUTINE RGEOM

Significance: The geometry type number read from data set GEOM is not 1, 2, or 3. It may be a two-dimensional type (4, 5, 6, 7, or 8), or may be meaningless. The geometry type selected in data set A.NIP (card type 3, cols. 13-18) should be checked.

Action taken: Execution is terminated.

RGROUP

Message: *****TRANSPORT CROSS-SECTION IS ZERO--DIFFUSION COEFFICIENT IS SET TO 1.0 E+15
FOR COMPOSITION aaaaaaaa, GROUP iii

Significance: A transport cross section of zero for this composition and group was read from data set XS.C.MIN, and hence the diffusion coefficient, $D_c^g = \frac{1.}{3(\sigma_{TR})_c^g}$, could not be calculated. The diffusion coefficient is set to 1.0×10^{15}.

Action taken: Processing continues. If this composition is found to be defined as black for this group the diffusion coefficient will be set to zero, as it should be.

RVEL

Message: *****NUMBER OF GROUPS DEFINED IN XS.ISO OR XS.ISO2
DOES NOT AGREE WITH NUMBER DEFINED IN GEOM
DSRN=iii GMAX=iii NGROUP=iii

ERROR NUMBER 1 IN SUBROUTINE RVEL

Significance: As stated, the number of groups defined in the two data sets do not agree. DSRN is the data set reference number associated with XS.ISO or XS.ISO2, GMAX is the number of groups defined in GEOM, and NGROUP is the number defined in XS.ISO or XS.ISO2.

Action taken: Execution is terminated.

SRCHIN

Message: *****ALPHA SEARCH TYPE AND SEARCH TO PRESCRIBED ALPHA HAVE BEEN SPECIFIED

ERROR NUMBER 1 IN SUBROUTINE SRCHIN

Significance: A prescribed value of alpha has been input, and the search quantity to be modified is also alpha.

Action taken: Execution is terminated.

Message: *****SEARCH TYPE NOT RECOGNIZED. TYPE=iiii

ERROR NUMBER 2 IN SUBROUTINE SRCHIN

Significance: A search type KSRCH not between 1 and 6, inclusive, has been input. KSRCH is printed.

Action taken: Execution is terminated.

Message: *****TWO-DIMENSIONAL SEARCHES NOT ALLOWED. TYPE=aaaaaaaa

ERROR NUMBER 3 IN SUBROUTINE SRCHIN

Significance: An X-dimension search or X- and Y-dimension search (KSRCH=3 or 6) has been specified.

Action taken: Execution is terminated.

SWEEP (entry in ISWEEP)

Message: *****UPSCATTERING ITERATION LIMIT EXCEEDED

ERROR NUMBER 1 IN SUBROUTINE SWEEP

Significance: The maximum number of upscatter iterations LMAX has been exceeded and the iterations have not converged.

Action taken: Execution is continued as though iterations had converged.

WSRCH

Message: *****DATA SET XS.C.MIN COULD NOT BE UPDATED
DSRN FOR OLD=iiii DSRN FOR NEW=iiii

ERROR NUMBER 1 IN SUBROUTINE WSRCH

Significance: One of the two data sets could not be found by ARC System routine SNIFF. This implies either that the old one was not written or that the new one was not set up. In practice, currently, both data sets have the same data set reference number, since the old is written over the new.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: *****ERROR IN READING RECORD 1 OF XS.C.MIN ON DSRN iii

ERROR NUMBER 2 IN SUBROUTINE WSRCH

Significance: An error was encountered in attempting to read the first record of this data set.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: ERROR NUMBER 3 IN SUBROUTINE WSRCH

Significance: An error was encountered in attempting to read some record of data set XS.C.MIN.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: *****DATA SET GEOM NOT FOUND

ERROR NUMBER 4 IN SUBROUTINE WSRCH

Significance: This data set could not be found by ARC System routine SNIFF. This implies that it has not been written.

Action taken: The IERR flag is set to 1 and execution is terminated.

