RETURN TO ANL (IDAHO) LIBRARY.

Argonne National Laboratory

The ARC System Neutronics Input Processor

by

E. A. Kovalsky and D. E. Neal

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-7713
Reactor Technology

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

The ARC System
Neutronics Input Processor

by

E. A. Kovalsky and D. E. Neal

Applied Mathematics Division

Contributors

D. J. Bingaman	Consultant, Applied Physics Division
G. C. Jensen*	Applied Mathematics Division
S. D. Sparck**	Applied Mathematics Division
B. J. Toppel	Applied Physics Division
P. M. Walker	EBR-II Project

January 1971

*Now at Computer Sciences Corporation, Richland, Washington
**Now at Philco-Ford Corporation, Palo Alto, California

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 comprise 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 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. Epithermal 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)

ANL-7723 The ARC System Spatial Flux Synthesis Capability

ANL-7724 The ARC System Two-dimensional Kinetics Capability

TABLE OF CONTENTS

Page

ABSTRACT 13

PART ONE. NEUTRONICS INPUT PROCESSOR MODULE, GNIP (NUI002) 15

1. GNIP (NUI002) Introduction 17
   a. Description of Capabilities 17
   b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path 18
   c. Data Sets Utilized by This Module 20

2. GNIP (NUI002) Description of Problem and Method of Solution. . . . 21
   a. Introduction. 21
   b. Mesh Structure Definition 21
   c. Region, Area, and Zone Definition 22
   d. Buckling Calculations 22
   e. Region Volume Calculations. 23
   f. Output Edits. 24

3. GNIP (NUI002) Description of Module Flow. 27
   a. Overlay Structure Diagram 28
   b. Diagrammatic Subprogram Logic Flow. 29
   c. Descriptive Subprogram Logic Flow 32

4. GNIP (NUI002) Subprogram Descriptions 37
   Subroutine ANISO1 39
   Subroutine ANISO2 42
   Subroutine AREADR 46
   Subroutine ARIA 48
   Subroutine BCRITE 50
   Subroutine BUCK 52
   Subroutine BUCKDR 55
   Subroutine BUCSRH 57
   Subroutine CDSCAN 59
   Subroutine COMREG 65
   Subroutine CONSCH 69
   Subroutine CRCDR. 71
   Subroutine CRITSH 72
   Subroutine DSGEM. 75
   Subroutine DZONE. 78
   Subroutine EXTBND 80
   Subroutine FACEN. 86
   Subroutine FBC. 88
   Subroutine FESD1D 94
   Subroutine FESD1H 96
   Subroutine FESD2D 99
   Subroutine FESD2H 102

TABLE OF CONTENTS (Cont'd)

Page

Subroutine FFAD1. 105
Subroutine FFAD1A 106
Subroutine FFAD2. 107
Subroutine FFAD2A 109
Subroutine FFRD1. 109
Subroutine FFRD1A 111
Subroutine FFRD2. 112
Subroutine FFRD2A 113
Subroutine FGEOM. 114
Subroutine FORMDS 116
Subroutine FSPCCN 119
Subroutine FSPCRT 122
Subroutine GMTYPE 127
Subroutine GNIP 128
Subroutine GROUPN 132
Subroutine GRPSOR 133
Subroutine INITZE 134
Subroutine INTBND 136
Subroutine I1SORD 141
Subroutine I2SORD 146
Main Program MAIN 151
Subroutine MESHC. 152
Subroutine MESHDR 154
Subroutine MESHV. 157
Subroutine MSHORD 159
Subroutine RCMDR. 160
Subroutine RCMESH 162
Subroutine REC10. 165
Subroutine REPORT 168
Subroutine SPACE. 170
Subroutine TEST 172
Subroutine TYPE01 174
Subroutine TYP01A 175
Subroutine TYPE02 177
Subroutine TYPE03 179
Subroutine TYPE04 182
Subroutine TYPE05 185
Subroutine TYP05A 187
Subroutine TYPE06 192
Subroutine TYP06A 194
Subroutine TYPE07 197
Subroutine TYP07A 198
Subroutine TYPE08 201
Subroutine TYP08A 203
Subroutine TYPE09 205
Subroutine TYP09A 207

TABLE OF CONTENTS (Cont'd)

Page

Subroutine TYPE10. 211
Subroutine TYP10A. 213
Subroutine TYPE11. 217
Subroutine TYP11A. 219
Subroutine TYPE12. 223
Subroutine TYP12A. 225
Subroutine TYPE13. 228
Subroutine TYP13A. 230
Subroutine TYPE14. 233
Subroutine TYPE15. 236
Subroutine TYP15A. 238
Subroutine TYPE16. 242
Subroutine TYPE17. 243
Subroutine TYPE18. 244
Subroutine TYPE19. 245
Subroutine TYP19A. 248
Subroutine TYP19B. 251
Subroutine TYPE20. 253
Subroutine TYPE21. 255
Subroutine TYPE22. 258
Subroutine TYPE23. 260
Subroutine TYP23A. 262
Subroutine TYPE24. 264
Subroutine TYP24A. 267
Subroutine TYPE25. 270
Subroutine TYP25A. 272
Subroutine TYPE26. 274
Subroutine TYPE27. 276
Subroutine TYP27A. 278
Subroutine TYPE28. 281
Subroutine TYP28A. 283
Subroutine ZONEDR. 286

5. GNIP (NUI002) Common Block Descriptions. 289
/ARRA/ 290
/BLNAME/ 290
/CTYP02/ 290
/CTYP03/ 290
/CTYP04/ 291
/CTYP05/ 291
/CTYP09/ 291
/CTYP10/ 292
/CTYP21/ 292
/CTYP22/ 293
/CTYP24/ 293
/CTYP26/ 294
/CTYP27/ 294
/GEOM1/. 295
/IERR/ 296
/LOCATE/ 296
/NCARDS/ 296
/XSCM/ 297

TABLE OF CONTENTS (Cont'd)

Page

6. GNIP (NUI002) Dynamic Storage Array Allocation. 299

7. GNIP (NUI002) Data Set Usage. 323

8. GNIP (NUI002) Error Messages. 331

9. GNIP (NUI002) Timing Information. 355

10. GNIP (NUI002) Sample Output 356

PART TWO. GEOMETRY DATA SET (GEOM) COPIER, CPGEOM (AJC006). 361

1. CPGEOM (AJC006) Introduction. 363
   a. Description of Capabilities 363
   b. Standard Paths Which Utilize This Module, and Other Modules Utilized by Each Path 364
   c. Data Sets Utilized by This Module 366

2. CPGEOM (AJC006) Description of Problem and Method of Solution . . . 367

3. CPGEOM (AJC006) Description of Module Flow. 369
   a. Overlay Structure Diagram 370
   b. Diagrammatic Subprogram Logic Flow. 371
   c. Descriptive Subprogram Logic Flow 372

4. CPGEOM (AJC006) Subprogram Descriptions 373
   Main Program MAIN 374
   Subroutine CPGEOM 374
   Subroutine CPMORE 377

5. CPGEOM (AJC006) Common Block Descriptions 379
   /BLKSTR/. 380

6. CPGEOM (AJC006) Dynamic Storage Array Allocation. 381

7. CPGEOM (AJC006) Data Set Usage. 385

8. CPGEOM (AJC006) Error Messages. 387

9. Timing Information. 389

ACKNOWLEDGMENT 390

THE ARC SYSTEM NEUTRONICS INPUT PROCESSOR

by

E. A. Kovalsky and D. E. Neal

ABSTRACT

This document presents the documentation of the ARC System modules GNIP (NUI002) and CPGEOM (AJC006).

NUI002 reads and processes the neutronics module independent BCD card images which define reactor geometry, mesh structure, region and composition assignments, boundary conditions, problem type, criticality search specifications and external source data.

AJC006 is a utility module which copies the ARC system interface data set GEOM from one logical unit to another.

PART ONE.
NEUTRONICS INPUT PROCESSOR MODULE,
GNIP (NUI002)

Formulated by - B. J. Toppel and G. C. Jensen*

Designed by - G. C. Jensen, B. J. Toppel, D. J. Bingaman,
S. D. Sparck,** and E. A. Kovalsky

Programmed by - E. A. Kovalsky, P. M. Walker, and S. D. Sparck

Documented by - E. A. Kovalsky

* Now at Computer Sciences Corporation, Richland, Washington

** Now at Philco-Ford Corporation, Palo Alto, California

1. GNIP (NUI002) Introduction

a. Description of Capabilities

This module reads and analyzes BCD card images which define reactor geometry, mesh structure, region and composition assignments, boundary conditions, problem type, criticality search specifications, and external source data. It then processes these data and writes them on the lower-interface binary data sets listed in 1.c. below.

Insufficient or inconsistent data is noted with an error message and card-image processing continues until a fatal error occurs.

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

Paths Utilizing This Module \ Modules Utilized By Each Path	DELDATA (CSI007) Delayed Neutron Data Preparation	INHOMG (NUI001) Multigroup Cross Section Homog. Specifications	HOMOG (NUC001) Multigroup Cross Section Homog.	HHOMOG (NUC007) Material-to-Composition Cross Section Homog.	GNIP (NUI002) Neutronics Input Processor	DIF1D (NUC002) 1-D Diffusion Theory k_{eff} and Source Calculations (DARC1D)	DIDSCH (NUC004) 1-D Diffusion Theory Criticality Search (DARC1D)	DIF2D (NUC005) 2-D Diffusion Theory k_{eff} and Source Calculations (DARC2D)	D2DSCH (NUC006) 2-D Diffusion Theory Criticality Search (DARC2D)	MODXSISO (NUI006) Run-Time Microscopic Cross-Section Modification	PARC1D (AJC003) 1-D Diffusion Theory Perturbation
STP001 1-D diffusion theory		X	X		X	X	X			X	
STP002 1-D transport theory		X	X		X					X	
STP003 2-D diffusion theory		X	X		X	X	X	X	X	X	
STP004 burnup diffusion theory		X	X	X	X	X	X	X	X		
STP006 1-D diffusion perturbation	X	X	X		X	X	X			X	X
STP007 2-D diffusion perturbation	X	X	X		X			X	X	X	
STP008 spatial flux synthesis		X	X		X						
STP011 2-D transport theory		X	X		X					X	

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

Paths Utilizing This Module \ Modules Utilized By Each Path	PARC2D (AJC004) 2-D Diffusion Theory Perturbation	SYN2D (AJC005) Spatial Flux Synthesis	SNARC1D: SNIN1D (NUI005) 1-D Discrete Ordinate Transport Theory Specifications	SNARC1D: SNARC1D (NUC003) 1-D Discrete Ordinate Transport Theory Calculations	TARGET (NUC008) 2-D Discrete Ordinate Transport Theory Calculations	OUTPUTR (NUE002) 2-D Discrete Ordinate Transport Theory Output	OUTMAN1D (NUE001) 1-D Neutronics Output Manipulation	INVENT1D (AJC001) 1-D Neutron Inventory	INVENT2D (AJC002) 2-D Neutron Inventory	CPGEOM (AJC006) Geometry Data Set (GEOM) Copier	REBUS: INFUEL (FCI001) Fuel Cycle Specifications	REBUS: FUELCY (FCC001) Fuel Cycle Analysis
STP001 1-D diffusion theory							X	X				
STP002 1-D transport theory			X	X			X	X				
STP003 2-D diffusion theory							X	X	X			
STP004 burnup diffusion theory											X	X
STP006 1-D diffusion perturbation							X	X		X		
STP007 2-D diffusion perturbation	X								X	X		
STP008 spatial flux synthesis		X										
STP011 2-D transport theory					X	X			X			

c. Data Sets Utilized by This Module

Upper-interface Data Sets	Records Read
A.NIP, General Input for Neutronics Calculations	All
GEOM, Geometry Data	Specifications (type 1)
XS.C.MIN, Macroscopic Composition Cross-sections, File 1	Specifications (type 1)
XS.ISO, Microscopic Group Cross-sections, File 1	File Size (type 1)

Lower-interface Data Sets	Records Written
BC, Boundary Condition Specifications	All
ES.D1D, One-dimensional Distributed External Source or ES.D2D, Two-dimensional Distributed External Source	All
ES.D1SH, One-dimensional Distributed Surface Source, or ES.D2SH, Two-dimensional Distributed Surface Source	All
FA.D1, One-dimensional Adjoint Group Fluxes, or FA.D2, Two-dimensional Adjoint Group Fluxes	All
FR.D1, One-dimensional Real Group Fluxes, or FR.D2, Two-dimensional Real Group Fluxes	All
GEOM, Geometry Data	All
SP.CICN, Module-independent Data	All
SP.CRIT, Criticality Search Specifications	All

2. GNIP (NUI002) Description of Problem and Method of Solution

a. Introduction

The physical characteristics of a reactor must be defined, analyzed, and prepared in a form acceptable to the computational modules of the ARC System. These data include reactor geometry type, mesh structure, region and composition assignments, boundary conditions, problem type, external source data, and criticality search specifications.

The processing includes the reading and analysis of BCD card images which define the characteristics listed above. These explicit data are processed along with certain quantities derived from the input information.

Output data sets which contain the data that have been processed by the module are written. These data sets, which are listed as lower-interface data sets in section 1.c, are used by the computational modules of the ARC System as module-independent input.

b. Mesh Structure Definition

The reactor mesh structure is set up using the following conventions. In all one-dimensional problems the "x-direction" is the direction of calculation; it is subdivided by (I+1) mesh lines, located as specified in the input. The mesh interval lying between the $(i-1)^{st}$ and i^{th} mesh lines, $1 \leq i \leq I$, is referred to as the ith interval, and the interval boundaries are said to be X_{i-1} on the left and X_i on the right.

For two-dimensional problems, the "x-direction" refers to the first direction specified in the geometry type; thus, in R-Z geometry, r is the "x-direction." The second direction specified is identified as the "y-direction;" it is subdivided by (J+1) mesh lines, defined analogously to the x-direction mesh lines. The mesh interval which lies between the $(i-1)^{st}$ and i^{th} vertical mesh lines, $1 \leq i \leq I$, and the $(j-1)^{st}$ and j^{th} horizontal mesh lines, $1 \leq j \leq J$, is referred to as the $(i,j)^{th}$ interval, and its boundaries are said to be the mesh lines X_{i-1} on the left, X_i on the right, Y_{j-1} on the bottom, and Y_j on the top.

c. Region, Area, and Zone Definitions

A reactor region consists of one or more mesh intervals, not necessarily contiguous in space, as defined by the input data. To each reactor region defined one and only one composition, which is a collection of materials or isotopes, is assigned.

An area is a collection of regions, again not necessarily contiguous, and a zone is a collection of mesh intervals not necessarily contiguous.

d. Buckling Calculations

One of the derived quantities written by GNIP is the buckling $B^2_{r,n}$ for each region r and transverse direction n for finite geometry types. The buckling is computed from the actual half-height of radius $(H_L)_r$, defined as the limit of integration, and the extrapolation distance, defined as the distance from the integration limit at which the flux goes to zero. The extrapolated half-height or radius H_r can be calculated as the sum of these input quantities. If these data are present, the buckling B^2 is defined according to geometry type, as follows.

For one-dimensional finite cylinder or column (rectangular slab with one finite transverse direction) geometry or for two-dimensional X-Y or R-θ geometry, the buckling is

$$B^2_r = \left(\frac{\pi}{2H_r}\right)^2 .$$

For a one-dimensional parallelepiped (rectangular slab with two finite transverse directions), the buckling in each of the two directions n is defined as

$$B^2_{r,n} = \left(\frac{\pi}{2H_{r,n}}\right)^2 .$$

For a one-dimensional finite cylindrical slab, H_r and $(H_L)_r$ represent the extrapolated and actual radii, respectively. In this case

$$B^2_r = \left(\frac{\alpha}{H_r}\right)^2 ,$$

where α is the first zero of the Bessel function J_0, α = 2.40482556.

e. Region Volume Calculations

GNIP also calculates and writes the physical region volumes based on the mesh structure, the region-to-mesh interval correspondence, and the geometry type. For finite geometries for which actual region half-heights or radii $(H_L)_r$ are defined, or for one-dimensional spherical or two-dimensional cylindrical geometries, the volume calculated is the physical volume of each region; for infinite one-dimensional cylinder or slab geometries and infinite two-dimensional X-Y or R-θ geometries, the volume calculated is the region volume per unit height or radius in the transverse direction.

In all cases, the region volume V_r is calculated by first summing the volumes V_i or $V_{i,j}$ per unit transverse height of all mesh intervals i or i,j contained in region r. This sum is then multiplied by the appropriate factor for the transverse direction.

For one-dimensional geometries, the volume per unit height of an interval i is given by

$$V_i = \frac{4}{3}\pi(X_i^3 - X_{i-1}^3) \quad \text{for a sphere,}$$

$$V_i = \pi(X_i^2 - X_{i-1}^2) \quad \text{for a cylinder,}$$

$$V_i = (X_i - X_{i-1}) \quad \text{for a rectangular slab,}$$

and $$V_i = \pi(X_i - X_{i-1}) \quad \text{for a cylindrical slab.}$$

The transverse factor in these geometries is given by

$$T_r = 1 \quad \text{for a sphere, infinite cylinder, or infinite slab,}$$

$$T_r = 2(H_L)_r \quad \text{for a finite cylinder or a column;}$$

$$T_r = 4(H_L)_{r,1}(H_L)_{r,2} \quad \text{for a parallelepiped;}$$

and $$T_r = (H_L)_r^2 \quad \text{for a finite cylindrical slab.}$$

The region volume is given by

$$V_r = T_r \cdot \sum_{i \varepsilon r} V_i \cdot$$

For two-dimensional geometries, a similar formulation holds. Here the volume per unit transverse height of an interval $V_{i,j}$ is defined as

$$V_{i,j} = (X_i - X_{i-1})\ (Y_j - Y_{j-1}) \quad \text{for X-Y or R-}\theta\text{ geometry,}$$

$$V_{i,j} = \pi(X_i^2 - X_{i-1}^2)\ (Y_j - Y_{j-1}) \quad \text{for R-Z geometry,}$$

and $V_{i,j} = \pi(Y_j^2 - Y_{j-1}^2)\ (X_i - X_{i-1})$ for Z-R geometry.

The transverse factor is given by

$$T_r = 2(H_L)_r \quad \text{for finite X-Y or R-}\theta\text{ geometry,}$$

and $T_r = 1$ for R-Z or Z-R geometry or infinite X-Y or R-θ geometry.

Again then,

$$V_r = T_r \cdot \sum_{i,j \varepsilon r} V_{i,j}$$

Note that volumes are calculated only for those portions of the reactor which are calculated explicitly; no account is taken of symmetry conditions.

f. Output Edits

The GNIP printed output optionally contains the following information; an example of the output appears in section 10.

The first edit of input data to appear is that of the type 03 card of A.NIP. The quantities printed are the geometry type, problem type, calculation type, and initial alpha.

This is followed by the type 04 card edit, which consists of the boundary condition types.

If the problem is a criticality search, edits of the type 21 and 22 cards appear. The search specification type, maximum number of passes, prescribed k_{eff} or alpha, and convergence criterion are printed from the type 21 card. The values input for the initial and second guesses for the search parameter X, the derivative of k_{eff} with respect to X, and the upper and lower bounds on X are printed from the type 22 card.

Following the print of the input data is the edit of the output data set GEOM. This begins with a list of all specifications from the first record of the data set, followed by an edit of the mesh interval boundaries defined in the second record. A table of composition numbers by region appears next.

When areas have been defined, Area Definitions (record type 6) of GEOM are displayed. The number of regions in each area, the length of the region index list, and the list of region indices for each portion of each area are printed.

Similarly, when zones have been defined, an edit of Zone Definitions (record type 7) is printed. The number of sections in each zone, the length of the section boundary list, and the list of section boundaries for each zone are given.

If finite geometry data is present, an edit of Finite Geometry Data (record type 8) is then given. This consists of a list of the buckling, actual half-height, and extrapolated half-height by region and transverse direction.

The edits following are lists of the region, composition, area, and zone Labels defined in record type 9, and the Physical Region Volumes in record type 10. All labels appearing on BCD input cards are left-justified, with imbedded blanks removed, before being used or written on record type 9.

The edit of data set GEOM is followed by a mesh point map of the reactor. The mesh points in the x-direction form the bottom boundary of the map, and the mesh points in the y-direction form the left boundary. In the space corresponding to each mesh interval, the names of the region and composition assigned to that interval are printed.

The next edit is that of the data set BC. The Specification record (type 1), and the Boundary Condition records (type 2) for diffusion calculations, are given.

The contents of data set SP.CICN are then printed.

Data set SP.CRIT is displayed for criticality search problems. The Specification record (type 1) is always shown, followed by the contents of the appropriate second record for concentration, dimension, or buckling searches.

For a source calculation, an edit of the appropriate source data set is given. For an external source calculation, all records of ES.DID or ES.D2D are shown; for a surface source calculation, all records of ES.D1SH or ES.D2SH are displayed.

The time spent in GNIP (NUI002) is printed when the execution of the module is complete.

3. GNIP (NUI002) 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 GNIP 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 GNIP module segment.