Message: ERROR NUMBER 5 IN SUBROUTINE WSRCH

Significance: An end-of-file was encountered after the second record of B.HOMOG. The data set was re-wound, and successive attempts to re-read record 1 resulted in an end-of-file being read. (This is highly unlikely.)

Action taken: The IERR flag is set to 1 and execution is terminated.

9. D1DSCH (NUC004) Timing Information

Times for some representative problem runs are given in the table below. For each problem the search type, number of search passes, number of groups, number of regions, number of compositions present and number used, type of fission spectrum χ present, number of mesh intervals, and time in minutes are given.

D1DSCH (NUC004) Sample Problem Times

Search Type	Number of Passes	Number of Groups	Number of Regions	Number of Compositions Present	Number of Compositions Used	Type of χ	Number of Mesh Intervals	Time (min.)
Real concentration	4	22	4	5	4	vector	130	1.56
Real concentration followed by adjoint calculation	7	22	4	10	4	matrix	12	1.00
Real concentration followed by adjoint calculation	9	22	4	10	4	matrix	12	1.15
Real concentration followed by adjoint calculation	5	22	4	7	3	vector	12	.53
Buckling followed by adjoint calculation	6	6	4	3	3	vector	16	.60
X-dimension followed by adjoint calculation	6	6	4	3	3	vector	16	.54

10. D1DSCH (NUC004) Sample Output

The following pages represent output from a sample D1DSCH run. For a detailed explanation of the output, see section 2.d.

The module begins by printing a message that it has been entered. The first fifteen pages of data are the same as those described in Part I for DIF1D, since this problem is a concentration search using the same reactor geometry and composition data as that problem. (Page 1 is repeated here.)

On page 16, the input for the concentration search from data set SP.CRIT is printed, along with the sizes of the container array in main core and in bulk for both the input and the calculation sections. For other types of searches, the line "COMPOSITION CP2 MODIFIED BY COMPOSITION CP1" is replaced by descriptions of the appropriate search modifier data. This page is followed by a k-effective iteration history using the first guess at the control parameter X.

Page 19 displays the parameters for the second search pass, including the new search control parameter X to be used, the change in the control parameter, and the value of k-effective generated by the preceding pass. This is again followed by an iteration history.

Search pass 4 is the final pass for this search, since the value of the control parameter used on pass 4 results in the value of k-effective which was being searched for. The data for pass 4 and the final iteration history are shown on pages 23 and 24.

When the concentration search has converged, the composition data are printed in the same format as for the input. (For a buckling or dimension search, the geometry data would be printed instead.) This is followed by the real k-effective calculation output in the form shown in Part I.

Finally, since this is a "BOTH" problem, an adjoint k-effective calculation is performed, and its output is printed. The data sets affected by the search (in this case XS.C.MIN and B.HOMOG) are rewritten, and a message to this effect is printed, followed by a message that the module has been left.

```
*** ENTERING MODULE D1DSCH (NUC004) ***


ONE-DIMENSIONAL DIFFUSION CONCENTRATION SEARCH SAMPLE OUTPUT

                K-EFF  CALCULATION          BOTH    SOLUTION

     NO KINETICS OPTION

                OUTER ITERATION CONTROL DATA

     MAXIMUM NUMBER OF OUTER ITERATIONS =   90

     CHEBYCHEV EXTRAPOLATION ON OUTER ITERATIONS

     DO    5 UNEXTRAPOLATED OUTER ITERATIONS BEFORE BEGINNING TO EXTRAPOLATE

     DO    3 EXTRAPOLATED ITERATIONS BEFORE TESTING FOR PAYOFF

     DO A MAXIMUM OF    8 ITERATIONS IN AN EXTRAPOLATION SEQUENCE

     DO AT LEAST    3 UNEXTRAPOLATED OUTER ITERATIONS
     BEFORE BEGINNING ANOTHER EXTRAPOLATION SEQUENCE

     INTERVAL SHIFT FACTOR FOR CHEBYCHEV EXTRAPOLATION = 0.0

                OUTER ITERATION CONVERGENCE CRITERIA

     DIFFERENCE IN SUCCESSIVE VALUES OF K-EFFECTIVE = 1.000000D-07

     SUM OF ABSOLUTE VALUE OF FLUX DIFFERENCE/SUM OF FLUX = 1.000000D-07

     DIFFERENCE IN UPPER AND LOWER FLUX BOUNDS LAMBDA = 1.000000D-07

     CORE STORAGE ALLOCATED FOR POINTR COMMON ARRAY       =    30000

     BULK (LCS) STORAGE ALLOCATED FOR POINTR COMMON ARRAY =        0


PRESCRIBED K-EFF   =        1.040000D 00

THIS IS A SEARCH RUN OF TYPE CONC

MAXIMUM NUMBER OF CONTROL CHANGES=        10

FIRST GUESS AT CONTROL PARAMETER X=    5.00000D-02

SECOND GUESS AT CONTROL PARAMETER X=    3.00000D-02

DK/DX=    0.0

LOWER BOUND ON CONTROL PARAMETER=   -1.00000D 01

UPPER BOUND ON CONTROL PARAMETER=    1.00000D 01

CONVERGENCE CRITERION=    1.00000D-06

PARABOLIC EXTRAPOLATION ON THE CONTROL SEARCH
```

```
ONE-DIMENSIONAL DIFFUSION CONCENTRATION SEARCH SAMPLE OUTPUT

            COMPOSITION     MODIFIER COMPOSITIONS
          CP2                  CP1

DATA SET SF.CRIT DOES NOT REPRESENT A CONVERGED SEARCH

LENGTH OF SPEC. RECORD =   4


        SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR SEARCH INPUT SECTION =     5630
        SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR SEARCH INPUT SECTION =        0
COMPOSITION CP2       MODIFIED BY COMPOSITION CP1


        SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =     9034
        SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =        0


SEARCH CONTROL PASS    2
NEW CONTROL PARAMETER =  3.0000000D-02
PREVIOUS KEFF =  1.0328634D 00
CHANGE IN CONTROL PARAMETER = -2.0000000D-02
COMPOSITION CP2       MODIFIED BY COMPOSITION CP1


        SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =     9034
        SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =        0


SEARCH CONTROL PASS    4
NEW CONTROL PARAMETER =  2.9026118D-02
PREVIOUS KEFF =  1.0400012D 00
CHANGE IN CONTROL PARAMETER =  3.6166001D-06
COMPOSITION CP2       MODIFIED BY COMPOSITION CP1


        SIZE OF POINTR CORE CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =     9034
        SIZE OF POINTR BULK CONTAINER ARRAY REQUIRED FOR CALCULATION SECTION =        0
```

ONE-DIMENSIONAL DIFFUSION CONCENTRATION SEARCH SAMPLE OUTPUT

ITERATION HISTORY REAL CALCULATION

ITER NO P	KEFF SIGMA	DIF KEFF ENPM1	LAM UPPER SIGMA BAR	LAM LOWER LAM RATIO	DIF LAM ALPHA	DIF FLUX BETA	NGL FLUX
1	1.0400000D 00	-1.2378192D-06	1.0000015D 00	9.9999676D-01	4.7714845D-06	1.5538816D-06	0.0
-4	0.0	0.0	1.5538816D-06	4.7714845D-06	0.0	0.0	
2	1.0400000D 00	-4.3894632D-08	1.0000002D 00	9.9999943D-01	7.5577055D-07	9.3002161D-08	0.0
-3	0.0	0.0	5.6851512D-02	1.5839317D-01	0.0	0.0	
3	1.0400000D 00	3.1173005D-08	1.0000001D 00	9.9999985D-01	2.4739283D-07	3.0347313D-08	0.0
-2	0.0	0.0	3.2630761D-01	3.2733854D-01	0.0	0.0	
4	1.0400000D 00	9.1850414D-09	1.0000000D 00	9.9999995D-01	8.1633550D-08	9.1578026D-09	0.0

CONVERGED KEFF = 1.0400000D 00

ARGONNE NATIONAL LAB WEST
3 4444 00011430 6

B