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. Note that Nnn is the number of cards of type nn present in data set A.NIP, and that NDIM is the number of problem dimensions, i.e., NDIM=1 indicates a one-dimensional problem.

a. Overlay Structure Diagram

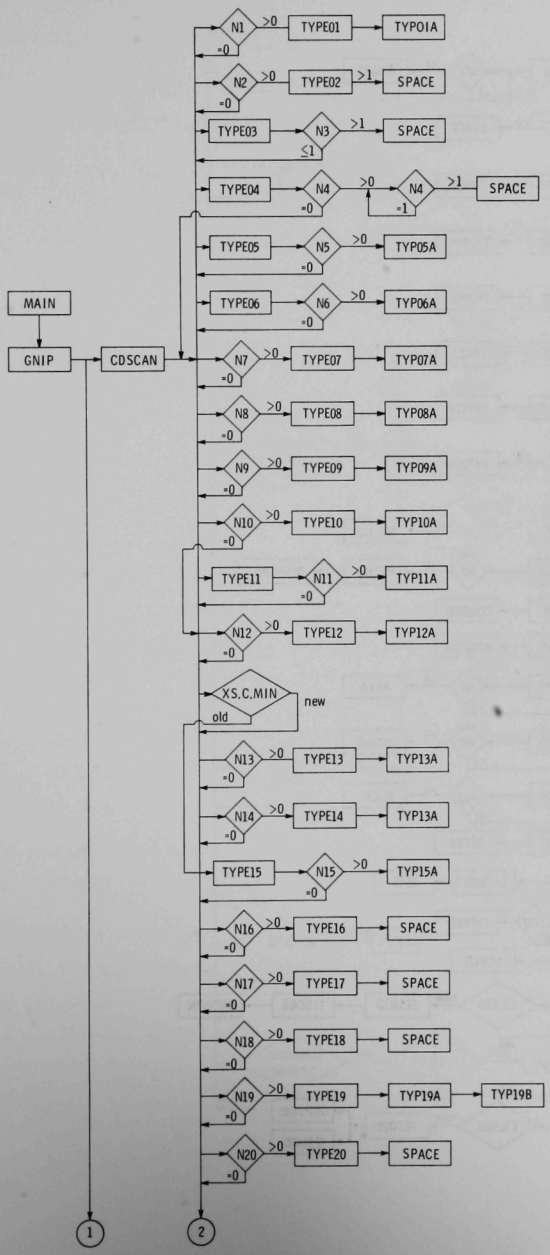
MAIN
SYSTEM
POINTR
SQUEZE
LINK
LOAD
IBCOM
GNIP1
GNIP2
GNIP3
GNIP4
GNIP5
GNIP6
GNIP7
GNIP8
GNIP9
GNIP10
GNIP11
GNIP12
GNIP13
GNIP14

Subprograms contained in each GNIP (NUI002) module segment

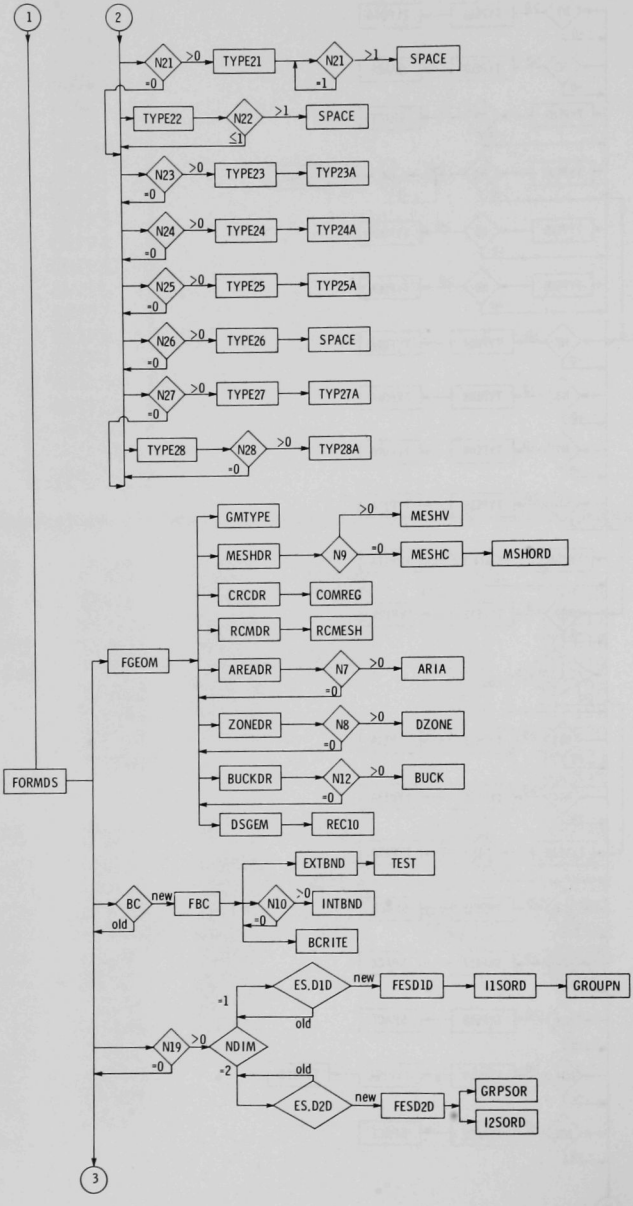
GNIP1	GNIP2	GNIP3	GNIP4	GNIP5	GNIP6	GNIP7
MAIN	SPACE	TYPE06	TYPE10	TYPE12	TYPE16	TYPE23
GNIP	TYPE01	TYP06A	TYP10A	TYP12A	TYPE17	TYP23A
REPORT	TYP01A	TYPE07	TYPE11	TYPE13	TYPE18	TYPE24
CDSCAN	TYPE02	TYP07A	TYP11A	TYP13A	TYPE19	TYP24A
	TYPE03	TYPE08		TYPE14	TYP19A	TYPE25
	TYPE04	TYPE09		TYPE15	TYP19B	TYP25A
	TYPE05	TYP09A		TYP15A	TYPE20	TYPE26
	TYP05A				TYPE21	
					TYPE22	

GNIP8	GNIP9	GNIP10	GNIP11	GNIP12	GNIP13	GNIP14
TYPE27	FGEOM	ZONEDR	CRITSH	EXTBND	FESD1D	FESD1H
TYP27A	GMTYPE	DZONE	BUCSRH	TEST	FESD2D	FESD2H
TYPE28	MESHDR	BUCKDR	CONSCH	INTBND	I1SORD	ANISO1
TYP28A	MESHORD	BUCK	FBC	BCRITE	I2SORD	ANISO2
FORMDS	MESHV	DSGEM			GRPSOR	INITZE
FFAD1A	MESHC	REC10			GROUPN	FACEN
FFAD2A	CRCDR	FSPCCN				
FFRD1A	COMREG	FSPCRT				
FFRD2A	RCMDR					
FFAD1	RCMESH					
FFRD1	AREADR					
FFAD2	ARIA					
FFRD2						

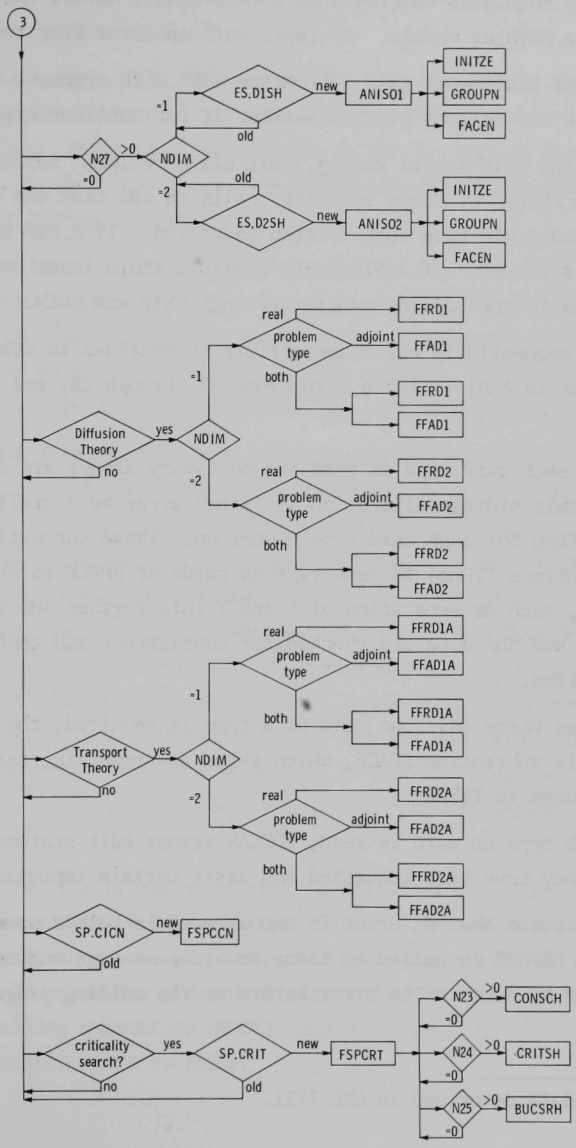
b. Diagrammatic Subprogram Logic Flow



b. Diagrammatic Subprogram Logic Flow (cont'd.)



b. Diagrammatic Subprogram Logic Flow (cont'd.)



c. Descriptive Subprogram Logic Flow

The GNIP module is entered when a CALL LINK ('NUI002' N1) statement appears in a path or module. N1 represents an error flag to the path.

The first statements call subroutine GNIP with argument IERR, which indicates to the calling program whether it can continue execution.

Subroutine GNIP starts timing, initializes POINTR† tables and data set names, then checks by means of SNIFF† calls to see that the necessary old data sets and A.NIP have been written and saved. If A.NIP exists, a call to CDSCAN is issued. If A.NIP doesn't exist, there is an error stop and flow returns to the calling program through GNIP and MAIN.

CDSCAN sequentially calls appropriate subroutines to read each card type present in A.NIP. These card types 01 through 28, are referred to as 'nn' below.

Before each card type is read the necessary arrays are defined in the POINTR dynamic storage allocation container array by a preliminary subroutine TYPEnn for each card type number nn. These subroutines in turn call subroutines TYPnnA to read type nn cards or SPACE to skip them. Any reordering, such as separation of X and Y into further sub-arrays, is performed, and the data are checked for consistency and conformity with specifications.

In cases where only one card of a type is required, the subroutine TYPnnA calls subroutine SPACE, which skips the remaining cards numbered nn and returns to TYPEnn.

As each type nn card is read, CDSCAN issues edit statements and POINTR dumps if they have been requested and lists certain important variables.

In the event that an error is encountered in POINTR array allocation, subroutine REPORT is called to issue an error message and to return immediately by means of an error return to the calling program.

---

†System module described in ANL-7711.

If a fatal error (see section 8) is encountered during a card read, REPORT is called and return is again made to the calling program without attempting to read further in A.NIP.

An error in data which will cause erroneous data sets to be written will allow card reading to continue. At the end of the TYPEnn subroutine the error will be noted. This will cause a call to REPORT, and an error message will be printed. In this case the error is considered serious and causes a skip to the next TYPEnn subroutine.

After all card types are read CDSCAN checks for an error stop. If one has occurred, a return is made via an error return to GNIP, where the error flag returned to the calling program is set to 1 and returned to MAIN and the calling program without any data sets being written. If no error has occurred, CDSCAN returns to subroutine GNIP and processing continues.

Subroutine GNIP then calls FORMDS to write the output data sets. FORMDS always calls FGEOM, which is the driver for the geometry routines. FGEOM calls GMTYPE to establish geometry, problem type, dimension, number of finite directions, and number of integration limits. Then FGEOM calls MESHDR to set up the mesh structure and, if there has not been a serious specification error, calls CRCDR, RCMDR, AREADR, ZONEDR and BUCKDR. A call is made to SNIFF in order to determine if an old GEOM data set exists. If one does, the first record of the old GEOM is compared with the new GEOM quantities. If they do not agree an error is assumed and return is made to the calling program; otherwise, FGEOM returns to FORMDS. When no old GEOM exists, FGEOM calls DSGEM to write a new geometry data set before returning to FORMDS.

MESHDR checks for the presence of type 06 and 09 cards. Card type 09 takes precedence over type 06 and when type 09 cards are present MESHDR calls MESHV to establish a variable mesh structure. If there are no type 09 cards and type 06 cards are present MESHDR calls MESHC, which orders the input points by calling subroutine MSHORD once for each dimension direction. Return is made through MESHDR to FGEOM.

FGEOM then calls CRCDR, which sets up arrays and calls COMREG to define regions and assign compositions to them. After regions and compositions are established return is made to CRCDR and FGEOM.

FGEOM next calls RCMDR. This routine sets up arrays required for its call to RCMESH, which establishes region-mesh correspondence. Return is made through RCMDR to FGEOM.

The next call of FGEOM is to AREADR, which returns if there are no type 07 cards in A.NIP. When type 07 cards are present, arrays are defined and a call is made to subroutine ARIA, where areas are defined and counted. Return is made to AREADR and FGEOM.

FGEOM then calls ZONEDR, which returns if there are no type 08 cards in A.NIP. When type 08 cards are present, arrays are defined and a call is issued to subroutine DZONE. DZONE returns to ZONEDR after counting and defining zones. ZONEDR returns to FGEOM.

FGEOM then calls BUCKDR. If the number of integration limits NIL is equal to zero return is made to FGEOM. If NIL is greater than zero the number of type 12 cards is checked. If there are none an error is noted and return is made to FGEOM. When type 12 cards are present and NIL is greater than zero BUCKDR calls BUCK. Here buckling, half-heights, and transverse distances are defined. Return is made through BUCKDR to FGEOM.

If a call to SNIFF returns a data set number greater than zero, data set GEOM is not written and FGEOM returns to FORMDS.

When no old GEOM exists, DSGEM is called to write the new GEOM data set. DSGEM calls REC10 to calculate region volumes and write record 10. Return is made through FGEOM to FORMDS.

The second call from FORMDS is to SNIFF to determine the existence of an old BC data set. If it doesn't exist, FORMDS calls FBC to write a new BC data set. FBC checks A.NIP for the type of boundary conditions. If they are input boundary condition types 8 through 12, the first record of BC is written and the routine returns to FORMDS. If boundary condition type numbers 1 through 7 are supplied the first record is also written and a call is issued to EXTBND. When type 10 cards are present in A.NIP, subroutine INTBND is called once for a 1-D problem and twice for a 2-D problem. TEST

is called from INTBND and returns to it after checking mesh-boundary correspondence.

After all boundary condition input has been processed, FBC calls BCRITE to write data set BC; BCRITE returns to FBC. If an error has occurred, BC is not written and FBC returns to FORMDS.

If type 19 cards are present for a 1-D problem and ES.D1D has not been written, FESD1D is called by FORMDS. Similarly, if type 19 cards are present for a 2-D problem and ES.D2D has not been written, FESD2D is called by FORMDS.

FESD1D calls I1SORD, which calls GROUPN to establish affected group numbers. I1SORD returns through FESD1D to FORMDS. FESD2D calls GRPSOR and I2SORD to perform the same functions. I2SORD writes ES.D2D.

If no type 19 cards are present for a 1-D problem and type 27 cards exist in A.NIP and ES.D1SH has not been written, FESD1H is called. It in turn calls ANISO1 to write ES.D1SH. If type 27 cards are present for a 2-D problem, a call is made to SNIFF to determine if ES.D2SH has been written. FESD2H is called; it in turn calls ANISO2 to write ES.D2SH. ANISO1 and ANISO2 both call INITZE, and then each calls GROUPN and FACEN for each type 27 card in A.NIP. Return is made through FESD1H or FESD2H to FORMDS.

If the problem type is real, FORMDS calls SNIFF to check for the existence of either data set FR.D1 or data set FR.D2, depending on the problem dimension. If the appropriate flux data set has not been written, subroutine FFRD1 or FFRD2 is called to write the appropriate initial guess flux data set. Then either FR.D1A or FR.D2A is sought. If the data set is present, either FFRD1A or FFRD2A is called. At present, angular fluxes are not written; these subroutines have been included for future use.

For an adjoint problem, SNIFF is called to check for the existence of FA.D1 or FA.D2. If the appropriate one is not found, either FFAD1 or FFAD2 is called to write the appropriate data set. FA.D1A or FA.D2A is sniffed next, and dummy subroutine FFAD1A or FFAD2A is called, as in the real case.

When the problem type is both real and adjoint, FORMDS calls SNIFF to check for the existence of both real and adjoint flux data sets, one- or two-dimensional, and calls the appropriate subroutines to write them. If the problem type is synthesis, FORMDS does not look for flux data sets.

From this point, the flow is the same for the real, adjoint, or synthesis problems. First, data set SP.CICN is sought. If it is present and has not been written, FORMDS calls FSPCCN to write it.

FORMDS next calls SNIFF to check for the existence of data set SP.CRIT. If SP.CRIT has not been written FORMDS calls FSPCRT, which writes the first record of SP.CRIT. When type 26 cards are present, FSPCRT returns to FORMDS. If types 23, 24, or 25 are given, a second record will be written.

If type 25 card input is present FSPCRT will call BUCSRH. If type 23 card input is present FSPCRT calls CONSCH, and when type 24 cards are present FSPCRT calls CRITSH. Each subroutine writes an appropriate second record of SP.CRIT. Return is then made to GNIP, which returns to the calling path with its argument to indicate whether continuation of path execution is possible.

In the data set formation subroutines, those named FXXXXX, error returns are provided, and when an error which would cause execution to fail occurs return is made through the subroutine error return, back through FORMDS. GNIP, and MAIN to the calling program, with the error flag set to 1. For normal termination, the error flag returned is 0.

4. GNIP (NUI002) Subprogram Descriptions

This section contains descriptions of all subprograms which are part of the GNIP (NUI002) module. The descriptions appear in alphabetical order by subprogram name, except that each TYPnnA subroutine follows the corresponding TYPEnn subroutine.

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 two types of variables used are I*4, denoting an integer fullword (32 bits), 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.

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 parenthesis.

Subsection 3 contains a listing of important variables which are local to the subprogram, in a format like that of 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 this subprogram or any of its entry points; subsection 6 lists the subprograms which call this subprogram or any of its entry points.

Subsection 7 describes error messages which can be generated by this subprogram, together with the action taken. A "fatal" error message causes immediate termination of the module; a "serious" error message allows the module to continue checking input data, though output data sets will not be written; a "minor" error does not prevent data sets from being written; and an "informative" message causes no action to be taken.

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

```
Subroutine ANISO1 (NVOL, SALABL, SSLABL, CORDIR, DNCORD, UPCORD, IVERT,
                   VSUR1, VSUR2, VSUR3, XMESH, YMESH, IAHEGN, IALEGN, CISORS,
                   AXSORS, AYSORS, NGROUP, NINTI, NINTJ, NPTI, NPTJ, *)
```

1. Arguments

Name	Type	Dimension	Definition
NVOL	I*4	--	ES.DISH data set reference number
SALABL	R*8	N(28)	Source label on type 28 card of A.NIP
SSLABL	R*8	N(27)	Source label on type 27 card of A.NIP
CORDIR	R*8	N(28)	Coordinate along which face lies
DNCORD	R*8	N(28)	Lower end coordinate of face
UPCORD	R*8	N(28)	Upper end coordinate of face
IVERT	I*4	NPTI*NINTJ	Pointer to position of given source in source output arrays
VSUR1	R*8	NPTI*NGROUP	Output isotropic component of source on 'X' direction face
VSUR2	R*8	NPTI*NGROUP	Output anisotropic 'X' direction source component for 'Y' direction face
VSUR3	R*8	NPTI*NGROUP	Output anisotropic 'Y' direction source component for 'Y' direction face
XMESH	R*8	NPTI	'X' direction mesh coordinates
YMESH	R*8	NPTJ	'Y' direction mesh coordinates
IAHEGN	I*4	N(27)	Higher energy group number to which source applies
IALEGN	I*4	N(27)	Lower energy group number to which source applies
CISORS	R*8	N(27)	Input isotropic component of source
AXSORS	R*8	N(27)	Input anisotropic 'X' direction component of source
AYSORS	R*8	N(27)	Input anisotropic 'Y' direction component of source

Name	Type	Dimension	Definition
NGROUP	I*4	--	Number of energy groups
NINTI	I*4	--	Number of 'X' direction mesh intervals
NINTJ	I*4	--	Number of 'Y' direction mesh intervals
NPTI	I*4	--	Number of 'X' direction mesh lines
NPTJ	I*4	--	Number of 'Y' direction mesh lines
*	--	--	Error return

2. Common Blocks

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag for GNIP output 0 - data set edit is suppressed 1 - data set edit is printed

/CTYP27/

Variables used from this common block

Name	Type	Definition
NORDER	I*4	Anisotropy flag for data set ES.D1SH

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions in geometry
NGEOM	I*4	Type of geometry

3. Local Variables

Name	Type	Definition
I	I*4	Index of type 27 card input arrays
J	I*4	Index on type 28 card input arrays
K	I*4	Group number index of output arrays
L	I*4	Index on line along which face lies
M	I*4	Mesh line index of ends of face
KT	I*4	Counter and index referring to output horizontal edges
LO	I*4	Lower limit of M
IHI	I*4	Upper limit of M
ION	I*4	Index of coordinate along which face lies
NG1	I*4	Lower limit of K index on output quantities
NG2	I*4	Upper limit of K index on output quantities
N27	I*4	Number of type 27 cards in A.NIP
N28	I*4	Number of type 28 cards in A.NIP
NVEDGE	I*4	Number of horizontal sources output

4. Functions and tasks performed by this subroutine

This subroutine defines the one-dimensional shell sources and writes them onto data set ES.D1SH.

Subroutine INITZE is called to initialize the output arrays to zero.

For each type 27 card input, GROUPN is called and the group limits are returned. If NGEOM = 2, NORDER = 2, indicating a cylinder with two angles to be considered.

Then SSLABL(I) on type 27 cards is compared with SALABL(J) on type 28 cards. When a matching label is found, subroutine FACEN is called and it returns shell source mesh limits.

The isotropic and anisotropic shell sources are defined from the input for each affected vertical face and an array of pointers to each given source is defined. This is done as follows:

VSUR1(KT,K) = CISORS(I), and if NORDER is 1 or 2, VSUR2(KT,K) = AXSORS(I). If NORDER is 2, VSUR3(KT,K) = AYSORS(I) and IVERT(L,M) = KT.

After all type 27 card input has been processed, data set ES.D1SH is written using data set reference number NVOL.

5. Subprograms called by this subroutine

INITZE
GROUPN
FACEN

6. Subprograms calling this subroutine

FESD1H

7. Error messages generated by this subroutine - None

Subroutine ANISO2 (NVOL, SALABL, SSLABL, CORDIR, DNCORD, UPCORD, ONCORD, IHORZ, IVERT, HSUR1, HSUR2, HSUR3, VSUR1, VSUR2, VSUR3, XMESH, YMESH, IAHEGN, IALEGN, CISORS, AXSORS, AYSORS, NGROUP, NINTI, NINTJ, NPTI, NPTJ, *)

1. Arguments

Name	Type	Dimension	Definition
NVOL	I*4	--	ES.D2SH data set reference number
SALABL	R*8	N(28)	Source label on type 28 card
SSLABL	R*8	N(27)	Source label on type 27 card
CORDIR	R*8	N(28)	Coordinate on which source face lies
DNCORD	R*8	N(28)	Lower end coordinate of source face
UPCORD	R*8	N(28)	Upper end coordinate of source face

Name	Type	Dimension	Definition
ONCORD	R*8	N(28)	Coordinate along which face lies
IHORZ	I*4	NPTJ*NINTI	Pointer to position of given source in source output arrays HSUR1, HSUR2, HSUR3
IVERT	I*4	NPTI*NINTJ	Pointer to position of given vertical source in VSUR1, VSUR2, VSUR3
HSUR1	R*8	NPTJ*NGROUP	Output isotropic component of given 'X' direction source
HSUR2	R*8	NPTJ*NGROUP	Output anisotropic 'X' direction component of 'X' direction source
HSUR3	R*8	NPTJ*NGROUP	Output anisotropic 'Y' direction component of 'X' direction source
VSUR1	R*8	NPTI*NGROUP	Output isotropic component of given 'Y' direction source
VSUR2	R*8	NPTI*NGROUP	Output anisotropic 'X' direction component of 'Y' direction source
VSUR3	R*8	NINTJ*NGROUP	Output anisotropic 'Y' direction component of 'Y' direction source
XMESH	R*8	NPTI	'X' direction mesh coordinates
YMESH	R*8	NPTJ	'Y' direction mesh coordinates
IAHEGN	I*4	N(27)	Higher energy group number to which source applies
IALEGN	I*4	N(27)	Lower energy group number to which source applies
CISORS	R*8	N(27)	Isotropic component of source
AXSORS	R*8	N(27)	Anisotropic 'Y' direction component of source
AYSORS	R*8	N(27)	Anisotropic 'Y' direction component source
NGROUP	I*4	--	Number of groups
NINTI	I*4	--	Number of 'X' direction mesh intervals
NINTJ	I*4	--	Number of 'Y' direction mesh intervals

Name	Type	Dimension	Definition
NPTI	I*4	--	Number of 'X' direction mesh lines
NPTJ	I*4	--	Number of 'Y' direction mesh lines
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/CTYP02/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag 0 - data set edit is suppressed 1 - data set edit is printed

/CTYP27/

Variables used from this common block

Name	Type	Definition
NORDER	I*4	Anisotropy flag for data set ES.D1SH

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 27 card input arrays
J	I*4	Index on type 28 card input arrays
K	I*4	Output array group index number

Name	Type	Definition
L	I*4	Output face index, i.e., index of coordinate along which face lies
M	I*4	Output mesh point index of end coordinates of face
X	R*8	Alphameric data name 'X'
KT	I*4	Counter and index referring to output horizontal edges
KV	I*4	Counter and index referring to output vertical edges
LO	I*4	Lower limit of index KT or KV
IHI	I*4	Upper limit of index KT or KV
ION	I*4	Index of coordinate along which given source face lies
NG1	I*4	Lower limit of index K
NG2	I*4	Upper limit of index K
N27	I*4	Number of type 27 cards in A.NIP
N28	I*4	Number of type 28 cards in A.NIP
NHEDGE	I*4	Total of 'X' direction sources output
NVEDGE	I*4	Total of 'Y' direction sources output

4. Functions and tasks performed by this subroutine

ANIS02 defines and writes the two-dimensional source data set ES.D2SH.

It calls subroutine INITZE to initialize arrays for each dimension, IHORZ, HSUR1, HSUR2, HSUR3, and then IVERT, VSUR1, VSUR2, and VSUR3.

Labels on the type 28 cards are compared with labels on the type 27 cards. When a matching label is found and the direction is checked, subroutine FACEN is called to return the shell source limits. Isotropic and anisotropic shell sources are defined from the input data arrays for each affected vertical or horizontal face. The arrays of pointers to these sources are also defined.

The quantities are defined as follows:

IHORZ(L,M) = KT

HSUR1(KT,K) = CISORS(I)

and, when anisotropy is present,

HSUR2(KT,K) = AXSORS(I)

HSUR3(KT,K) = AYSORS(I).

Processing then continues to the next type 27 card input.

Arrays IVERT, VSUR1, VSUR2, and VSUR3 are similarly defined. When all input has been processed, data set ES.D2SH is written on reference number NVOL.

5. Subprograms called by this subroutine

INITZE
GROUPN
FACEN

6. Subprograms calling this subroutine

FESH2H

7. Error messages generated by this subroutine - None

Subroutine AREADR (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/GEOM1/

Variables used from this common block

Name	Type	Definition
NAREA	I*4	Number of areas defined
LAREA	I*4	Length of record 7 in data set GEOM

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

3. Local Variables

Name	Type	Definition
NA	I*4	Pointer to array AREA
BLANK	R*8	Alphameric blanks
DUMMY	R*8	Dummy

4. Functions and tasks performed by this subroutine

This subroutine handles defintions of NAREA and LAREA and creates arrays AREA, NREGA, and NRA for the GEOM data set.

When no type 07 cards are present in A.NIP, an array of blank area names is created. NAREA is set to 1 and the length of record 6, LAREA, is set to 0.

If there are type 07 cards AREADR calls subroutine ARIA, after requesting POINTR array space, in order to determine the definition of the above quantities.

In the case of an error in the POINTR request, a message is printed to indicate that there was an array overflow and an error return is used to return through FGEOM, FORMDS, GNIP, and MAIN to the path, thus ending execution.

Normal return is to FGEOM.

5. Subprograms called by this subroutine

ARIA
IGET (entry in POINTR)
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FGEOM

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - SUBROUTINE AREADR, N2 = 'ii'

Subroutine ARIA (AREA, NREGA, NRA, ALABEL, AREGN, REG)

1. Arguments

Name	Type	Dimension	Definition
AREA	R*8	N(7)	Area labels
NREGA	I*4	N(7)	Number of regions in each area
NRA	I*4	10*N(7)	Region indices for given portion of area
ALABEL	R*8	N(7)	Area label on input card type 07 in A.NIP
AREGN	R*8	10*N(7)	Region assignment for portion of area
REG	R*8	NREG	Region labels

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NREG	I*4	Number of regions
NAREA	I*4	Number of areas
LAREA	I*4	Length of record 6 in GEOM, equal to NAREA + LREGA + 1
LREGA	I*4	Length of vector NRA

3. Local Variables

Name	Type	Definition
I	I*4	Index on area labels
J	I*4	Index on region indices
K	I*4	Index on number of areas
L	I*4	Index on number of regions
I1	I*4	Lower limit of region index number in loop
I2	I*4	Upper limit of region index number in loop
N7	I*4	Number of type 07 cards
BLANK	R*8	Blank used for testing

4. Functions and tasks performed by this subroutine

ARIA defines quantities NAREA and LAREA, as well as arrays AREA, NREGA, and NRA, for the GEOM data set.

While the number of different area names given is counted, array AREA, containing area names, and sum NAREA are constructed.

Then array NREGA, of length NAREA, consisting of the number of regions in each area, is filled in while adding to sum LREGA. At the same time array NRA, of length LREGA, consisting of the region index number for each portion of an area, is constructed.

Finally LAREA, the length of the type 6 GEOM record, is defined as NAREA + LREGA + 1.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

AREADR

7. Error messages generated by this subroutine - None

Subroutine BCRITE (NBC, MFACE, M1, M2, NBIT, JI, JF, C1, C2, C3, C4, N1)

1. Arguments

Name	Type	Dimension	Definition
NBC	I*4	--	Total number of boundary condition records of type 2
MFACE	I*4	NBC	Mesh line number and side indicator for which given record applies
M1	I*4	NBC	Line number which defines lower end of interval along given face
M2	I*4	NBC	Line number which defines upper end of interval along given face

Name	Type	Dimension	Definition
NBIT	I*4	NBC	Type number of boundary or interface condition
JI	I*4	NBC	Initial higher-energy group number for which given constants apply
JF	I*4	NBC	Final lower-energy group number for which given constants apply
C1	R*8	NBC	First constant
C2	R*8	NBC	Second constant
C3	R*8	NBC	Third constant
C4	R*8	NBC	Fourth constant
N1	I*4	--	BC data set reference number

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/CTYP02/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag 0 - data set edit is suppressed 1 - data set edit is printed

3. Local Variables

Name	Type	Definition
I	I*4	Index on output records

4. Functions and tasks performed by this subroutine

After all boundary condition quantities have been defined in their respective arrays, this routine writes data set BC.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FBC

7. Error messages generated by this subroutine - None

Subroutine BUCK (BSQ, H, X, DLABEL, ATHD1, TEXD1, ATHD2, TEXD2, REG, NREG)

1. Arguments

Name	Type	Dimension	Definition
BSQ	R*8	NREG*NIL	Output buckling by regions
H	R*8	NREG*NIL	Output actual half-height by region and direction
X	R*8	NREG*NIL	Output H + extrapolation distance
DLABEL	R*8	N(12)	Input region identification label
ATHD1	R*8	N(12)	Input actual transverse half-distance in first direction
TEXD1	R*8	N(12)	Input transverse extrapolation distance in first direction
ATHD2	R*8	N(12)	Input actual transverse half-distance in second direction
TEXD2	R*8	N(12)	Input transverse extrapolation distance in second direction
REG	R*8	NREG	Region names
NREG	I*4	--	Number of regions

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NFIN	I*4	Number of finite geometry type
NIL	I*4	Number of integration limits

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of integration limits
J	I*4	Index on number of regions
K	I*4	Index on number of regions; also index on type 12 card input
PI	R*8	3.141592653
N12	I*4	Number of type 12 cards
BLANKS	R*8	Alphameric blank data

4. Functions and tasks performed by this subroutine

BUCK is called by subroutine BUCKDR to define the quantities in arrays BSQ, H, and X for the GEOM data set.

These arrays having dimensions NREG*NIL are initialized to zero.

If input label DLABEL(K) on card type 12 is blank, then the input is applied to all regions and H(J,1) is defined as ATHD1(K), X(J,1) is defined as ATHD1(K) + TEXD1(K) and

$$BSQ(J,1) = \left(\frac{\pi}{2X(J,1)}\right)^2 ,$$

for each region.

If DLABEL(K) is a region label REG(J), then the quantities are defined as before for the one region J.

However, if NFIN = 1,

$$BSQ(J,1) = \left(\frac{\alpha}{X(J,1)}\right)^2 ,$$

where α = 2.40482556, the first zero of the Bessel function J_0.

If NIL = 2, the following additional quantities are defined for each region J:

$$H(J,2) = ATHD2(K)$$

$$X(J,2) = ATHD2(K) + TEXD2(K)$$

$$BSQ(J,2) = \left(\frac{\pi}{2X(J,2)}\right)^2 .$$

In the event that X(J,I) is zero, the definition is bypassed and BSQ(J,I) remains zero.

If a region is infinite in one direction, the half-height and the extrapolated half-height are defined as follows for the given direction:

$$H(J,1) = 10^{50}$$

and

$$X(J,1) = 10^{50}$$

or

$$H(J,2) = 10^{50}$$

and

$$X(J,2) = 10^{50}$$

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

BUCKDR

7. Error messages generated by this subroutine - None

Subroutine BUCKDR (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NREG	--	Number of regions
NIL	--	Number of integration limits

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy
N2	I*4	Number of POINTR words requested

4. Functions and tasks performed by this subroutine

BUCKDR is called from FGEOM to set up the necessary arrays when there is buckling input.

If NIL is equal to zero it returns immediately to FORMDS.

If N(12) is zero, a message is printed, IQUIT is set to 1, and the error return is used.

PUTPNT calls are issued for BSQ, H, and X and if space is available the pointers to these arrays are extracted and passed to subroutine BUCK. After the return from BUCK, the input arrays are wiped out and BUCKDR returns to FGEOM.

5. Subprograms called by this subroutine

BUCK
IGET (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)

6. Subprograms calling this subroutine

FGEOM

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - SUBROUTINE BUCKDR, N2 = 'ii'

(BUCKDR) - NO TYPE 12 CARDS FOR FINITE GEOMETRY WERE INPUT.

Subroutine BUCSRH (DMODB, NREG, MM, SLABLR, REG, BMOD1, BMOD2, M, *)

1. Arguments

Name	Type	Dimension	Definition
DMODB	R*8	NREG*NIL	Output buckling modifier by region and direction
NREG	I*4	--	Number of regions
MM	I*4	--	Number of type 25 cards in A.NIP
SLABLR	R*8	N(25)	Input region identifier label
REG	R*8	NREG	Region labels
BMOD1	R*8	N(25)	Input buckling modifier in first transverse direction
BMOD2	R*4	N(25)	Input buckling modifier in second transverse direction
M	I*4	--	Data set reference number of SP.CRIT
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/CTYP02/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag 0 - data set edit is suppressed 1 - data set edit is printed

/CTYP03/

Variables used from this common block

Name	Type	Definition
IGTYPE	I*4	Geometry type

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 25 card input; also region index on output modifier DMODB
J	I*4	Index on number of finite directions; also NIL index on output DMODB
L	I*4	Index on regions
NIL	I*4	Number of finite directions or integration limits
NREG	I*4	Number of regions
BLANKS	R*4	Alphameric blank data

4. Functions and tasks performed by this subroutine

Subroutine BUCSRH defines and writes the second record of SP.CRIT when buckling modifiers are input on type 25 cards.

By checking the second input modifiers for zeroes, it is determined whether the input is consistent with the geometry type. If the second direction modifiers are not all zero, NIL is set equal to 2 and the geometry type must be finite in two directions. If not, a fatal error is noted.

If the input is valid, DMODB is initialized to zero.

Input region labels are compared with all region labels.

If

SLABLR(I) = blanks,

DMOD1(L,2) = DMOD1(I)

and

DMOD2(L,2) = DMOD2(I)

for each region.

If

SLABLR(I) = REG(L),

DMOD1(L,1) = BMOD1(I)

and

DMOD2(L,2) = BMOD2(I).

If

NIL = 1,

the DMOD2 array is ignored.

5. Subprograms called by this subroutine

DUMP (entry in POINTR)

6. Subprograms calling this subroutine

FSPCRT

7. Error messages generated by this subroutine

Fatal:

(BUCSRH) - NUMBER OF INTEGRATION LIMITS IS NOT CONSISTENT WITH GEOMETRY TYPE.

Subroutine CDSCAN (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	Data set reference number for A.NIP
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/IERR/

Variables used from this common block

Name	Type	Definition
IQUIT	I*4	Execution stop flag 0 - execution of module successful 1 - execution of module terminated

/LOCATE/

Variables used from this common block

Name	Type	Definition
IPRINT	I*4	Print edit indicator for POINTR

/CTYP02/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag 0 - data set edit is suppressed 1 - data set edit is printed

/CTYP03/

Variables used from this common block

Name	Type	Definition
ALPHA	R*8	Initial alpha or fixed alpha
IGTYPE	I*4	Geometry type
IPTYPE	I*4	Problem type
FIXEDK	R*8	Fixed k-effective
ICTYPE	I*4	Calculation type

/CTYP04/

Variables used from this common block

Name	Type	Definition
IEBC(4,4)	I*4	External boundary condition number

/CTYP05/

Variables used from this common block

Name	Type	Definition
NLX	I*4	Number of input cards for lower 'X' external boundary
NUX	I*4	Number of input cards for upper 'X' external boundary
NLY	I*4	Number of input cards for lower 'Y' external boundary
NUY	I*4	Number of input cards for upper 'Y' external boundary

/CTYP09/

Variables used from this common block

Name	Type	Definition
NVX	I*4	Number of variable-mesh 'X' direction input quantities
NVY	I*4	Number of variable-mesh 'Y' direction input quantities

/CTYP10/

Variables used from this common block

Name	Type	Definition
NIX	I*4	Index number of type 10 'X' direction internal boundary input
NIY	I*4	Index number of type 10 'Y' direction internal boundary input

Name	Type	Definition
NX	I*4	Number of different 'X' direction internal boundaries
NY	I*4	Number of different 'Y' direction internal boundaries

/CTYP21/

Variables used from this common block

Name	Type	Definition
ISSPEC	I*4	Search type specification
MAXNSP	I*4	Maximum number of search passes
PKOA	R*8	Prescribed k-effective or initial alpha
EPSLON	R*8	Convergence criterion, epsilon for k-eff or alpha

/CTYP22/

Variables used from this common block

Name	Type	Definition
SIVX	R*8	Initial value of X, the search parameter
SSVX	R*8	Second value of X
SLBX	R*8	Lower bound for X
SVBX	R*8	Upper bound for X
SDRX	R*8	Derivative of k-eff or alpha with respect to X

/CTYP24/

Variables used from this common block

Name	Type	Definition
NMX	I*4	Number of 'X'-direction mesh modifiers input in criticality search
NMY	I*4	Number of 'Y'-direction mesh modifiers input in criticality search

/CTYP26/

Variables used from this common block

Name	Type	Definition
ALPHAM	R*8	Alpha modifier for criticality search

/CTYP27/

Variables used from this common block

Name	Type	Definition
NORDER	I*4	Anisotropy flag for data set ES.D1SH

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type in A.NIP

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups
NCMP	I*4	Number of compositions
MINFLG	I*4	Flag = 1 if old XS.C.MIN was read

3. Local Variables

Name	Type	Definition
I	I*4	Index for external boundaries

4. Functions and tasks performed by this subroutine

Subroutine CDSCAN is called by subroutine GNIP after it has been determined that A.NIP exists.

The number of cards of each type present is tested for a value greater than zero.

When a card type nn is present, the TYPEnn subroutine is called to read those cards and the data set reference number for A.NIP is passed to the called subroutine, along with an error return number for immediate return to the program calling the module and a second error return for a skip to the next card type.

After all card types have been processed, data set A.NIP is rewound and a call to STATUS is issued to determine and print the container array's condition.

If the variable IQUIT has been set to zero or if at any point in the data-set reading the first immediate error return has been involved, CDSCAN uses its error return to GNIP and MAIN and thence to the calling program, stopping execution.

5. Subprograms called by this subroutine

DUMP (entry in POINTR)	TYPE09	TYPE19
STATUS (entry in POINTR)	TYPE10	TYPE20
TYPE01	TYPE11	TYPE21
TYPE02	TYPE12	TYPE22
TYPE03	TYPE13	TYPE23
TYPE04	TYPE14	TYPE24
TYPE05	TYPE15	TYPE25
TYPE06	TYPE16	TYPE26
TYPE07	TYPE17	TYPE27
TYPE08	TYPE18	TYPE28

6. Subprograms calling this subroutine

GNIP

7. Error messages generated by this subroutine

When search specifications on a type 21 card are given and parameters are supplied by a type 22 card, but no type 23, 24, 25, or 26 cards are present, SP.CICN and SP.CRIT cannot be written. An error message is issued, as follows:

(CDSCAN) - TYPE 21 AND 22 CARDS WERE SUPPLIED IN A.NIP, BUT NO CARDS OF TYPES 23, 24, 25, OR 26 WERE SUPPLIED.

Subroutine COMREG (NC, REG, NUMR, RLABEL, CNAME, NUMC, CRLABL, CREGN, CLABEL, *)

1. Arguments

Name	Type	Dimension	Definition
NC	I*4	N(6)	Composition indices of regions (I)
REG	R*8	NREG	Region names
NUMR	I*4	N(6)	Region number of REG(I)
RLABEL	R*8	N(6)	Region label from card type 6
CNAME	R*8	NCMP	Composition name
NUMC	I*4	N(15)	Composition number of CRLABL(I)
CRLABL	R*8	N(15)	Composition label from card 15, contained in region CREGN
CREGN	R*8	10*N(15)	Region label of region containing specified composition
CLABEL	R*8	N(14)	Composition label from card type 14
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups
NCMP	I*4	Number of compositions
MINFLG	I*4	Flag = 1 if old XS.C.MIN was read
NDUM	I*4	Dummy for alignment
CNAME(199)	R*8	Array of composition names

/GEOM1/

Variables used from this common block

Name	Type	Definition
NREG	I*4	Number of regions
NCMP	I*4	Number of compositions

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 06 cards; also index on type 14 cards
J	I*4	Index on NUMC
K	I*4	Index on composition name being defined; also index on region names
IJ	I*4	Index on CREGN
J1	I*4	Lower limit of IJ, equal to 10*(J-1) + 1 where J = 1, N(15)
J2	I*4	Upper limit of IJ, equal to 10*J where J = 1, N(15)

Name	Type	Definition
K1	I*4	Counter for additional composition names on card 14
NR	I*4	Counter and index on number of regions
N6	I*4	Number of type 06 cards
NCP	I*4	Count and index number of compositions
NR1	I*4	Redefinition of number of regions
N14	I*4	Number of type 14 cards
N15	I*4	Number of type 15 cards
TEMP	R*8	Temporary storage for CNAME(K)
IM1	I*4	Upper index on CLABEL(J)

4. Functions and tasks performed by this subroutine

COMREG counts the number of compositions and number of regions and defines NCMP and NREG for the GEOM data set. In addition, the arrays CNAME (composition names), REG (region names), and NC (region-composition index) are defined for the GEOM data set.

The CRLABL names on type 15 cards are compared with composition labels CLABEL which originate on type 14 cards or on an old XS.C.MIN data set. CNAME is assigned from the labels and duplicates are eliminated:

CNAME(NCP) = CRLABL(I).

Composition numbers, NUMC, are then assigned from the compositions:

for CRLABL(I) = CNAME(K),

NUMC(I) = K.

NCMP is set to the total NCP.

Region-composition correspondence is established from CREGN labels on type 15 cards, and regions are numbered.

All type 15 card region labels CREGN(IJ), where IJ = J + (1, ···, 10), are compared with all REG(NR). Matching labels cause definitions

NC(NR) = NUMC(J) and NUMR(I) = NR.

Finally, NREG is set equal to the total NR.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

CRCDR

7. Error messages generated by this subroutine

Fatal:

(COMREG) - NO REGION LABEL ON TYPE 15 CARDS AGREES WITH REGION LABEL 'aaaaaa' ON TYPE 06 CARDS

Subroutine CONSCH (CLABEL, SLABL1, NCMP, NN, NMC, NCOMPN, SLABL2, N5, CNAME, M)

1. Arguments

Name	Type	Dimension	Definition
CLABEL	R*8	N(14) or NCMP	Composition label from card type 14 or data set XS.C.MIN
SLABL1	R*8	N(23)	Composition label of modifier composition
NCMP	I*4	--	Number of compositions defined
NN	I*4	--	Number of type 23 cards
NMC	I*4	NCMP	Modifier composition for the Ith composition
NCOMPN	I*4	N(23)	Modifier composition numbers
SLABL2	R*8	10 * N(23)	Composition labels of compositions with which modifier is used
N5	I*4	--	Number of type 14 cards
CNAME	R*8	--	Composition name
M	I*4	--	SP.CRIT data set reference number

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(I)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag- 0 - data set edit is suppressed 1 - data set edit is printed

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 14 cards also index on number of compositions
J	I*4	Index on type 23 cards
L	I*4	Increment to basic data index on a given card
JJ	I*4	Base index of data on card, equal to 10 * J-10

4. Functions and tasks performed by this subroutine

CONSCH writes the second record of the SP.CRIT data set when a composition search is to be performed.

The working array MCOMPN of input modifier numbers and the array NMC are initialized to zero. NMC(I) is the modifier composition for the Ith composition.

The input SLABL1(J) is compared with CLABEL(I) until a matching label is found. The composition index number MCOMPN is found by comparing all CNAME(L) with that CLABEL(I). When like labels are found, MCOMPN(J) is set to L.

The SLABL2 names to be modified are compared with CNAME(I) and when a label matches the modifier label, the modifier NMC of composition (I) is set to MCOMPN(J).

The array NMC is then written on data set SP.CRIT and return is made to FSPCRT.

5. Subprograms called by this subroutine

DUMP (entry in POINTR)

6. Subprograms calling this subroutine

FSPCRT

7. Error messages generated by this subroutine - None

Subroutine CRCDR (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(I)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

3. Local Variables - None

4. Functions and tasks performed by this subroutine

FGEOM calls CRCDR to establish region names and composition names and count them.

The arrays NC, REG, NUMR, NUMC, and CNAME are formed and their pointers are passed to subroutine COMREG along with the pointers to RLABEL, CLABEL, CRLABL, and CREGN.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
COMREG
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FGEOM

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - SUBROUTINE CRCDR, N2 = 'ii'

Subroutine CRITSH (DIMOD, DJMOD, XMODLC, YMODLC, XMESHM, YMESHM, XMODUC, YMODUC, NINTI, NINTJ, NMX, NMY, M, XMESH, YMESH, *)

1. Arguments

Name	Type	Dimensions	Definition
DIMOD	R*8	NINTI	Output 'X' mesh modifier for criticality search
DJMOD	R*8	NINTJ	Output 'Y' mesh modifier for criticality search
XMODLC	R*8	NMX	'X' direction lower coordinate of mesh interval range being modified
YMODLC	R*8	NMY	'Y' direction lower coordinate of mesh interval range being modified
XMESHM	R*8	NMX	Dimensionless mesh modifier for the given 'X' range
YMESHM	R*8	NMY	Dimensionless mesh modifier for the given 'Y' range
XMODUC	R*8	NMX	'X' direction upper coordinate of mesh interval range being modified
YMODUC	R*8	NMY	'Y' direction upper coordinate of mesh interval range being modified
NINTI	I*4	--	Number of intervals in 'X' direction

Name	Type	Dimensions	Definition
NINTJ	I*4	--	Number of intervals in 'Y' direction
NMX	I*4	--	Number of 'X' direction input modifiers in criticality search
NMY	I*4	--	Number of 'Y' direction mesh modifiers in criticality search
M	I*4	--	Data set reference number for SP.CRIT
XMESH	R*8	NPTI	'X' direction mesh coordinates
YMESH	R*8	NPTJ	'Y' direction mesh coordinates
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag - 0 - data set edit is suppressed 1 - data set edit is printed

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of X modifiers
J	I*4	Index on number of Y modifiers
K	I*4	Lower limit on index for outputing mesh modifiers

Name	Type	Definition
L	I*4	Index on input modifiers and coordinates
N	I*4	Upper limit on index for outputing mesh modifiers
D1	R*8	Absolute value of lower input coordinate minus mesh coordinate
D2	R*8	Absolute value of upper input coordinate minus mesh coordinate
LL	I*4	Index on mesh points
NPTI	I*4	Number of points in 'X' direction
NPTJ	I*4	Number of points in 'Y' direction

4. Functions and tasks performed by this subroutine

Subroutine CRITSH defines the mesh modifiers for a criticality search. Arrays DIMOD and DJMOD are initialized to zero.

Then for each mesh direction, the input lower and upper coordinates are compared with the actual mesh coordinates to see whether they are valid limits. If not, a specific message is printed and the error return is used.

The coordinates are valid for each X direction input card L

if

$$|XMODLC(L)-XMESH(LL)| \leq 0.001,\ K = LL,$$

and if

$$|XMODUC(L)-XMESH(LL)| \leq 0.001,\ N = LL-1,$$

where

$$LL = 1, \cdots, NPTI.$$

DIMOD(I) is defined as XMESHM(L) over the coordinate range specified on the type 24 card.

The same procedure is followed for the YMODLC and YMODUC arrays, and DJMOD(I) is defined as YMESH(L) over the specified range.

A record consisting of NINTI, NINTJ and DIMOD(I) for all intervals I is written on data set SP. CRIT.

5. Subprograms called by this subroutine

DUMP (entry in POINTR)

6. Subprograms calling this subroutine

FSPCRT

7. Error messages generated by this subroutine

Fatal:

(CRITSH) - THERE IS NO MESH POINT CORRESPONDING TO THE 'ii' CARD (TYPE 24)

Subroutine DSGEM (XMESH, YMESH, MR, MC, NC, NREGA, NRA, NSECTZ, NSB, BSQ, H, X, REG, CNAME, AREA, ZONE, REGVOL, NI, NR, N)

1. Arguments

Name	Type	Dimension	Definition
XMESH	R*8	NPTI	'X' direction mesh coordinate
YMESH	R*8	NPTJ	'Y' direction mesh coordinate
MR	I*4	NI*NINTJ	Region index of mesh
MC	I*4	NI*NINTJ	Composition index of mesh
NC	I*4	NREG	Composition index of region
NREGA	I*4	NAREA	Number of regions in each area
NRA	I*4	LREGA	List of region indices for areas
NSECTZ	I*4	NZONE	Number of sections in each zone
NSB	I*4	LSECT	List of section boundary points
BSQ	R*8	NREG*NIL	Buckling by regions
H	R*8	NREG*NIL	Actual half-height for region
X	R*8	NREG*NIL	Extrapolated half height
REG	R*8	NREG	Region labels
CNAME	R*8	NCMP	Composition labels
AREA	R*8	NAREA	Area labels
ZONE	R*8	NZONE	Zone labels

Name	Type	Dimension	Definition
REGVOL	R*8	NR	Region volumes
NI	I*4	--	Number of 'X' direction intervals
NR	I*4	--	Number of regions
N	I*4	--	Data set reference number of GEOM data set

2. Common Blocks

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions in geometry
NGEOM	I*4	Type of geometry
NPTI	I*4	Number of lines in 'X' direction
NPTJ	I*4	Number of lines in 'Y' direction
NINTI	I*4	Number of intervals in 'X' direction
NINTJ	I*4	Number of intervals in 'Y' direction
NREG	I*4	Number of regions
NCMP	I*4	Number of compositions
NAREA	I*4	Number of areas
LAREA	I*4	Length of record type 6 in GEOM
NZONE	I*4	Number of zones
LZONE	I*4	Length of record type 7 in GEOM
NFIN	I*4	Number of finite geometry type
NIL	I*4	Number of finite directions (integration limits)
LREGA	I*4	Length of vector NRA
LSECT	I*4	Length of vector NSB

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag - 0 - data set edit is suppressed 1 - data set edit is printed

3. Local Variables

Name	Type	Definition
I	I*4	Index variously used as 1 to NINTI, NPTI, NREG, LREGA, NZONE, LSECT, NCMP, or NAREA; also a counter for the number of mesh lines printed
J	I*4	Index variously used as 1 to NPTJ, NINTJ, or NIL
K	I*4	Index on quantities printed per output line
II	I*4	Upper limit of printing index numbers
NJ	I*4	'Y' index of printed quantity
NP	I*4	'X' index of last quantity printed on line

4. Functions and tasks performed by this subroutine

Subroutine DSGEM writes the GEOM data set as it is specified. If desired, the data are also printed, along with a reactor profile map which gives all mesh points and the region number and composition number for every interval.

5. Subprograms called by this subroutine

REC10

6. Subprograms calling this subroutine

FGEOM

7. Error messages generated by this subroutine - None

Subroutine DZONE (ZONE, NSECT, NSB, ZLABEL, ZXL, ZXU, ZYL, ZYU, XMESH, YMESH, *)

1. Arguments

Name	Type	Dimension	Definition
ZONE	R*8	N(8)	Output zone label
NSECT	I*4	N(8)	Number of sections in each zone
NSB	I*4	2*NDIM*N(8)	List of section boundary mesh index numbers
ZLABEL	R*8	N(8)	Zone definition label
ZXL	R*8	N(8)	Lower 'X' boundary of section
ZXU	R*8	N(8)	Upper 'X' boundary of section
ZYL	R*8	N(8)	Lower 'Y' boundary of section
ZYU	R*8	N(8)	Upper 'Y' boundary of section
XMESH	R*8	NPTI	'X' direction mesh coordinates
YMESH	R*8	NPTJ	'Y' direction mesh coordinates
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions in geometry
NPTI	I*4	Number of 'X' direction lines
NPTJ	I*4	Number of 'Y' direction lines

Name	Type	Definition
NZONE	I*4	Number of zones
LZONE	I*4	Length of record type 7 in GEOM; also NZONE + LSECT + 1
LSECT	I*4	Length of vector NSB

3. Local Variables

Name	Type	Definition
I	I*4	Index of zone labels Index on type 08 card input
J	I*4	Index on number of 'Y' lines
K	I*4	Index on number of zones
I1	I*4	Index on number of 'X' lines
I2	I*4	Index on number of 'X' lines
J2	I*4	Index on number of 'Y' lines
N8	I*4	Number of type 08 cards

4. Functions and tasks performed by this subroutine

DZONE defines and counts the zones described on type 08 cards of A.NIP for the GEOM data set. Also defined are NZONE, LZONE, and LSECT for the common block GEOM1. ZONE(1) is set to ZLABEL(1) and NZONE = 1.

For each new ZLABEL(I), NZONE = NZONE + 1 and ZONE(NZONE) = ZLABEL(I). After all labels have been classified, LSECT and NSECT(K) are initialized to zero.

For every ZLABEL(I) = ZONE(K),

NSECT(K) = NSECT(K) + 1

and

LSECT = LSECT + 2.

The input coordinates ZYL and ZXU are compared with the XMESH structure to find the 'X' indices of those coordinates which define the section, I1 and I2.

Then NSB(LSECT-1) = I1,

NSB(LSECT) = I2,

and

LSECT = LSECT + 2.

The 'Y' direction input is handled the same way to define

NSB(LSECT-1) = J

and

NSB(LSECT) = J2.

If the input coordinates do not coincide with some mesh line, a specific message is printed and the error return is used.

Finally, LZONE, the length of record 7 in GEOM, is defined as NZONE + LSECT + 1.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

ZONEDR

7. Error messages generated by this subroutine

Fatal:

(DZONE) - COORDINATE IN ZONE 'aaaaaa' DOES NOT CORRESPOND TO ANY MESH LINE.

Subroutine EXTBND (NBCL, NBCR, NBCB, NBCT, NVB, NBC, MFACE, M1, M2, NBIT, JI, JF, C1, C2, C3, C4, EBCA, EBCB, RLCOB, RUCOB, BD, XMESH, YMESH, IHEGN, ILEGN, *)

1. Arguments

Name	Type	Dimension	Definition
NBCL	I*4	--	Left boundary condition type number
NBCR	I*4	--	Right boundary condition type number
NBCB	I*4	--	Bottom boundary condition type number
NBCT	I*4	--	Top boundary condition type number
NVB	I*4	--	Number of vertical boundary records

Name	Type	Dimension	Definition
NBC	I*4	--	Total number of boundary condition records
MFACE	I*4	NBC	Output mesh line number and side indicator
M1	I*4	NBC	Output line number defining beginning of interval along face
M2	I*4	NBC	Output line number defining end of interval along face
NBIT	I*4	NBC	Output type number of boundary or interface condition
J1	I*4	NBC	Output initial higher energy group number to which set of constants applies
JF	I*4	NBC	Output final lower energy group number to which set of constants applies
C1	R*8	NBC	First output constant
C2	R*8	NBC	Second output constant
C3	R*8	NBC	Third output constant
C4	R*8	NBC	Fourth output constant
EBCA	R*8	N(5)	Input first constant to external boundary
EBCB	R*8	N(5)	Input second constant to external boundary
RLCOB	R*8	N(5)	Input lower end coordinate on external boundary
RUCOB	R*8	N(5)	Input upper end coordinate on external boundary
BD	R*8	N(5)	External boundary names
XMESH	R*8	NPTI	'X' direction mesh coordinates
YMESH	R*8	NPTJ	'Y' direction mesh coordinates
IHEGN	I*4	N(5)	Input higher energy group number to which constants apply

Name	Type	Dimension	Definition
ILEGN	I*4	N(5)	Input lower energy group number to which constants apply
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NPTI	I*4	Number of 'X' direction mesh lines
NPTJ	I*4	Number of 'Y' direction mesh lines

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag - 0 - data set edit is suppressed 1 - data set edit is printed

/CTYPO4/

Variables used from this common block

Name	Type	Definition
IEBC(4,4)	I*4	External boundary condition number
NIEBC	I*4	Number of external boundary condition numbers given

/CTYPO5/

Variables used from this common block

Name	Type	Definition
NLX	I*4	Number of lower 'X' direction external boundary input cards
NUX	I*4	Number of upper 'X' direction external boundary input cards
NLY	I*4	Number of lower 'Y' direction external boundary input cards
NUY	I*4	Number of upper 'Y' direction external boundary input cards

/CTYP10/

Variables used from this common block

Name	Type	Definition
NIX	I*4	Number of type 10 'X' direction internal boundary input cards
NIY	I*4	Number of type 10 'Y' direction internal boundary input cards
NX	I*4	Number of type 11 'X' direction input cards
NY	I*4	Number of type 11 'Y' direction input cards

3. Local Variables

Name	Type	Definition
B	R*8	EBCB(J), the second input constant
I	I*4	Index on IEBC on card type 04
J	I*4	Index on type 05 card input
K	I*4	IEBC(4,I)
L	I*4	Index on output quantities
AA	R*8	EBCA(J), the first input constant
L1	I*4	Counter for left external boundary output
L2	I*4	Counter for right external boundary output
L3	I*4	Counter for bottom external boundary input
L4	I*4	Counter for top external boundary output
N5	I*4	Number of type 05 cards in A.NIP
XL	R*8	Data name of lower 'X' boundary
XU	R*8	Data name of upper 'X' boundary
YL	R*8	Data name of lower 'Y' boundary
YU	R*8	Data name of upper 'Y' boundary
NEED	I*4	Flag indicating need for type 05 cards in A.NIP

4. Functions and tasks performed by this subroutine

Subroutine EXTBND structures the output arrays used in writing data set BC. The external boundary input quantities are used to fill the array positions which are associated with external boundary output records.

The need for type 05 card input is first established. If constants are required, type 05 cards must exist in A.NIP.

The subscript (L) for output is defined and MFACE(L) and NBIT(L) are defined.

For XL, L = 1 + L1, MFACE(L) = 1, and NBIT(L) = NBCL.

For XU, L = 1 + L2 + NLX + NY * 2, MFACE(L) = NPTI, and NBIT(L) = NBCR. If NLX = 0, L = 2 + L2 + NY * 2.

For YL, L = NVB + 1 + L3, MFACE(L) = 1, and NBIT(L) = NBCL.

For YU, L = NVB + NLY + L4 + 1 + NX * 2, MFACE(L) = -NPTJ, and NBIT(L) = NBCT. If NLY = 0, L = 2 + NVB + L4 + NX *2.

If no type 05 cards exist, for vertical boundaries M1(L) = 1 and M2(L) = NPTJ and for horizontal boundaries M1(L) = 1 and M2(L) = NPTI.

Otherwise, TEST is called to define M1(L) and M2(L) after checking RLCOB(J) and RUCOB(J) against the mesh structure.

If type 05 cards exist, AA = EBCA(J) and B = EBCB(J). For different values of K, the boundary condition type number, the following constants are defined.

For K = 1, C1(L) = 0, C2(L) = 1, and C3(L) = AA.

For K = 2, C1(L) = 0, C2(L) = 1, and C3(L) = 0.

For K = 3, C1(L) = 1, C2(L) = 0, and C3(L) = 0.

For K = 4, C1(L) = 1, C2(L) = AA, and C3(L) = 0.

For K = 5, C1(L) = 1, C2(L) = AA, and C3(L) = 0.

For K = 6, an error message is printed.

For K = 7, C1(L) = AA, C2(L) = B, and C3(L) = 1.

For K = 8, C1(L) = AA, C2(L) = B, and C3(L) = 1;

and in all cases, C4(L) = 0.

These constants apply to the groups JI(L) to JF(L), inclusive.

When IHEGN(J) = 0, JI(L) = 1 and JF(L) = NGROUP.

If ILEGN(J) = 0, JI(L) = IHEGN(J) and JF(L) = IHEGN(J). Otherwise, JI(L) = IHEGN(J) and JF(L) = ILEGN(J).

5. Subprograms called by this subroutine

DUMP (entry in POINTR)
TEST

6. Subprograms calling this subroutine

FBC

7. Error messages generated by this subroutine

Fatal:

(EXTBND) - TYPE 6 BOUNDARY CONDITION NOT IMPLEMENTED.
(EXTBND) - INVALID BOUNDARY CONDITION TYPE NUMBER 'ii' GIVEN ON TYPE 04 CARD. WHEN NO TYPE 05 CARDS ARE PRESENT, TYPE MUST BE 2, 3, 6, 9, 10, 11, OR 12.

Subroutine FACEN (ZMSHA, ZMSHB, I, J, NPTA, NPTB, DNCORD, UPCORD, ONCORD, LO, IHI, ION, *)

1. Arguments

Name	Type	Dimension	Definition
ZMSHA	R*8	NPTA	'A' direction mesh coordinate array
ZMSHB	R*8	NPTB	'B' direction mesh coordinate array
I	I*4	--	Index of type 28 card array
J	I*4	--	Index of type 27 card array
NPTA	I*4	--	Number of lines in 'A' direction
NPTB	I*4	--	Number of lines in 'B' direction
DNCORD	R*8	N(28)	Input lower coordinate of source face
UPCORD	R*8	N(28)	Input upper coordinate of source face
ONCORD	R*8	N(28)	Coordinate along which source face lies
LO	I*4	--	Index of lower mesh line defining source face
IHI	I*4	--	Index of upper mesh line defining source face
ION	I*4	--	Index of line along which source face lies
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

3. Local Variables

Name	Type	Definition
K	I*4	Index on 'B' direction mesh line coordinates
N	I*4	Index on 'A' direction mesh line coordinates

4. Functions and tasks performed by this subroutine

Subroutine FACEN validates the input coordinate limits and defines the indices for the output source arrays. It is called once for each TYPE27 input card. The input coordinates, DNCORD and UPCORD, on the type 27 card are checked against the appropriate mesh coordinate array, and when points coincide the point index is used to define the lower and upper limits of a given source. The second direction coordinate array is checked against ONCORD to see that the line really coincides with a mesh coordinate of the system. Thus, LO, IHI and ION are returned to the calling subroutine to define the source limits for a given input source.

The following default options are observed under the conditions listed below.

If DNCORD is -1, the source limits are defined along the entire mesh line.

If UPCORD is -1, the source limits are defined along the single mesh interval specified by DNCORD.

If ONCORD is -1, ION equals the maximum coordinate index of the mesh structure.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

ANISO1
ANISO2

7. Error messages generated by this subroutine

Fatal:

(FACEN) - THE INPUT IN COLS. 20-36 ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH THE MESH STRUCTURE.

(FACEN) - THE INPUT IN COLS. 37-48 ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH THE MESH STRUCTURE.

(FACEN) - THE INPUT IN COLS. 49-60 ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH THE MESH STRUCTURE.

Subroutine FBC (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	BC data set reference number
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type in A.NIP

/CTYPO4/

Variables used from this common block

Name	Type	Definition
IEBC(4,4)	I*4	External boundary condition number
NIEBC	I*4	Number of external boundary condition numbers given

/CTYPO5/

Variables used from this common block

Name	Type	Definition
NLX	I*4	Number of lower 'X' boundary input cards

Name	Type	Definition
NUX	I*4	Number of upper 'X' boundary input cards
NLY	I*4	Number of lower 'Y' boundary input cards
NUY	I*4	Number of upper 'Y' boundary input cards

/CTYP10/

Variables used from this common block

Name	Type	Definition
NIX	I*4	Number of type 10 card 'X' direction internal interfaces input
NIY	I*4	Number of type 10 card 'Y' direction internal interfaces input
NX	I*4	Number of type 11 'X' direction input cards
NY	I*4	Number of type 11 'Y' direction input cards

/GEOM1/

Variables used from this common block

Name	Type	Definition
NPTI	I*4	Number of 'X' direction points
NPTJ	I*4	Number of 'Y' direction points

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

3. Local Variables

Name	Type	Definition
I	I*4	Index on IEBC
M	I*4	Index of next array position to be filled by subroutine

Name	Type	Definition
IB	I*4	Representation of IEBC(4,I)
IT	I*4	Boundary condition type number
NN	I*4	NIEBC, number of quantities on card type 04
N4	I*4	NIEBC
N5	I*4	Number of type 05 cards in A.NIP
IBD	I*4	Pointer to array BD
IC1	I*4	Pointer to array C1
IC2	I*4	Pointer to array C2
IC3	I*4	Pointer to array C3
IC4	I*4	Pointer to array C4
IJF	I*4	Pointer to array JF
IJI	I*4	Pointer to array JI
IM1	I*4	Pointer to array M1
IM2	I*4	Pointer to array M2
NBC	I*4	Total number of boundary condition records
BD	I*4	Input external boundary name
NHB	I*4	Total number of horizontal boundary condition records
NVB	I*4	Number of vertical boundary condition records
N10	I*4	Number of type 10 cards in A.NIP
N11	I*4	Number of type 11 cards in A.NIP
NBCB	I*4	Bottom boundary condition type number
NBCL	I*4	Left boundary condition type number
NBCR	I*4	Right boundary condition type number
NBCT	I*4	Top boundary condition type number

Name	Type	Definition
DUMMY	R*8	Dummy
IBCA1	I*4	Pointer to array BCA1
IBCA2	I*4	Pointer to array BCA2
IBCB1	I*4	Pointer to array BCB1
IBCB2	I*4	Pointer to array BCB2
IEBCA	I*4	Pointer to array EBCA
IEBCB	I*4	Pointer to array EBCB
IFLAG	I*4	Count of dimensions
INBIT	I*4	Pointer to array NBIT
IXLEI	I*4	Pointer to array XLEI
IXUEI	I*4	Pointer to array XUEI
IXYIC	I*4	Pointer to array XYIC
IYLEI	I*4	Pointer to array YLEI
IYUEI	I*4	Pointer to array YUEI
IYXIC	I*4	Pointer to array YXIC
IICONX	I*4	Pointer to array ICONX
IICONY	I*4	Pointer to array ICONY
IIHBGN	I*4	Pointer to array IBHEGN
IIHEGN	I*4	Pointer to array IHEGN
IILBGN	I*4	Pointer to array IBLEGN
IILEGN	I*4	Pointer to array ILEGN
ILABEL	I*4	Pointer to array ULABEL
IMFACE	I*4	Pointer to array MFACE
IRLCOB	I*4	Pointer to array RLCOB
IRUCOB	I*4	Pointer to array RUCOB
IXMESH	I*4	Pointer to array XMESH

Name	Type	Definition
IYMESH	I*4	Pointer to array YMESH
JLABEL	I*4	Pointer to array XILABL
KLABEL	I*4	Pointer to array YILABL

4. Functions and tasks performed by this subroutine

FBC controls the formation of the boundary condition data set BC.

All counting, indexing and array allocation is handled in this routine, and parameters are passed to separate external and internal data-handling subroutines.

The type 04 card is first analyzed and NBCL, NBCR, NBCB, and NBCT are defined.

If the type numbers indicate a transport theory problem, return is made to FORMDS.

For a diffusion problem, the number of vertical boundary records to be written in specified as NVB = 2 + (NLX-1) + (NUX-1) + (2*NX), where NLX is the number of type 05 lower 'X' boundary input cards, NUX is the number of type 05 upper 'X' boundary input cards, and NX is the number of different vertical internal interface input specifications on type 11 cards.

If the problem is two-dimensional, NVB = 2 + (NLY-1) + (NLY-1) + (2*NY), where NLY is the number of type 05 lower 'Y' boundary input cards, NUY is the number of type 05 upper 'Y' boundary input cards, and NY is the number of different horizontal internal interface input specifications on type 11 cards.

After all boundaries are counted, subroutine EXTBND is called to define the output array contents for the external boundaries.

Subroutine INTBND is called to define the horizontal and vertical internal interface array contents.

In each case the index of the first array position to be filled in the called subroutine is computed in FBC and passed to the routine.

After all data have been defined, subroutine BCRITE is called to write the NBC boundary constant records on data set BC.

Arrays which are no longer needed are wiped out after the various subroutine calls.

5. Subprograms called by this subroutine

BCRITE
EXTBND
GETPNT (entry in POINTR)
INTBND
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - IN FBC, N2 = 'ii'

(FBC) - THERE ARE NO BOUNDARY NUMBERS GIVEN - TYPE 04 CARD MAY NOT BE CORRECT.

Subroutine FESD1D (N1, *)

1. Arguments

Name	Type	Definition
N1	I*4	A.NIP data set reference number
*	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	R*8	Array of number of cards of each type in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of 'X' direction intervals
NINTJ	I*4	Number of 'Y' direction intervals

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

3. Local Variables

Name	Type	Definition
IMC	I*4	Pointer to array MC
IMR	I*4	Pointer to array MR
NIL	I*4	Number of integration limits
INRA	I*4	Pointer to array NRA
INSB	I*4	Pointer to array NSB
DUMMY	R*8	Dummy
ICODE	I*4	Pointer to array IICODE
IFSEXT	I*4	Pointer to array FSEXT
IIFLAG	I*4	Pointer to array IFLAG
IIHEGN	I*4	Pointer to array ISHEGN
IILEGN	I*4	Pointer to array ISLEGN
IISPEC	I*4	Pointer to array ISPECN
ILABEL	I*4	Pointer to array SILABL
INREGA	I*4	Pointer to array NREGA
INSECT	I*4	Pointer to array NSECTZ
IRLISC	I*4	Pointer to array RLISC
IRUISC	I*4	Pointer to array RUISC
ISOURC	I*4	Pointer to array SOURCE
IXMESH	I*4	Pointer to array XMESH
IYMESH	I*4	Pointer to array YMESH
JLABEL	I*4	Pointer to array REG
KLABEL	I*4	Pointer to array AREA
LLABEL	I*4	Pointer to array ZONE

4. Functions and tasks performed by this subroutine

FESD1D defines the arrays needed for preparation of ES.D1D, the one-dimensional distributed isotropic inhomogeneous source output data set. After a PUTPNT request for storage of arrays IFLG and FSEXT, the pointers are extracted for arrays needed. These are passed to subroutine I1SORD.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
I1SORD
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - IN FESD1D, N2 = 'ii'

Subroutine FESD1H (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	ES.D1SH data set reference number
*	---	--	Error return

2. Common blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NPTJ	I*4	Number of lines in 'Y' direction
NINTI	I*4	Number of intervals in 'X' direction

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

/CTYP27/

Variables used from this common block

Name	Type	Definition
NORDER	I*4	Anisotropy flag for data set ES.D1SH

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

3. Local Variables

Name	Type	Definition
N2	I*4	Number of array words requested
DUMMY	R*8	Dummy
ICORD	I*4	Pointer to array CORDIR

Name	Type	Definition
IALABL	I*4	Pointer to array SALABL
ICSORS	I*4	Pointer to array CISORS
IDNCOR	I*4	Pointer to array DNCORD
IVSUR1	I*4	Pointer to array VSUR1
IVSUR2	I*4	Pointer to array VSUR2
IVSUR3	I*4	Pointer to array VSUR3
IIHEGN	I*4	Pointer to array IAHEGN
IIVERT	I*4	Pointer to array IVERT
IILEGN	I*4	Pointer to array IALEGN
ICSORS	I*4	Pointer to array CISORS
IONCOR	I*4	Pointer to array ONCORD
IUPCOR	I*4	Pointer to array UPCORD
ISLABL	I*4	Pointer to array SSLABL
IXMESH	I*4	Pointer to array XMESH
IXSORS	I*4	Pointer to array AXSORS
IYSORS	I*4	Pointer to array AYSORS
IYMESH	I*4	Pointer to array YMESH

4. Functions and tasks performed by this subroutine

Subroutine FESD1H sets up the arrays IVERT, VSUR1, VSUR2, and VSUR3 in the large container array. It extracts the pointers for arrays needed in defining the one-dimensional surface source data for data set ES.D1SH and passes them to subroutine ANISO1. Unneeded arrays are wiped out after the call to ANISO1.

5. Subprograms called by this subroutine

ANISO1
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

(FESD1H) - ARRAY OVERFLOW, N2 = 'ii'

Subroutine FESD2D (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of 'X' direction intervals
NINTJ	I*4	Number of 'Y' direction intervals

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

3. Local Variables

Name	Type	Definition
IMC	I*4	Pointer to array MC
IMR	I*4	Pointer to array MR
INRA	I*4	Pointer to array NRA
INSB	I*4	Pointer to array NSB
DUMMY	R*8	Dummy
ICODE	I*4	Pointer to array IICODE
IFSEXT	I*4	Pointer to array FSEXT
IIFLAG	I*4	Pointer to array IFLAG
IIHEGN	I*4	Pointer to array IHEGN
IILEGN	I*4	Pointer to array ILEGN
IISPEC	I*4	Pointer to array ISPECN
ILABEL	I*4	Pointer to array SILABL
ING1	I*4	Pointer to array NG1
ING2	I*4	Pointer to array NG2
INREGA	I*4	Pointer to array NREGA

Name	Type	Definition
INSECT	I*4	Pointer to array NSECTZ
IRLISC	I*4	Pointer to array RLISC
IRUISC	I*4	Pointer to array RUISC
ISOURC	I*4	Pointer to array SOURCE
IXMESH	I*4	Pointer to array XMESH
IYMESH	I*4	Pointer to array YMESH
JLABEL	I*4	Pointer to array REG
LKABEL	I*4	Pointer to array AREA
LLABEL	I*4	Pointer to array ZONE

4. Functions and tasks performed by this subroutine

FESD2D requests storage and pointers for the arrays needed in preparing the two-dimensional distributed isotropic inhomogenous source data set ES.D2D. It sets aside storage for the arrays IFLAG and FSEXT in the container array. The pointers to arrays needed are extracted and used in calls to subroutine GRPSOR, which defines group limits for the source data, and to subroutine I2SORD, where the sources are defined for output.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
GRPSOR
I2SORD (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - IN FESD2D, N2 = 'ii'

Subroutine FESD2H (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	ES.D2SH data set reference number
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NPTI	I*4	Number of 'X' direction lines
NPTJ	I*4	Number of 'Y' direction lines
NINTI	I*4	Number of 'X' direction intervals
NINTJ	I*4	Number of 'Y' direction intervals

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/TYPE27/

Variables used from this common block

Name	Type	Definition
NORDER	I*4	Anisotropy flag for data set ES.D2SH

3. Local Variables

Name	Type	Definition
N2	I*4	Amount of array space requested in double words
DUMMY	R*8	Dummy
ICORD	I*4	Pointer to array CORDIR
BLANKS	R*8	Alphameric blanks
IALABL	I*4	Pointer to array SALABL
ICSORS	I*4	Pointer to array CISORS
IDNCOR	I*4	Pointer to array DNCORD
IHSUR1	I*4	Pointer to array HSUR1
IHSUR2	I*4	Pointer to array HSUR2
IHSUR3	I*4	Pointer to array HSUR3
IIHEGN	I*4	Pointer to array IAHEGN
IIHORZ	I*4	Pointer to array IHORZ
IILEGN	I*4	Pointer to array IAHEGN
IIVERT	I*4	Pointer to array IVERT
IONCOR	I*4	Pointer to array ONCORD
ISLABL	I*4	Pointer to array SSLABL

Name	Type	Definition
IUPCOR	I*4	Pointer to array UPCORD
IVSUR1	I*4	Pointer to array VSUR1
IVSUR2	I*4	Pointer to array VSUR2
IVSUR3	I*4	Pointer to array VSUR3
IXMESH	I*4	Pointer to array XMESH
IXSORS	I*4	Pointer to array AXSORS
IYMESH	I*4	Pointer to array YMESH
IYSORS	I*4	Pointer to array AYSORS

4. Functions and tasks performed by this subroutine

FESD2H handles the allocation for arrays used to write the two-dimensional anisotropic surface source data set ES.D2SH.

Arrays IHORZ, IVERT, HSUR1, HSUR2,HSUR3, VSUR1, VSUR2, and VSUR3 are allotted space by PUTPNT calls. The pointers to these arrays are extracted along with pointers to other arrays needed. These are passed to subroutine ANISO2. All arrays except XMESH and YMESH are wiped out after normal return from ANISO2.

5. Subprograms called by this subroutine

ANISO2
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

(FESD2H) - ARRAY OVERFLOW, N2 - 'ii'

Subroutine FFAD1 (NFA, *)

1. Arguments

Name	Type	Dimension	Definition
NFA	I*4	--	Data set reference number
*	--	--	Error return

2. Common Blocks

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of 'X' direction intervals

/XSCM/

Name	Type	Definition
NGROUP	I*4	Number of energy groups

3. Local Variables

Name	Type	Definition
I	I*4	Index of N
N	I*4	NINTI * NGROUP
FAN1	R*8	One-dimensional adjoint flux
ITER	I*4	Iteration number
KIND	I*4	Indication of source of flux values
EVALUE	R*8	Eigenvalue for this set of fluxes

4. Functions and tasks performed by this subroutine

This subroutine generates an initial guess adjoint flux data set FA.D1.

The first record is defined and written including

NINTI from common block /GEOM1/,

NGROUP from common block /XSCM/,

EVALUE set to 1.0,

KIND set to -1, and

ITER set to 0.

After defining N as NINTI * NGROUP and FAN1 as 1.0, a second record containing N FAN1's is written.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FFAD1A (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	One-dimensional adjoint angular flux data set reference number
*	--	--	Error return

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This dummy subroutine may be used in the future to generate a one-dimensional adjoint angular flux data set FA.D1A when there is no old one available.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FFAD2 (NFA, *)

1. Arguments

Name	Type	Dimension	Definition
NFA	I*4	--	Two-dimensional adjoint flux data set reference number
*	--	--	Error return

2. Common Blocks

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of 'X' direction intervals
NINTJ	I*4	Number of 'Y' direction intervals

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

3. Local Variables

Name	Type	Definition
I	I*4	Index of NIJ
N	I*4	Index of group number
NIJ	I*4	NINTI * NINTJ
FAN2	R*8	Two-dimensional adjoint flux
ITER	I*4	Iteration number
KIND	I*4	Source of flux set
EVALUE	R*8	Eigenvalue for this flux set

4. Functions and tasks performed by this subroutine

FFAD2 is the two-dimensional counterpart of FFAD1. It generates a set of two-dimensional adjoint fluxes for data set FA.D2. A first record is written containing the following:

NINTI from common block /GEOM1/,
NINTJ from common block /GEOM1/,
NGROUP from common block /XSCM/,
EVALUE set to 1.0,
KIND set to -1, and
ITER set to 0.
NIJ is defined as NINTI*NINTJ and FAN2 as 1.0.

NGROUP records containing NIJ each of FAN2 are then written on data set FA.D2

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FFAD2A (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	Two-dimensional adjoint angular flux data set reference number
*	--	--	Error return

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This is a dummy subroutine which may be used in the future to generate a two-dimensional adjoint angular flux data set FA.D2A if no old data set is available.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FFRD1 (NFR, *)

1. Arguments

Name	Type	Dimension	Definition
NFR	I*4	--	Data set reference number of one-dimensional real fluxes
*	--	--	Error return

2. Common Blocks

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of 'X' intervals

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

3. Local Variables

Name	Type	Definition
I	I*4	Index of interval number
N	I*4	Index representing group number
FRN1	R*8	One-dimensional real flux array
ITER	I*4	Iteration number
KIND	I*4	Source indicator of this set of fluxes
EVALUE	R*8	Eigenvalue for this set of fluxes

4. Functions and tasks performed by this subroutine

FFRD1 generates an initial guess real flux data set FR.D1
The first record is written containing
NINTI from common block /GEOM1/,
NGROUP from common block /XSCM/,
EVALUE set to 1.0,
KIND set to -1, and
ITER set to 0.

FRN1 is defined as 1.0 and N as NINTI*GROUP. A second record containing N FRN1's is then written.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FFRD1A (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	One-dimensional real angular flux data set reference number
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This is a dummy subroutine which represents a "black box" for the generation of the one-dimensional real angular flux data set FR.D1A.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FFRD2 (NFR, *)

1. Arguments

Name	Type	Dimension	Definition
NFR	I*4	--	Data set reference number of two-dimensional real fluxes
*	--	--	Error return

2. Common Blocks

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of intervals in 'X' direction
NINTJ	I*4	Number of intervals in 'Y' direction

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups

3. Local Variables

Name	Type	Definition
I	I*4	Index representing JIJ count
N	I*4	Index representing group number
NIJ	I*4	NINTI*NINTJ
FRN2	R*8	Two-dimensional real flux

Name	Type	Definition
ITER	I*4	Iteration number
KIND	I*4	Source of this flux set
EVALUE	R*8	Eigenvalue for this flux set

4. Functions and tasks performed by this subroutine

FFRD2 writes an initial guess real flux data set FR.D2.
The first record consists of
NINTI from common block /GEOM1/,
NINTJ from common block /GEOM1/,
NGROUP from common block /XSCM/,
EVALUE set to 1.0,
KIND set to -1, and
ITER set to 0.
NIJ is set to equal NINTI * NINTJ, and FRN2 is set to 1.0.
NGROUP records each containing NIJ of FRN2 are written.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FFRD2A (N1, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	Two-dimensional real angular flux data set reference number
*	--	--	Error return

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

This dummy subroutine represents a "black box" for generation of a two-dimensional real angular flux data set FR.D2A.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FGEOM (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NCM(16)	I*4	16 variables describing the geometry data set

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

3. Local Variables

Name	Type	Definition
N1	I*4	GEOM data set reference number

4. Functions and tasks performed by this subroutine

FGEOM is called by FORMDS to build the GEOM data set. It calls in turn,

GMTYPE,

MESHDR to establish the mesh,

CRCDR to establish compositions and regions,

RCMDR to establish mesh-region and mesh-composition correspondence,

AREADR to define the areas,

ZONEDR to define zones, and

BUCKDR to determine the finite geometry data, if any is defined.

Then if an old GEOM data set exists, a new one is not written. However, the first records of the old and new GEOM data are compared and if there is a discrepancy, the error return is used to travel back to the calling program. Otherwise, subroutine DSGEM is called to write the new GEOM.

5. Subprograms called by this subroutine

IGET (entry in POINTR)	AREADR
CRCDR	BUCKDR
DSGEM	GMTYPE
RCMDR	MESHDR
SNIFF (system module)	ZONEDR

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

(FGEOM) - OLD AND NEW GEOM QUANTITIES NUMBER 'iiii' IN FIRST RECORD OF GEOM DATA SET DO NOT AGREE, ERROR IS ASSUMED.

Subroutine FORMDS (BC, ESD1D, ESD2D, ESD1SH, ESD2SH, FAD1, FAD2, FAD1A, FAD2A, FRD1, FRD2, FRD1A, FRD2A, GEOM, SPCICN, SPCRIT, *)

1. Arguments

Name	Type	Dimension	Definition
BC	R*8	--	Boundary condition specification data set name
ESD1D	R*8	--	One-dimensional distributed external source data set name
ESD2D	R*8	--	Two-dimensional distributed external source data set name
ESD1SH	R*8	--	One-dimensional sufrace source data set name
ESD2SH	R*8	--	Two-dimensional surface source data set name
FAD1	R*8	--	One-dimensional adjoint flux data set name
FAD2	R*8	--	Two-dimensional adjoint flux data set name

Name	Type	Dimension	Definition
FAD1A	R*8	--	One-dimensional adjoint angular flux data set name
FAD2A	R*8	--	Two-dimensional adjoint angular flux data set name
FRD1	R*8	--	One-dimensional real flux data set name
FRD2	R*8	--	Two-dimensional real flux data set name
FRD1A	R*8	--	One-dimensional real angular flux data set name
FRD2A	R*8	--	Two-dimensional real angular flux data set name
GEOM	R*8	--	Geometry data set name
SPCICN	R*8	--	Module-independent constants specification data set name
SPCRIT	R*8	--	Criticality search specification data set name
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Numbers of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

/CTYP03/

Variables used from this common block

Name	Type	Definition
IPTYPE	I*4	Problem type

/CTYPO4/

Variables used from this common block

Name	Type	Definition
IEBC(4,4)	I*4	External boundary condition

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions given in geometry

3. Local Variables

Name	Type	Definition
N1	I*4	SNIFF argument representing data set reference number
SBLOCK	R*8	Data set name of data set in question

4. Functions and tasks performed by this subroutine

FORMDS handles all data sets alike by substituting various data set names for SBLOCK in the call to subroutine SNIFF.

First, a call to SNIFF with an argument of zero determines whether the data set named in SBLOCK exists. Then another SNIFF call with a 1 returns the data set reference number, which is used in the calling list to the subroutine which writes the given data set.

FGEOM is called to write GEOM and FBC is called to write BC.

(1) If N(19) > 0 and NDIM = 1, FESD1D is called; if NDIM = 2, FESD2D is called. If IPTYPE = 3 (synthesis) flux data sets are not written. FSPCCN is called. If N(21) > 0, FSPCRT is called to write SP.CRIT.

(2) However, if IPTYPE = 0 (real) and NDIM = 1, FFRD1 is called; if NDIM = 2, FFRD2 is called. If the problem is transport type, IECB(4,I)

> 8, FFRD1A or FFRD2A is called also. In either case, FSPCCN and FSPCRT, if N(21) > 0, are called.

(3) If IPTYPE = 1, subroutines FFAD1, FFAD1A, or FFAD2 and FFAD2A are called, depending upon the dimension and theory type. As before, FSPCCN and FSPCRT are called.

(4) If IPTYPE = 2, (both) the appropriate subroutines are called to write the real and adjoint flux data sets, followed by calls to FSPCCN and FSPCRT.

5. Subprograms called by this subroutine

FBC	FESD2D
FFAD1	FESD2H
FFAD2	FFAD1A
FFRD1	FFAD2A
FFRD2	FFRD1A
FGEOM	FFRD2A
SNIFF (system module)	FSPCCN
FESD1D	FSPCRT
FESD1H	

6. Subprograms calling this subroutine

GNIP

7. Error messages generated by this subroutine

Fatal:

(FORMDS) - ERROR IN SNIFF. SUB-BLOCK = 'aaaaaaaa' WAS LOCATED IN DSNAME ARRAY BUT DOES NOT YET EXIST. ON CALL TO SNIFF TO OBTAIN DATA SET REFERENCE NUMBER TO WRITE THE DATA SET, ITS NAME COULD NOT BE FOUND IN DSNAME.

(FORMDS) - ERROR IN SNIFF. SUB-BLOCK = 'aaaaaaaa' WAS LOCATED IN DSNAME ARRAY BUT DOES NOT YET EXIST. ON CALL TO SNIFF TO OBTAIN DATA SET REFERENCE NUMBER TO WRITE THE DATA SET, A REFERENCE NUMBER OF ZERO WAS RETURNED.

Subroutine FSPCCN

1. Arguments - None

2. Common Blocks

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions given in geometry
NGEOM	I*4	Type of geometry
NINTI	I*4	Number of intervals in 'X' direction
NINTJ	I*4	Number of intervals in 'Y' direction

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag - 0 - data set edit is suppressed 1 - data set edit is printed

/CTYPO3/

Variables used from this common block

Name	Type	Definition
ALPHA	R*8	Initial alpha or fixed alpha
IPTYPE	I*4	Problem type
FIXEDK	R*8	Fixed k-effective
ICTYPE	I*4	Calculation type

/CTYP21/

Variables used from this common block

Name	Type	Definition
EPSLON	I*4	Convergence criterion, epsilon for k-eff or alpha

/CTYP22/

Variables used from this common block

Name	Type	Definition
SIVX	R*8	Initial value of X, the search parameter
SSVX	R*8	Second value of X, the search parameter

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of groups

3. Local Variables

Name	Type	Definition
M	I*4	Data set reference number of SP.CRIT
X(2)	R*8	Search parameter
IPROB	I*4	Type of problem
CONVCR	R*8	Standard convergence criterion for search
MKRTHY	I*4	Problem solution type, equal to IPTYPE, card 03

4. Functions and tasks performed by this subprogram

FSPCCN writes data set SP.CICN.

If the calculation type, ICTYPE, is equal to 1, IPROB is set to 1.

If N(23) is > 0, IPROB = 3.

If N(25) is > 0, IPROB = 4.

If N(26) is > 0, IPROB = 5.

If N(24) is > 0, IPROB = 6.

Otherwise, IPROB = 2.

If N(21) = 0, EPSLON is defined as 0.0.

If N(22) = 0, SIVX = 0.0 and SSVX = 0.0.

MKRTHY is defined as IPTYPE.

CONVCR = EPSLON from common block /CTYP21/.

X(1) = SIVX from common block /CTYP22/, and X(2) = SSVX from common block /CTYP22/.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine - None

Subroutine FSPCRT (M,*)

1. Arguments

Name	Type	Dimension	Definition
M	I*4	--	SP.CRIT data set reference number
*	--	--	Error return

2. Common Blocks

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of 'X' direction intervals
NINTJ	I*4	Number of 'Y' direction intervals
NREG	I*4	Number of regions
NCMP	I*4	Number of compositions

Name	Type	Definition
NIL	I*4	Number of integration limits per region

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag - 0 - data set edit is suppressed 1 - data set edit is printed

/CTYP21/

Variables used from this common block

Name	Type	Definition
ISSPEC	I*4	Search type specification
MAXNSP	I*4	Maximum number of search passes
PKOA	R*8	Prescribed k-effective or initial alpha
EPSLON	R*8	Convergence criterion, epsilon, for k-eff or alpha

/CTYP22/

Variables used from this common block

Name	Type	Definition
SIVX	R*8	Initial value of X, the search parameter
SSVX	R*8	Second value of X
SLBX	R*8	Lower bound for X

Name	Type	Definition
SUBX	R*8	Upper bound for X
SDRX	R*8	Derivative of k-eff or alpha with respect to X

/CTYP24/

Variables used from this common block

Name	Type	Definition
NMX	I*4	Number of 'X' direction input mesh modifiers in criticality search
NMY	I*4	Number of 'Y' direction input mesh modifiers in criticality search

/CTYP26/

Variables used from this common block

Name	Type	Definition
ALPHAM	R*8	Alpha modifier for a criticality search

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

3. Local Variables

Name	Type	Definition
I	I*4	Index on X
X(3)	R*8	Parameter values of X
NN	I*4	N(23) or N(24) or N(25)
N5	I*4	N(14)
XL	R*8	Lower output bound for X

Name	Type	Definition
XU	R*8	Output upper bound for X
DKDX	R*8	Slope used to compute initial X(2) if X(2) was given as 0
DMOD	R*8	Change quantity, if necessary
LORP	I*4	Interpolation flag 1 - linear interpolation on X 0 or 2 - parabolic interpolation on X
LSPR	I*4	Length of specification record
DUMMY	R*8	Dummy
NPASS	I*4	Pass number
NPMAX	I*4	Maximum number of search passes
CONVCR	R*8	Convergence criterion
DSKEFF	R*8	Desired k-effective or desired alpha

4. Functions and tasks performed by this subroutine

FSPCRT is called from subroutine FORMDS when data set SP.CRIT has not been written and there is type 21 card input.

The following are defined and then written as the first record of SP.CRIT.

KSRCH is defined as follows:

4 if N(25) > 0, indicating a buckling search;

5 if N(26) > 0, indicating an alpha search;

1 if N(23) > 0, indicating a concentration search;

6 if both NMX and NMY > 0, indicating a dimension search in both directions;

2 if NMX > 0, indicating a dimension search in the I-direction; or

3 if none of the above are true, indicating a dimension search in the J-direction.

ICONV = 0, LORP = 0, and NPASS = 0.

NPMAX = MAXNSP from common block /CTYP21/.

LSPR = NCMP if N(23) > 0, NINTI*NINTJ if N(24) > 0, NREG*NIL if N(25) > 0, or 0 if N(26) > 0.

CONVCR = EPSLON from common block /CTYP22/, XL = SLBX from common block /CTYP22/, and XU = SUBX from common block /CTYP22/. DSKEFF = PKOA from common block /CTYP21/; X(1) = SIVX from common block /CTYP22/, and X(2) = SSVX from common block /CTYP22/. X(3) = 0.0 (not used).

DKDX = SDRX from common block /CTYP22/. DMOD = 0.0, or ALPHAM from common block /CTYP26/ if N(26) > 0. KFLAG = ISSPEC from common block /CTYP21/.

Then if N(26) > 0 set SP.CRIT is rewround and the routine returns to FORMDS.

If N(23) > 0 arrays NMC and MCOMPN are defined by PUTPNT calls. Subroutine CONSCH is called to construct and write the second record of M.

If N(24) > 0, arrays DIMOD and DJMOD are defined by PUTPNT calls and subroutine CRITSCH is called in to form and write the second record.

When N(25) > 0, array DMODB is defined by a PUTPNT call and subroutine BUCSRH is called in to write the second record of SP.CRIT. The subroutine wipes out unneeded arrays before returning.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
CONSCH
CRITSH
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)
BUCSRH

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - SUBROUTINE FSPCRT, N2 = 'ii'

(FSPCRT) - THERE IS NO SEARCH DATA GIVEN - NO CARDS OF TYPES 23, 24, 25, or 26 ARE PRESENT.

Subroutine GMTYPE

1. Arguments - None

2. Common Blocks

/CTYPO3/

Variables used from this common block

Name	Type	Definition
IGTYPE	I*4	Geometry type number

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of problem dimensions
NGEOM	I*4	Geometry type number
NFIN	I*4	Number of finite geometry type
NIL	I*4	Number of finite directions

3. Local Variables - None

4. Functions and tasks performed by this subroutine

GMTYPE defines NDIM, NGEOM, NFIN, and NIL for the first record of the GEOM data set, using imput quantity IGTYPE from card type 03.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

FGEOM

7. Error messages generated by this subroutine - None

Subroutine GNIP (NERR)

1. Arguments

Name	Type	Dimension	Definition
NERR	I*4	--	Path continuation flag

2. Common blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(30000)	R*8	Large container array

/BLNAME/

Variables used from this common block

Name	Type	Definition
BLNAME	R*8	Name of data block being read

/NCARDS/

Variables used from this common block

Name	Type	Definition
MAXN	I*4	Number of highest card type present in A.NIP
N(99)	I*4	Array of number of cards of each type present in A.NIP

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups
NCMP	I*4	Number of compositions
MINFLG	I*4	Flag = 1 if old XS.C.MIN was read

Name	Type	Definition
NDUM	I*4	Dummy for alignment
CNAME(199)	R*8	Array of composition names

3. Local Variables

Name	Type	Definition
I	I*4	Index on CNAME
BC	R*8	Boundary condition specification data set name
N1	I*4	Data set reference number
T1	R*8	Initial time
T2	R*8	Final time
ESD1D	R*8	One-dimensional distributed external source data set name
ESD2D	R*8	Two-dimensional distributed external source data set name
ESD1SH	R*8	One-dimensional surface source data set name
ESD2SH	R*8	Two-dimensional surface source data set name
FAD1	R*8	One-dimensional adjoint flux data set name
FAD2	R*8	Two-dimensional adjoint flux data set name
FAD1A	R*8	One-dimensional adjoint angular flux data set name
FAD2A	R*8	Two-dimensional adjoint angular flux data set name
FRD1	R*8	One-dimensional real flux data set name
FRD2	R*8	Two-dimensional real flux data set name
FRD1A	R*8	One-dimensional real angular flux data set name

Name	Type	Definition
FRD2A	R*8	Two-dimensional real angular flux data set name
GEOM	R*8	Geometry data set name
TIME	R*4	Time = (T2-T1)/6000 (in minutes) in module
SBLOCK	R*8	Name of sub block being located
SBNAME	R*8	Data set name in sub block being read
SPCICN	R*8	Module-independent constants specification data set name
SPCRIT	R*8	Criticality search specification data set name

4. Functions and tasks performed by this subroutine

Subroutine GNIP is called from MAIN, the module entry subroutine. Its purpose is to initialize the data set name list, to start timing the module, to initialize the large POINTR array and the error flags, and to call routines to read the input data sets and write the output data sets.

By means of SNIFF calls, it locates the necessary old data set which contains the definition of NGROUP, whether it is XS.C.MIN or XS.ISO or XS.ISO2. If none of these is found, a message stating this fact is printed and card processing and data set formation are bypassed. The module argument, NERR, is set to 1 and return is made to MAIN, which conveys the argument to the calling where execution is stopped.

When the proper old data set is found and successfully read, input data set A.NIP, which has been prepared by BCDDS, is located by a SNIFF call.

Array N of card type numbers is initialized to zero and A.NIP is read to supply the highest card type number present and the number of each card type present in A.NIP. Any error during a data set reading will cause an appropriate explanatory message to be printed.

If no error has occured, subroutine CDSCAN is called to read the various card types. Similarly, if no error has occured, FORMDS is called to write the appropriate output data sets.

Should an error occur in GNIP, CDSCAN, or FORMDS, the flag NERR is set to 1 and the argument is returned to MAIN.

After a normal return from FORMDS, there is a call to TLEFT. A Statement is printed which indicates that the module has been utilized, along with the time in minutes spent in execution of the module.

5. Subprograms called by this subroutine

TLEFT (IBM routine)
SNIFF (system module)
CDSCAN
FORMDS
POINTR (system module)

6. Subprograms calling this subroutine

MAIN

7. Error messages generated by this subroutine

Fatal:

(GNIP) - ERROR IN SNIFF. CANNOT MATCH DSNAME ENTRY BLOCK = 'aaaaaaaa', SUB-BLOCK = 'aaaaaaaa'.

(GNIP) - READ ERROR. ERROR IN DATA TRANSFER, DSNAME = 'aaaaaaaa'.

(GNIP) - XS.C.MIN HAS BEEN DECLARED AS OLD FOR GEOMETRY/NEUTRONICS INPUT PREPARATION BUT DOES NOT EXIST.

(GNIP) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. DSNAME = 'aaaaaaaa'.

(GNIP) - INPUT ERRORS HAVE MADE IT IMPOSSIBLE TO WRITE OUTPUT DATA SETS. TIME IN NUI002 WAS 'x.xx' MINUTES.

(GNIP) - A.NIP IS NEEDED FOR GEOMETRY/NEUTRONICS INPUT PREPARATION BUT DOES NOT EXIST.

Subroutine GROUPN (IHEGN, ILEGN, N, NGROUP, NG1, NG2)

1. Arguments

Name	Type	Dimension	Definition
IHEGN	I*4	N(19) N(27)	Input higher energy group for which source is given
ILEGN	I*4	N(19) N(27)	Input lower energy group for which source is given
N	I*4	--	Index number of type 19 card or index number of type 27 card
NGROUP	I*4	--	Total number of groups
NG1	I*4	--	Output initial group affected by source
NG2	I*4	--	Output final group affected by source

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

3. Local Variables - None

4. Functions and tasks performed by this subroutine

GROUPN establishes which group numbers are affected by a given source input.

Given the Nth type 19 card or the Nth type 27 card,

if IHEGN(N) = 0, the initial group = 1 and the final group = NGROUP, i.e., the source applies to all groups.

If ILEGN(N) = 0, the initial group = IHEGN(I) and the final group = IHEGN(I), i.e., the source applies to one group.

Otherwise, the initial group = IHEGN(I) and the final group = ILEGN(I), i.e., the source applies over the range of groups given as input.

These two group numbers are passed back to the calling subroutine as NG1 and NG2.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

I1SORD
I2SORD
ANISO1
ANISO2

7. Error messages generated by this subroutine - None

Subroutine GRPSOR (IHEGN, ILEGN, M, NGROUP, NG1, NG2)

1. Arguments

Name	Type	Dimension	Definition
IHEGN	I*4	N(19)	Input higher energy group number to which source value applies
ILEGN	I*4	N(19)	Input lower energy group number to which source value applies
M	I*4	--	Number of type 19 cards
NGROUP	I*4	--	Number of energy groups
NG1	I*4	N(19)	Initial energy group number to which source on given card applies
NG2	I*4	N(19)	Final energy group number to which source on given card applies

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
N	I*4	Index on type 19 card input; also index on arrays NG1 and NG2

4. Functions and tasks performed by this subroutine

GRPSOR defines the arrays NG1 and NG2, which are used as indices in subroutine I2SORD.

When the input higher energy group number, IHEGN(N), equals zero, NG1(N) = 1 and NG2(N) = NGROUP; the source applies to all groups.

When the input lower energy group number, ILEGN(N), equals zero, NG1(N) = IHEGN(N) and NG2(N) = IHEGN(N); i.e., the source applies only to the given group.

Otherwise NG1(N) = IHEGN(N) and NG2(N) = ILEGN(N); the source applies over the range of groups IHEGN(N) to ILEGN(N).

5. Subprograms called by this subroutine - None

6. Subroutines calling this subroutine

FESD2D

7. Error messages generated by this subroutine - None

Subroutine INITZE (HDUR1, HDUR2, HDUR3, NPTA, NINTB, NGROUP, NORDER, IDUMZ)

1. Arguments

Name	Type	Dimension	Definition
HDUR1	R*8	NPTA*NGROUP	Isotropic component of source
HDUR2	R*8	NPTA*NGROUP	Anisotropic component in 'X' direction
HDUR3	R*8	NPTA*NGROUP	Anisotropic component in 'Y' direction
NPTA	I*4	--	Number of mesh lines in first direction
NINTB	I*4	--	Number of mesh intervals in second direction
NGROUP	I*4	--	Number of energy groups

Name	Type	Dimension	Definition
NORDER	I*4	--	Order of anisotropy
IDUMZ	I*4	NPTA*NINTB	Pointers to locations in source arrays HDUR1, HDUR2, HDUR3

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of intervals; also index on number of points
J	I*4	Index on number of intervals
K	I*4	Index on number of energy groups

4. Functions and tasks performed by this subroutine

Subroutine INITZE initializes the arrays used in forming the inhomogeneous anisotropic source data sets, ES.D1SH or ES.D2SH.

The isotropic, anisotropic 'X' direction and anisotropic 'Y' direction arrays are initialized to zero.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

ANISO1
ANISO2

7. Error messages generated by this subroutine - None

Subroutine INTBND (N11, NIY, YMESH, NPTJ, M, MFACE, M1, M2, NBIT, J1, JF, C1, C2, C3, C4, ULABEL, YILABL, IBHEGN, IBLEGN, YLEI, YUEI, BCA1, BCB1, BCA2, BCB2, ICONY, YXIC, IFLAG, XMESH, NPTI, *)

1. Arguments

Name	Type	Dimension	Definition
N11	I*4	--	Number of type 11 cards
NIY	I*4	--	Size of internal input quantity array for first direction
YMESH	R*8	NPTJ	First direction mesh coordinate array
NPTJ	I*4	--	Number of lines in first direction
M	I*4	--	Index of next record to be defined
MFACE	I*4	NBC	Output mesh line number and side indicator
M1	I*4	NBC	Output line number defining lower end of interval along given face
NBIT	I*4	NBC	Output type number of boundary or interface condition
JI	I*4	NBC	Output initial higher-energy group number for which constants apply
JF	I*4	NBC	Output final lower energy group number for which constants apply
C1	R*8	NBC	First output constant
C2	R*8	NBC	Second output constant
C3	R*8	NBC	Third output constant
C4	R*8	NBC	Fourth output constant
ULABEL	R*8	N(11)	Internal interface label from type 11 card
YILABL	R*8	NIY	First direction internal interface label from type 10 card
IBHEGN	I*4	N(11)	Input higher energy group number for which constants apply
IBLEGN	I*4	N(11)	Input lower energy group number for which constants apply
YLEI	R*8	NIY	Input coordinate at lower end of internal interface in first direction

Name	Type	Dimension	Definition
YUEI	R*8	NIY	Input coordinate at upper end of internal interface in first direction
BCA1	R*8	N(11)	First input constant to interface
BCB1	R*8	N(11)	Second input constant to interface
BCA2	R*8	N(11)	Third input constant to interface
BCB2	R*8	N(11)	Fourth input constant to interface
ICONY	I*4	NIY	Internal condition number on first direction interface
YXIC	R*8	NIY	Second direction coordinate along which first direction interface lies
IFLAG	I*4	--	Input dimension flag
XMESH	R*8	NPTI	Second direction mesh coordinates
NPTI	I*4	--	Number of second direction mesh lines
*	--	--	Error return

2. Common Blocks

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of groups in problem

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions in geometry

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag - 0 - data set edit is suppressed 1 - data set edit is printed

3. Local Variables

Name	Type	Definition
B	R*8	Represents |YLEI(I)-YMESH(1)|
D	R*8	Represents |YUEI(I)-YMESH(NPTJ)|
H	R*8	C1(M)*C4(M)-C3(M)*C2(M) The determinant of the matrix $\begin{bmatrix} C1 & C2 \\ C3 & C4 \end{bmatrix}$
I	I*4	Index on first direction input
K	I*4	Index on card type 11 input
L	I*4	Index on first direction input
IY	I*4	ICONY(I)+1
DIF	R*8	Input mesh coordinate minus actual mesh coordinate, or YXIC(I) minus XMESH(J)
NPTJ	I*4	Number of lines in first direction
J	--	Index on second direction line numbers

4. Functions and tasks performed by this subroutine

INTBND is called from subroutine FBC. It fills in the output arrays for the BC data set by processing the type 10 and type 11 card input quantities, one dimension at a time. Therefore, for a two-dimensional problem it is entered twice.

For the sake of convenience, the procedure is described for Y-direction boundary input.

In the following, I represents the Ith Y direction input card and K represents the Kth type 10 card.

First, the ULABEL(K) on a type 11 card is matched to some label YILABL(I) from a type 10 card. Then the YXIC(I) is compared to all XMESH(J) to see if the boundary lies on some mesh line. If not, the error return is used.

If agreement is found, MFACE(M) = -J and MFACE(M+1) = J, where M is the index of the next array member and BC record.

If the problem is two-dimensional, a check is made to see that the internal interface does not extend across the entire reactor when the data constants apply to all groups.

Then,

if |YLEI(I)-YMESH(L)| < 0.001 M1(M) = L,

and if |YUEI(I)-YMESH(L)| < 0.001 M2(M) = L.

Then, M1(M+1) = M1(M)

and M2(M+1) = M2(M).

If IBHEGN(K) = 0, JI(M) = 1

and JF(M) = NGROUP;

also, JI(M+1) = 1,

and JF(M+1) = NGROUP.

If IBLEGN(K) = 0,

JI(M) = IBHEGN(K), JF(M) = IBHEGN(K), JI(M+1) = IBHEGN(K),

and JF(M+1) = IBHEGN(K).

Otherwise JI(M) = IBHEGN(K), JF(M) = IBLEGN(K),

JI(M+1) = IBHEGN(K), and JF(M+1) = IBLEGN(K).

IY is set equal to ICONY(I), the internal condition + 1.

If IY = 1 or IY = 2,

NBIT(M) = 2 and NBIT(M+1) = 2.

C1(M) = 1.0, C1(M+1) = 1.0,

C2(M) = BCA1(K), C2(M+1) = BCA2(K),

C3(M) = BCB1(K), C3(M+1) = BCB2(K),

C4(M) = 0.0, and C4(M+1) = 0.0

If IY = 3,

NBIT(M) = 3 and NBIT(M+1) = 3.

C1(M) = BCA1(K),

C2(M) = BCB1(K),

C3(M) = BCA2(K), and

C4(M) = BCB2(K).

H = C1(M)*C4(M)-C3(M)*C2(M) and if H = 0 an error message is printed, and processing continues. Quantity ISTOP is set to 1.

C1(M+1) = C4(M)/H,

C2(M+1) = -C2(M)/H,

C3(M+1) = -C3(M)/H, and

C4(M+1) = C1(M)/H.

If IY = 4,

NBIT(M) = 1 and NBIT(M+1) = 1.

C1(M) = BCA1(K), C1(M+1) = BCA2(K),

C2(M) = BCB1(K), C2(M+1) = BCB2(K),

C3(M) = 0.0, C3(M+1) = 0.0,

C4(M) = 0.0, and C4(M+1) = 0.0.

If IY = 5,

NBIT(M) = 2, NBIT(M+1) = 2, and the constants are defined as when IY = 4.

When IY = 6,

NBIT(M) = 1 and NBIT(M+1) = 1.

C1(M) = BCA1(K), C1(M+1) = BCA2(K),

C2(M) = 1.0, C2(M+1) = 1.0

C3(M) = BCB1(K), C3(M+1) = BCB2(K),

C4(M) = 0.0, and C4(M+1) = 0.0.

If IY = 7,

NBIT(M) = 2, NBIT(M+1) = 2, and the constants are defined as when IY = 6.

In any case, M is incremented by 2 after the definitions have been made, and the next two numbers of each array are defined, until 2 * NX records have been defined.

INTBND is entered a second time for two-dimensional problems in order to define 2 * NY vertical boundary condition records. The process is the same but X's and Y's are transposed in the argument list.

If ISTOP is 1, the error return is used.

5. Subprograms called by this subroutine

DUMP (entry in POINTR)

6. Subprograms calling this subroutine

FBC

7. Error messages generated by this subroutine

Fatal:

(INTBND) - COORDINATE INPUT LINE NUMBER 'ii' ON CARD TYPE 10 DOES NOT AGREE WITH MESH STRUCTURE.

(INTBND) - ERROR ON CARD NUMBER 'ii' OF TYPE 11, 2D BOUNDARY CONSTANTS CANNOT APPLY TO ALL GROUPS WHEN INTERNAL BOUNDARY EXTENDS THE WIDTH OF THE REACTOR.

(INTBND) - MATRIX 'ii' HAS NO INVERSE.

Subroutine I1SORD (IFLAG, FSEXT, ISPECN, XMESH, YMESH, IICODE, ILEGN, IHEGN, RLSIC, RUSIC, AREA, REG, ZONE, SILABL, SOURC, MR, MC, NRA, NSB, NINTI, NINTJ, N1, NREGA, NSECTZ, *)

1. Arguments

Name	Type	Dimension	Definition
IFLAG	I*4	NINTI	Output group weighting flag for mesh interval
FSEXT	R*8	NINTI*NGROUP	Output isotropic source value
ISPECN	I*4	N(19)	Spectrum distribution flag
XMESH	R*8	NINTI	'X' direction mesh coordinates
YMESH	R*8	NINTJ	'Y' direction mesh coordinates
IICODE	I*4	N(19)	Input type flag
ILEGN	I*4	N(19)	Lower energy group number to which source applies
IHEGN	I*4	N(19)	Higher energy group number to which source applies

Name	Type	Dimension	Definition
RLSIC	R*8	N(19)	Lower 'X' direction coordinate of interval containing source
RUSIC	R*8	N(19)	Lower 'Y' direction coordinate of interval containing source
AREA	R*8	NAREA	Area labels
REG	R*8	NREG	Region labels
ZONE	R*8	NZONE	Zone labels
SILABL	R*8	N(19)	Input label for source range
SOURC	R*8	N(19)	Input isotropic source value
MR	I*4	NINTI*NINTJ	Region index of mesh points
MC	I*4	NINTI*NINTJ	Composition index of mesh points
NRA	I*4	LREGA	Region indices of areas
NSB	I*4	LSECT	List of section boundary mesh index numbers
NINTI	I*4	--	Number of 'X' direction intervals
NINTJ	I*4	--	Number of 'Y' direction intervals
N1	I*4	--	Data set reference number of ES.D1D
NREGA	I*4	NAREA	Number of regions in each area
NSECTZ	I*4	NZONE	Number of sections in each zone
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/GEOM1/

Variables used from this common block

Name	Type	Definition
NAREA	I*4	Number of areas
NZONE	I*4	Number of zones
NPTI	I*4	Number of 'X' direction mesh lines
NREG	I*4	Number of regions

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of groups

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of 'X' coordinates; also index on number of 'X' intervals; also index on output FSEXT and IFLAG
J	I*4	Index on number of 'Y' intervals
K	I*4	Index on group numbers; also index on NRA; also index on output FSEXT and IFLAG
L	I*4	Index on number of regions; also index on number of areas; also index on number of zones
M	I*4	Index on number of type 19 cards

Name	Type	Definition
II	I*4	X-mesh index number on output FSEXT and IFLAG
KK	I*4	Group index number
LO	I*4	Lower limit of index M1 on NSB; also lower limit on index K on NRA
LR	I*4	Quantity equal to L, the region index number
L1	I*4	Lower limit of index II on FSEXT and IFLAG
M1	I*4	Index on NSB
M2	I*4	M1 + 1, also index on NSB
IHI	I*4	Upper limit of index K on NRA; also upper limit of index M1 on NSB
NG2	I*4	Final group number
NG1	I*4	Initial group number
N19	I*4	Number of type 19 cards
NBLANKS	R*8	Alphanumeric blanks

4. Functions and tasks performed by this subroutine

I1SORD takes type 19 card input and writes output data set ES.D1D.

Arrays IFLAG and FSEXT are initialized to zero.

For every input card M, the following is performed.

The quantities IHEGN(M) and ILEGN(M) are passed to subroutine NGROUP to obtain NG1 and NG2, the initial and final group indices for the output.

Then, if IICODE(M) = 0, SILABL(M) contains a region label which is compared with all REG(L) labels. When a match is found, a search of array MR is performed. For every MR(I,J) = L,

FSEXT(I,J) = SOURCE(M) and, if ISPECN(M) = 1, IFLAG(I) = 1, where I is the interval number and J is over the range of group numbers.

If IICODE(M) = 1, SLABL(M) contains an area label which is compared to all AREA(L) labels. Counter LO, initialized to 1, is incremented by

NREGA(L), the number of regions in AREA(L), until a matching label is found.

IHI is then set to LO + NREGA(L)-1 to obtain the index of last element of NRA in area (L).

All MR(I,J) are then compared to NRA(K), where K ranges from LO to IHI, and for every instance where MR(I,J) = NRA(K), FSEXT(I,KK) is defined as SOURCE(M). KK is over the range of group numbers.

If ISPECN(M) = 1, IFLAG(I) = 1.

When IICODE(M) = 2, SILABLE(M) is a zone label which is compared with all ZONE(L) labels. Counter LO, equal to 1 at the start, is incremented by 2 * NSECTZ(L)-2 for each zone checked, until a match is found. This then, is the lower index of the NSB contained in zone (L). The upper index of the NSB contained in zone (L) is equal to (L) + 2 * NSECTZ(L)-2.

The interval index for outputting FSEXT(II,K) is found by comparing every second NSB(M1) to the mesh line index numbers.

Setting M2 = M1 + 1, if NSB(M1) = I, L1 = I, and if NSB(M2) = I, L2 = I-1.

Then FSEXT (II,K) is defined as SOURCE(M), and if ISPECN(M) = 1, IFLAG(II) = 1, where II takes the integar values from L1 to L2, and K is over the range of group numbers.

Where IICODE(M) is blank, the input applies to one mesh interval. Therefore, RLSIC(M) is compared to all XMESH(I) until a match is found. Then FSEXT(II,K) is defined as SOURCE(M), where II = I, and K is over the range of group numbers. If ISPECN(M) = 1, IFLAG(II) is set to 1.

If no mesh line is found in the mesh structure to correspond within 0.001 to an input mesh line, an error message is printed and the error return is made.

When all type 19 cards are processed, data set ES.D1D is written. Unneeded arrays are wiped out before returning.

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

DUMP (entry in POINTR)
GROUPN
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FESD1D

7. Error messages generated by this subroutine

Fatal:

(I1SORD) - DATA ON CARD NUMBER 'ii' OF TYPE 19 IN A.NIP DOES NOT MATCH MESH STRUCTURE.

Subroutine I2SORD (IFLAG, FSEXT, ISPECN, XMESH, YMESH, IICODE, ILEGN, IHEGN, RLSIC, RUSIC, AREA, REG, ZONE, SILABL, SOURC, MR, MC, NRA, NSB, NINTI, NINTJ, N1, NREGA, NSECTZ, NG1, NG2, *)

1. Arguments

Name	Type	Dimension	Definition
IFLAG	I*4	NINTI*NINTJ	Output group weighting flag for mesh interval
FSEXT	R*8	NINTI*NINTJ	Output isotropic source value
ISPECN	I*4	N(19)	Spectrum distribution flag
XMESH	R*8	NINTI	'X' direction mesh coordinates
YMESH	R*8	NINTJ	'Y' direction mesh coordinates
IICODE	I*4	N(19)	Input type flag
ILEGN	I*4	N(19)	Lower energy group number to which source applies
IHEGN	I*4	N(19)	Higher energy group number to which source applies
RLSIC	R*8	N(19)	Lower 'X' direction coordinate of interval containing source
AREA	R*8	NAREA	Area labels
REG	R*8	NREG	Region labels
ZONE	R*8	NZONE	Zone labels

Name	Type	Dimension	Definition
SILABL	R*8	N(19)	Input label for source range
SOURC	R*8	N(19)	Input isotropic source value
MR	I*4	NINTI*NINTJ	Region index of mesh intervals
MC	I*4	NINTI*NINTJ	Composition index of mesh points
NRA	I*4	LREGA	Region indices of areas
NSB	I*4	LSECT	List of section boundary mesh index numbers
NINTI	I*4	--	Number of 'X' direction intervals
NINTJ	I*4	--	Number of 'Y' direction intervals
N1	I*4	--	Data set reference number of ES.D1D
NREGA	I*4	NAREA	Number of regions in each area
NSECTZ	I*4	NZONE	Number of sections in each zone
NG1	I*4	N(19)	Initial group number for the data on type 19 card
NG2	I*4	N(19)	Final group number for the data on type 19 card
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/GEOM1/

Variables used from this common block

Name	Type	Definition
NPTI	I*4	Number of 'X' direction mesh lines
NPTJ	I*4	Number of 'Y' direction mesh lines
NREG	I*4	Number of regions
NAREA	I*4	Number of areas
NZONE	I*4	Number of zones

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of groups

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of 'X' intervals; index on number of 'Y' corrdinates; index on output FSEXT and IFLAG
J	I*4	Index on number of 'Y' intervals; index on number of 'Y coordinates; also 'Y' interval index on output FSEXT and IFLAG
K	I*4	Index on group numbers for output FSEXT and IFLAG; also index on NRA
L	I*4	Index on number of regions, areas, and zones

Name	Type	Definition
M	I*4	Index on number of type 19 cards
II	I*4	X mesh index number on output FSEXT and IFLAG
JJ	I*4	Y mesh index number on output FSEXT and IFLAG
JN	I*4	Index on groups affected by a type 19 card ranging from N3 to N2
KK	I*4	Group index number
LO	I*4	Lower limit of index M1 on NSB; also lower limit of index K on NRA
L1	I*4	Lower limit of index II on FSEXT and IFLAG
L2	I*4	Upper limit of index II on FSEXT and IFLAG
L3	I*4	Lower limit of index JJ on FSEXT and IFLAG
L4	I*4	Upper limit of index JJ on FSEXT and IFLAG
M1	I*4	Index on NSB
M2	I*4	M1 + 1, index on NSB
M3	I*4	M1 + 2, index on NSB
M4	I*4	M1 + 3, index on NSB
N2	I*4	Equal to NG2(M), the upper limit of index JN defining groups affected by card (M)
N3	I*4	Equal to NG1(M), the lower limit of index JN defining groups affected by card (M)
IHI	I*4	Upper limit of index M1 on NSB; also upper limit of index K on NRA
N19	I*4	Number of type 19 cards
NFLAG	I*4	Flag - if 1, first record of ES.D2D has been written
BLANKS	R*8	Alphameric blanks

4. Functions and tasks performed by this subroutine

I2SORD writes ES.D2D from type 19 card input.

Since it is necessary to write the sources one group at a time to conserve array space, the code is utilized several times. Data for the first record of ES.D2D is collected first.

Array IFLAG and flag NFLAG are initialized to zero. The IFLAG array is defined as described below, and the specification record of ES.D2D is written on data set N1. NFLAG is than set to 1.

The input to each group KK is collected as follows. After initializing source array FSEXT to zero, the group limit arrays NG1(M) and NG2(M) from the type 19 cards are searched for limits which contain KK. For each card M containing data for group KK, FSEXT is defined as described below. After all the data for group KK has been collected and the sources defined, the source record for that group is written on data set ES.D2D.

If IICODE(M) = 0, SILABL(M) contains a region label, which is compared with all REG(L) labels. When a matching REG(L) is found, array MR is searched for MR(I,J) equal to L.

For every MR(I,J) = L,

FSEXT(I,J) = SOURCE(M) and, if ISPECN(M) = 1, IFLAG(I) = 1, where I is the 'X' interval number and K is the range of group numbers.

If IICODE(M) = 1, SILABL(M) contains an area label. After initializing LO to 1, all AREA(L) labels are compared with SILABL(M), adding NREGA(L) to LO for each AREA(L) passed over until a match is found. Then IHI is defined as LO + NREGA(L)-1 to obtain the index of the last element in NRA in AREA (L). All MR(I,J) are then compared with those NRA(K), where K is the range of affected region indices in the given area, and for every MR(I,J) = NRA(K), FSEXT(I,J) is defined as SOURCE(M). If ISPECN(M) = 1, IFLAG(I,J) is set to 1, where I is the 'X' interval number, and J is the 'Y' interval number.

If IICODE(M) = 2, element SILABL(M) contains a zone label, which is compared to all ZONE(L) labels. Counter LO, set equal to 1, is incremented by 4*NSECTZ(L) for every ZONE(L) until a match is found. Then IHI is set to LO + 4*NSECTZ(L)-4 for the upper index on M1. Setting M2 = M1 + 1, M3 = M1 + 2, and M4 = M1 + 3, every fourth element of NSB is compared as follows:

With 'X' mesh line numbers I, for each NSB(M1) = I, L1 is set to I; for each NSB(M2) = I, L2 is set to I-1. Then with each 'Y' mesh line number J, if NSB(M3) = J, L3 is set to J, and if NSB(M4) = J, L4 is set to J-1.

FSEXT(II,JJ) is defined as SOURCE(M) for all II over the range L1 to L2, JJ over the range L3 to L4. If ISPECN(M) = 1, IFLAG(II,JJ) is set to 1.

Finally, if SILABL(M) is blank, indicating input to a single mesh cell, the coordinate RLSIC(M) is compared with all YMESH(J) until correspondence is found for the affected interval (I,J).

FSEXT(II,JJ) is then defined as SOURCE(M), and if ISPECN(M) = 1, IFLAG(II,JJ) is set to 1, where II = I and JJ = J.

After the last group record has been written, data set ES.D2D is rewound and unneeded arrays are wiped out.

5. Subprograms called by this subroutine

DUMP (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FESD2D

7. Error messages generated by this subroutine

Fatal:

(I2SORD) - DATA ON CARD NUMBER 'ii' OF TYPE 19 IN A.NIP DOES NOT MATCH MESH STRUCTURE.

Main Program MAIN (IERR)

1. Arguments

Name	Type	Dimension	Definition
IERR	I*4	--	ERROR FLAG - if IERR = 1, module execution has not been successful

2. Common Blocks - None

3. Local Variables - None

4. Functions and tasks performed by this program

This program calls the module driver and returns an argument to the calling program.

5. Subprograms called by this program

GNIP

6. Programs calling this program

ARC System standard paths and other ARC System modules

7. Error messages generated by this program - None

Subroutine MESHC (XMESH, XL, XU, NINTX, YMESH, YL, YU, NINTY)

1. Arguments

Name	Type	Dimension	Definition
XMESH	R*8	NPTI	'X' coordinate
XL	R*8	N(6)	Lower 'X' coordinate defining left side of region
XU	R*8	N(6)	Upper 'X' coordinate defining right side of region
NINTX	I*4	N(6)	Number of intervals inserted in 'X' direction
YMESH	R*8	NPTJ	'Y' coordinates
YL	R*8	N(6)	Lower 'Y' coordinate defining bottom of region
YU	R*8	N(6)	Upper 'Y' coordinate defining top of region
NINTY	I*4	N(6)	Number of intervals to be inserted in 'Y' direction

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	--	Number of dimensions in geometry
NPTI	--	Number of lines in 'X' direction
NPTJ	--	Number of lines in 'Y' direction
NINTI	--	Number of intervals in 'X' direction
NINTJ	--	Number of intervals in 'Y' direction

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 6 card 'X' direction input
J	I*4	Index on type 6 card 'Y' direction input
K	I*4	Index on type 6 card input
NI	I*4	Equal to NINTX(K), the number of 'X' direction intervals to be inserted in region K
NJ	I*4	Equal to NINTY(K), the number of 'Y' direction intervals to be inserted in region K
N6	I*4	Number of type 06 cards in A.NIP

4. Functions and tasks performed by this subroutine

MESHC is the constant mesh definition subroutine. If the region boundaries are defined by only the upper coordinates on other than the

first card, the rest of the lower coordinates XL(I) are defined as XU(I-1).

The number of mesh intervals is defined by counting the number of type 06 cards from data set A.NIP.

Then for the number of inserted 'X' direction intervals,

XMESH (NPTI + I) = XMESH(NPTI + I - 1) + XI, where XI represents the interval length to be added, or $\frac{XU(K) - XL(K)}{NINTX(K)}$, and XMESH(NPTI + NI) = XU(K).

Then NPTI is incremented by the number of intervals added.

Subroutine MSHORD is then called to eliminate duplicate coordinates and to adjust the value of NPTI. If NDIM = 2, the same procedure is carried out using YMESH, NPTJ, YL, YU, etc. instead of 'X' - direction quantities.

For one-dimensional problems, NPTJ = 1 and YMESH(1) = 0.0.

5. Subprograms called by this subroutine

MSHORD

6. Subprograms calling this subroutine

MESHDR

7. Error messages generated by this subroutine - None

Subroutine MESHDR (NINVX, NINVY, NINTX, NINTY, *)

1. Arguments

Name	Type	Dimension	Definition
NINVX	I*4	NVX	Input number of intervals to be inserted in 'X' direction variable mesh
NINVY	I*4	NVY	Input number of intervals to be inserted in 'Y' direction variable mesh
NINTX	I*4	N(6)	Input number of intervals to be inserted in 'X' direction constant mesh

Name	Type	Dimension	Definition
NINTY	I*4	N(6)	Input number of intervals to be inserted in 'Y' direction constant mesh
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/CTYPO9/

Variables used from this common block

Name	Type	Definition
NVX	I*4	Number of input quantities in 'X' direction
NVY	I*4	Number of input quantities in 'Y' direction

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions in geometry

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 06 'X' direction input
J	I*4	Index on type 06 'Y' direction input
N6	I*4	Number of type 06 cards
DUMMY	R*8	Dummy

4. Functions and tasks performed by this subroutine

MESHDR sets up the required arrays for mesh definition. NPTI and NPTJ are initialized to 1. If type 09 cards are input, the NVX 'X' - direction type 09 input is searched and the total number of 'X' points and 'X' intervals to be inserted is determined. The same is done for the NVY 'Y' direction input quantities. Request is then made for space for array XMESH of length NPTI and for YMESH of length NPTJ. If a POINTR error does not occur, the pointers to these arrays and also to arrays XL, YL, and UCX are passed to MESHV, where the variable mesh structure is determined. The arrays NINVX and UCX are then wiped out and the subroutine returns.

If type 09 input does not exist, the type 06 card arrays are used and the numbers of points and intervals are similarly counted. Request is made as before for array space for XMESH and YMESH, but subroutine MESHC is called. After the return, NINTX, and NINTY, the input arrays, are wiped out.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
MESHC
MESHV
INTERR (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - SUBROUTINE MESHDR, N2 = 'ii'

Subroutine MESHV (XMESH, NINVX, UCX, YMESH, NINVY, UCY, XL, YL)

1. Arguments

Name	Type	Dimension	Definition
XMESH	R*8	NPTI	Array of X direction coordinates
NINVX	I*4	NVX	Array of number of 'X' intervals to be inserted between X's
UCX	R*8	NVX	Upper 'X' coordinate of mesh interval to be divided
YMESH	R**	NPTJ	Array of 'Y' direction coordinates
NINVY	I*4	NVY	Array of number of 'Y' intervals to be inserted between Y coordinates
UCY	R*8	NVY	Upper Y coordinate of mesh interval to be divided
XL	R*8	NVX	'X'-direction lower coordinate of region boundary
YL	R*8	NVY	'Y'-direction lower coordinate of region boundary

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/CTYPO9/

Variables used from this common block

Name	Type	Definition
NVX	I*4	Number of type 09 input values for 'X'-direction variable mesh
NVY	I*4	Number of type 09 input values for 'Y'-direction variable mesh

/GEOM1/

Variables used from this common block

Name	Type	Definition
NPTI	I*4	Number of 'X' direction lines
NPTJ	I*4	Number of 'Y' direction lines
NINTI	I*4	Number of 'X' direction intervals
NINTJ	I*4	Number of 'Y' direction intervals

3. Local Variables

Name	Type	Definition
I	I*4	Index used in counting added 'X' intervals
J	I*4	Index used to count added 'Y' intervals
K	I*4	Index on NUX or NUY
IN	I*4	Counter for 'X' direction interval number at start of each 'X' insertion
JN	I*4	Counter for 'Y' direction interval number at start of each 'Y' insertion
N6	I*4	Number of Type 06 cards
UC	R*8	Distance between coordinates which will be divided
NPI	I*4	'X' direction counter of inserted 'X' intervals
NPJ	I*4	'Y' direction counter of inserted 'Y' intervals

4. Functions and tasks performed by this subroutine

MESHV is called when type 09 cards are present in A.NIP. The quantities NINTI and NINTJ as well as arrays XMESH and YMESH are defined for the GEOM data set.

From card 09 UCX represents the upper limit for each interval insertion range.

Array XL from card type 06 represents the array of lower region coordinates.

XMESH(1) is defined as the lowest 'X' coordinate contained in this array, and is then incremented to form array XMESH(I).

With I taking values of 1 to NVX, the number of X input values, and NPI representing the index number of intervals inserted, NPI is initialized to 1 and UC = UCX(K) - XMESH(NPI). The lower limit, IN, is set to 1 + NPI; the upper limit, NPI, is set to NPI + NINVX(K).

Then, for each mesh interval I over the interval [IN,NPI],

$$XMESH(I) = XMESH(I-1) + \frac{UC}{NINVX(K)} \quad .$$

This is repeated for YMESH when the problem is two-dimensional; otherwise, NINTJ = 1 and YMESH(1) = 0.0. NINTI is set equal to NPTI-1, and NINTJ is set to NPTJ-1.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

MESHDR

7. Error messages generated by this subroutine - None

Subroutine MSHORD (XMESH, NPTI)

1. Arguments

Name	Type	Dimension	Definition
XMESH	R*8	NPTI	Array of 'X' coordinates
NPTI	I*4	--	Number of 'X' mesh lines

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
I	I*4	Index on 'X' temporary coordinates
J	I*r	Index on remaining lines
K	I*4	Index on 'X' coordinates
NPT	I*4	Counter for number of 'X' lines
TEMP	R*8	Temporary name for 'X' coordinate

4. Functions and tasks performed by this subroutine

MSHORD is called to reorder the mesh points in ascending order and to eliminate duplicate coordinates in a given direction.

It is called once for 'X' coordinates and once for 'Y' coordinates.

In eliminating duplicate coordinates, the correct value of NPTI or NPTJ is obtained by subtracting 1 for each duplicate coordinate.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

MESHC

7. Error messages generated by this subroutine - None

Subroutine RCMDR(*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/GEOM1/

Variables used from this common block

Name	Type	Definition
NINTI	I*4	Number of intervals in 'X' direction
NINTJ	I*4	Number of intervals in 'Y' direction

3. Local Variables

Name	Type	Definition
DUMMY	R*8	Dummy

4. Functions and tasks performed by this subroutine

RCMDR is a driver routine to organize the arrays necessary to the region-mesh correspondence and the composition-mesh correspondence. It requests space for arrays MR and MC and relays the pointers of these and several other arrays, NUMR, NUMC, XL, XU, XMESH, YL, YU, YMESH, and NC to subroutine RCMESH.

If a POINTR error occurs, a message is printed to indicate array overflow and the error return is used. Otherwise, unneeded arrays are wiped out before returning.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
RCMESH
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FGEOM

7. Error messages generated by this subroutine

Fatal:

ARRAY OVERFLOW - SUBROUTINE RCMDR, N2 = 'ii'

Subroutine RCMESH (MR, NUMR, MC, NUMC, XL, XU, XMESH, YL, YU, YMESH, NC, NINTI, *)

1. Arguments

Name	Type	Dimension	Definition
MR	I*4	NINTI*NREG	Region index of mesh interval
NUMR	I*4	N(6)	Region numbers
MC	I*4	NINTI*NCMP	Composition index of mesh interval
NUMC	I*4	N(14)	Composition numbers
XL	R*8	N(6)	Lower 'X' direction coordinate defining left region boundary
XU	R*8	N(6)	Upper 'X' coordinate defining right region boundary
XMESH	R*8	NPTI	'X' coordinate array
YL	R*8	N(6)	Lower 'Y' coordinate defining bottom region boundary
YU	R*8	N(6)	Upper 'Y' coordinate defining top region boundary
YMESH	R*8	NPTJ	'Y' coordinate array
NC	I*r	N(6)	Composition index of regions
NINTI	I*4	---	Number of 'X' intervals
*	--	--	Error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Number of dimensions given in geometry
NPTI	I*4	Number of 'X' direction mesh lines
NPTJ	I*4	Number of 'Y' direction mesh lines
NINTJ	I*4	Number of 'Y' direction intervals

3. Local Variables

Name	Type	Definition
I	I*4	Index on XMESH: also index for MR and MC output
J	I*4	Index on YMESH; also index for MR and MC output
K	I*4	Index on type 06 output
I2	I*4	Second index on XMESH
J2	I*4	Second index on YMESH
NI	I*4	Lower index value of I in defining MR and MC
NJ	I*4	Upper index value of I in defining MR and MC
N6	I*4	Number of type 06 cards
NI2	I*4	Lower index value of J used in defining MR and MC
NJ2	I*4	Upper index value of J used in defining MR and MC
NERR	I*4	Error flag
IFLAG	I*4	Flag on input type

4. Functions and taks performed by this subroutine

RCMESH is called by subroutine RCMDR to define arrays MR, the mesh-region correspondence, and MC, the mesh-composition correspondence.

If the input has been by upper coordinate only, i.e. XU(K) > and XL(K)=0, then XL(K) is defined as XU-(K-1) for all regions after the first.

Proceeding through the YMESH array, it is determined which mesh point the X2(K) is closest to by testing if |XMESH(I)-XL(K)|>|XMESH(I-1)-XL(K)|. If this is true, I is set to I-1.

Otherwise, a test for |XMESH(I)-XL(K)|>0.01 is performed. If this is true, the input boundary does not coincide with the mesh structure. The error is noted and processing continues.

If I=1, only the second test is made. When no error occurs, XMESH(I)=XL(K) and NI=I.

Similar methods are followed to define XMESH(I2)=XU(K) and NI2=I-1; YMESH(J)=YL(K) and NJ=J; and, lastly, YMESH(J2)=YU(K) and NJ2=J2-1.

These limits are used to define MR(I,J)=NUMR(K) and MC(I,J)-NC(NUMR(K)), where I and J refer to the interval, NUMR(K) is the Kth region number, and NC is the composition number of the Kth region.

The MR(I,J) array is then scanned for zeroes. If any are encountered, a message is printed saying that this mesh interval did not get a region assignment.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

RCMDR

7. Error messages generated by this subroutine

Fatal:

(RCMESH) - MESH 'ii', 'ii' DID NOT GET A REGION ASSIGNMENT - CHECK INPUT TO A.NIP.

(RCMESH) - ERROR IN REGION SPECIFICATIONS HAS MADE IT IMPOSSIBLE TO CONTINUE WITH FGEOM.

ERROR XL, 'K', 'I', 'J', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'

ERROR XU, 'K', 'I', 'J2', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'

ERROR YL, 'K', 'I', 'J', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'

ERROR YU, 'K', 'I', 'J2', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'

The above 4 messages contain the values of the variables given in quotation marks, as defined in subroutine RCMESH.

Subroutine REC10 (X, Y, REGMSH, REGHHT, REGVOL, NI, NR, N1)

1. Arguments

Name	Type	Dimension	Definition
X	R*8	NPTI	'X' direction mesh coordinates
Y	R*8	NPTJ	'Y' direction mesh coordinates
REGMSH	I*4	NINTI*NINTJ	Array MC, the mesh interval-region correspondence
REGHHT	R*8	NREG*NIL	Region volumes
NI	I*4	--	Number of 'X' direction mesh intervals NINTI
NR	I*4	--	Number of regions NREG
N1	I*4	--	Data set reference number of GEOM

2. Common Blocks

/GEOM/

Variables used from this common block

Name	Type	Definition
NGEOM	I*4	Geometry type number
NINTI	I*4	Number of 'X' direction intervals
NINTJ	I*4	Number of 'Y' direction intervals
NREG	I*4	Number of regions
NFIN	I*4	Number of finite geometry type
NIL	I*4	Number of integration limits

/CTYPO2/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag -
		0 - data set edit is suppressed
		1 - data set edit is printed

3. Local Variables

Name	Type	Definition
I	I*4	Index on 'X' intervals
J	I*4	Index on 'Y' intervals
N	I*4	Index on finite directions
R	I*4	Index on regions
PI	R*8	3.14159265
FACT	R*8	Common volume factor for mesh interval volumes

4. Functions and tasks performed by this subroutine

Region volumes are computed by summing interval volumes over each region and then multiplying by a common factor and the region height, if any.

First, R = REGMSH(I,J), $\Delta X = X_{(I+1)} - X_{(I)}$, for I = 1, NINTI, and $\Delta Y = Y_{(J+1)} - Y_{(J)}$ for J = 1, NINTJ.

For one-dimensional slab geometry,

$$REGVOL_{(R)} = \sum_{I \varepsilon R} \Delta X_{(I)} \quad .$$

For one-dimensional cylinder geometry,

$$REGVOL_{(R)} = \sum_{I \varepsilon R} (X^2_{(I+1)} - X^2_{(I)}) \quad ,$$

and for one-dimensional sphere geometry,

$$REGVOL_{(R)} = \sum_{I \varepsilon R} (X^3_{(I+1)} - X^3_{(I)}) \quad .$$

For two-dimensional X-Y or X-Y with 45-degree symmetry geometry,

$$REGVOL_{(R)} = \sum_{I,J\varepsilon R} \Delta X_{(I)} \cdot \Delta Y_{(J)} \quad .$$

For two-dimensional R-Z or R-Θ geometry,

$$REGVOL_{(R)} = \sum_{I,J\varepsilon R} (X^2_{(I+1)} - X^2_{(I)}) \cdot \Delta Y_{(J)} \quad ,$$

and for two-dimensional Z-R geometry,

$$REGVOL_{(R)} = \sum_{I,J\varepsilon R} (Y^2_{(J+1)} - Y^2_{(J)}) \cdot \Delta X_{(I)} \quad .$$

These volumes are then multiplied by a factor, FACT, and the region height (where geometry is finite).

For cylindrical, R-Z, Z-R, or cylindrical slab geometries,

FACT = π.

For a sphere, FACT = $\frac{4\pi}{3}$.

For R-Θ geometry, FACT = 0.5.

Then, for one-dimensional finite cylindrical slab geometry,

$REGVOL_{(R)}$ = (REGHHT(R,1)**2)*FACT*$REGVOL_{(R)}$.

For all other finite geometries,

$$REGVOL_{(R)} = \sum_{N=1}^{NIL} FACT*REGVOL_{(R)}*2*REGHHT_{(R,N)} .$$

The volumes are printed, if desired, and record type 10 of data set GEOM is written.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

   DSGEM

7. Error messages generated by this subroutine - None

Subroutine REPORT (NN,*,*)

1. Arguments

Name	Type	Dimension	Definition
NN	I*4	--	Type number of card being processed
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag
IWARN	I*4	Warning flag
IQUIT	I*4	Path termination flag
IGOOF	I*4	Warning counter

3. Local Variables - None

4. Functions and tasks performed by this subroutine

Subroutine REPORT is called from most card reading routines when either flag ISTOP is greater than zero or flag IWARN is greater than zero; it uses the appropriate return system after printing an explanatory message incorporating the offending card type 'nn'.

ISTOP=1 causes flag IQUIT to be set to one. This implies no data set output. The second error return is utilized to continue card processing in subroutine CDSCAN. ISTOP is set to zero again. If ISTOP=2, flag IQUIT set to 1. The routine utilizes the first or immediate return system to effectively terminate module execution by proceeding back through the first error routine structure to the calling program.

ISTOP=3 has the same effect as ISTOP=2, but a different message is printed.

IWARN=1 causes counter IGOOF to be incremented by 1 and a warning statement is printed. IWARN is set back to zero, and a normal return is made to the calling routine.

If REPORT is entered with IQUIT=1, the effect is the same as when ISTOP=2.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

SPACE	TYPE13
TYPE01	TYP13A
TYP01A	TYPE14
TYPE02	TYPE15
TYPE03	TYP15A
TYPE04	TYPE19
TYPE05	TYP19A
TYP05A	TYPE21
TYPE06	TYPE22
TYP06A	TYPE23
TYPE07	TYP23A
TYP07A	TYPE24
TYPE08	TYP24A
TYP08A	TYPE25
TYPE09	TYP25A
TYPE10	TYPE26
TYP10A	TYPE27
TYPE11	TYP27A
TYP11A	TYPE28
TYPE12	TYP28A
TYP12A	

7. Error messages generated by this subroutine

Fatal:

(REPORT) - AN ERROR HAS OCCURRED DURING A CALL TO POINTR WHILE READING CARD TYPE 'ii'. NO ATTEMPT TO READ FURTHER WILL BE MADE.

(REPORT) - AN ERROR OCCURRED WHILE READING CARD TYPE 'ii' WHICH MAKES IT IMPOSSIBLE TO CONTINUE READING - RETURN WILL BE MADE TO THE CALLING PATH. EXECUTION WILL BE TERMINATED.

Serious:

(REPORT) - AN ERROR DETECTED WHILE READING CARD TYPE 'ii' HAS MADE IT IMPOSSIBLE TO WRITE GNIP OUTPUT DATA SETS.

Minor:

(REPORT) - MINOR ERROR NUMBER 'ii' TYPE 'ii' CARD.

Subroutine SPACE (IREFNO, NUMCDS, ITYPNO, *, *)

1. Arguments

Name	Type	Dimension	Definition
IREFNO	I*4	--	Data set reference number (A.NIP)
NUMCDS	I*4	--	Number of remaining cards of type being processed
ITYPNO	I*4	--	Type number being processed
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag 0 - path continues 1 - path terminates

3. Local Variables

Name	Type	Definition
I	I*4	Index on cards being processed
N2	I*4	Index number of card being processed
N3	I*4	Type number on card being processed

4. Functions and tasks performed by this subroutine

SPACE is called by the various subroutines which only require one card of the given type.

It reads NUMCDS in the data set defined by IREFNO, and compares the type number on them with the ITYPNO. These quantities are passed in the argument list.

If there is a discrepancy, a message is printed, ISTOP is set to 3, and processing continues.

Finally if ISTOP is greater than zero, subroutine REPORT is called.

If no errors have been detected, the number and type of cards encountered were exactly correct, the cards have effectively been skipped, and return is made to the calling program.

5. Subprograms called by this subroutine

REPORT

6. Subprograms calling this subroutine

TYPE02	TYPE18
TYPE03	TYPE20
TYPE04	TYPE21
TYPE16	TYPE22
TYPE17	TYPE26

7. Error messages generated by this subroutine

Fatal:

(SPACE) - CARD SUPPOSEDLY OF TYPE 'ii' FOUND TO BE OF TYPE 'ii' IN A.NIP.

(SPACE) - DISCREPANCY IN NUMBER OF TYPE 'ii' CARDS. N2='ii' BUT NUMBER OF CARDS TO BE SPACED PAST = 'ii'

(SPACE) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2='ii'

(SPACE) - READ ERROR. ERROR IN DATA TRANSFER. N2='ii'

Subroutine TEST (N1, RL, XM, I, NP, L, *)

1. Arguments

Name	Type	Dimension	Definition
M1	I*4	--	M1 or M2
RL	R*8	--	RLCOB or RUCOB
XM	R*8	--	XMESH or YMESH
I	I*4	--	Index on number of mesh lines
NP	I*4	--	NPTI or NPTJ
L	I*4	--	Index on mesh line being tested
*	--	--	Error return

2. Common Blocks - None

3. Local Variables

Name	Type	Definition
J	I*4	Index of mesh line array
VAL	R*8	Difference between coordinates being tested

4. Functions and tasks performed by this subroutine

Subroutine TEST is called from subroutine EXTBND to test that the lower and upper coordinates given as endpoints of boundaries are legitimate ones.

The output quantity M1(L) is initialized to zero.

When an 'X' coordinate RLCOB(I) is passed to it, TEST compares this with the whole XMESH(J) array to determine if that coordinate is contained in the array. If it is, M1(L) is redefined as J, the mesh line index number.

If an equality is not found, an error message states that there is no point which corresponds to the input coordinate, and the error return is used.

The test is carried out for 'Y' direction coordinates when these quantities are received through the calling sequence.

Both lower RLCOB input and upper RUCOB input are tested by means of this routine.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

EXTBND

7. Error messages generated by this subroutine

Fatal:

(TEST) - THERE IS NO MESH POINT WHICH CORRESPONDS WITH
INPUT COORDINATE 'xx.xxxxx'

Subroutine TYPE01 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Flag indicator for error return

3. Local Variables

Name	Type	Definition
NC	I*4	Number of type 01 cards
NT	I*4	Number of title array elements
N2	I*4	Number of words requested of POINTR

Name	Type	Definition
DUMMY	R*8	--
ITITLE	I*4	Pointer for array TITLE

4. Functions and tasks performed by this subroutine

This subroutine prepares for reading the number of type 01 cards in A.NIP. Space for the array "TITLE' is requested, and if space is available TYP01A is called to read in the array.

If space is not available, a message is printed giving the amount of full words requested and REPORT is called to provide an immediate return.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR) TYP01A
IPTERR (entry in POINTR) REPORT
PUTPNT (entry in POINTR)

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE01) - ARRAY OVERFLOW, N2='ii'

Subroutine TYP01A (N1, TITLE, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
TITLE	I*4	11*N(1)	Array containing title
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array containing number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Flag for error return

3. Local Variables

Name	Type	Definition
I	I*4	Title array index
J	I*r	Card count index increment
L	I*4	Card count index
J1	I*4	Lower index for title card
J2	I*4	Upper index of title card
NC	I*4	Number of type 01 cards
N2	I*4	Number of card type 01 being read
N3	I*4	Card type number field

4. Functions and tasks performed by this subroutine

This subroutine is called from TYPE01 to read in the title cards supplied in A.NIP.

If any read error is encountered, ISTOP is set to 1 and subroutine REPORT is called. This gives an error return, which in this case is the minor one since title information is not vital to execution. Further reading will be attempted.

N2 in the error messages represents the Nth card of type 01.

5. Subprograms called by this subroutine

REPORT

6. Subprograms calling this subroutine

TYPE01

7. Error messages generated by this subroutine

Fatal:

(TYP01A) - CARD SUPPOSEDLY OF TYPE 01 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP01A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED, N2='ii'

(TYP01A) - READ ERROR. ERROR IN DATA TRANSFER.

Subroutine TYPE02 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/LOCATE/

Variables used from this common block

Name	Type	Definition
IPRINT	I*4	Edit flag for POINTR

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Flag indicator for error return

/CTYP02/

Variables used from this common block

Name	Type	Definition
IOFLAG	I*4	Flag -
		0 - data set edit is suppressed
		1 - data set edit is printed

3. Local Variables

Name	Type	Definition
N2	I*4	Sequence number of card type 02 being read; also N(2) - 1
N3	I*4	Type number of card being read

4. Functions and tasks performed by this subroutine

One card of type 02 is read if there are any present, and any others are skipped by a call to SPACE.

The quantity IPRINT is read to be used by POINTR in editing its output. The printed output flag, IOFLAG, is also read in this routine.

If any read errors occur, ISTOP is set to 1 and subroutine REPORT is called. In this instance the second error return is used.

5. Subprograms called by this subroutine

SPACE
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE02) - CARD SUPPOSEDLY OF TYPE 02 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYPE02) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED, N2='ii'

(TYPE02) - READ ERROR. ERROR IN DATA TRANSFER, N2='ii'

Subroutine TYPE03 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*r	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP03/

Variables used from this common block

Name	Type	Definition
ALPHA	R*8	Initial alpha or fixed alpha
IGTYPE	I*4	Geometry type
IPTYPE	I*4	Problem type

Name	Type	Definition
FIXEDK	R*8	Fixed K-effective
ICTYPE	I*4	Calculation type

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Flag for error return

3. Local Variables

Name	Type	Definition
N2	I*4	Number of type 3 card being read; also N(2) - 1
N3	I*4	Card type number being read

4. Functions and tasks performed by this subroutine

TYPE03 is always called by CDSCAN. It checks for the presence of type 03 cards; if none are present, it returns to CDSCAN.

If type 03 cards are present only one is read; the rest are skipped by a call to SPACE.

The quantities read are

IGTYPE, geometry type (may be 10, 11, 12, 13, 20, 21, 30, 40, 41, 42, 43, 50, 51, 60, 61);

IPTYPE, problem type (may be 0, 1, 2 or 3);

ICTYPE, calculation type (may be 0 or 1);

FIXEDK, k_{fixed} for source calculation (will be set to 0.0 when no value is supplied); and

ALPHA, initial alpha for an alpha search or fixed alpha.

If a serious error occurs, ISTOP is set to 3 and the immediate error return is used.

Invalid input for a variable will cause the second error return to be used. This will cause a return to the next card type read.

5. Subprograms called by this subroutine

SPACE
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE03) - CARD SUPPOSEDLY OF TYPE 03 FOUND TO BE OF TYPE 'ii' IN A.NIP

(TYPE03) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED, N2='ii'

(TYPE03) - READ ERROR. ERROR IN DATA TRANSFER. N2='ii'

Serious:

(TYPE03) - NO TYPE 03 CARD SUPPLIED IN A.NIP.

(TYPE03) - INVALID GEOMETRY TYPE OF 'ii' GIVEN ON FIRST 03 CARDS IN A.NIP.

(TYPE03) - INVALID CALCULATION TYPE OF 'ii' GIVEN ON FIRST 03 CARD IN A.NIP.

(TYPE03) - INVALID PROBLEM TYPE OF 'ii' GIVEN ON FIRST 03 CARD IN A.NIP.

(TYPE03) - INVALID PROBLEM TYPE OF 2 GIVEN FOR A CALCULATION TYPE OF 1 ON FIRST 03 CARD IN A.NIP.

Subroutine TYPE04 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP03/

Variables used from this common block

Name	Type	Definition
IGTYPE	I*4	Geometry type

/CTYP04/

Variables used from this common block

Name	Type	Definition
IEBC	I*4	External boundary condition number
NIEBC	I*4	Number of external boundary condition numbers given on type 04 card

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of boundaries
N2	I*4	Number of type 04 card being read; also N(4) - 1
N3	I*4	Type number of card being read
N4	I*4	Number of boundary condition types on the 04 card
N6	I*4	Number of diffusion-type numbers on the 04 card
I2	I*4	Dummy sum

4. Functions and tasks performed by this subroutine

TYPE04 is always called by CDSCAN. If no 04 cards are supplied in A.NIP, an error message stating this is written and the error flag ISTOP is set to 3. A call to REPORT is issued and the first error return is used to return control to the path and stop execution.

If 04 cards do exist, only one will be read; the rest are skipped by a call to SPACE.

A serious error causes ISTOP to be set to 3 and an appropriate message to be printed. A call to REPORT is made and the first error return is used to cause immediate return.

The following conditions are checked on the boundary condition types, IEBC(I).

All must be less than or equal to 12. All must be of either transport or diffusion type.

X-Y type geometries (40,41,42,43), may not have IEBC(I)=12, i.e. white boundaries.

R-Z geometry (50) may only have IEBC(2)=12, i.e. only the right-hand boundary may be white.

R-Z geometry may not have both left and right side periodic, i.e. IEBC(1)=11 and IEBC(2)=11.

The only way periodic boundaries may occur is on opposite sides of the reactor:

IEBC(1)=IEBC(2)=11, other than in R-Z type,

or IEBC(3)=IEBC(4)=11.

If these conditions are not satisfied, an appropriate message is printed.

Errors in data values will not stop processing but ISTOP is set to 1, and the second error return is used after the call to REPORT.

5. Subprograms called by this subroutine

SPACE
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE04) - NO TYPE 04 CARD SUPPLIED IN A.NIP.

(TYPE04) - CARD SUPPOSEDLY OF TYPE 04 IN A.NIP FOUND TO BE OF TYPE 'ii'.

(TYPE04) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2='ii'.

(TYPE04) - READ ERROR. ERROR IN DATA TRANSFER. N2='ii'.

Serious:

(TYPE04) - INVALID BOUNDARY CONDITION OF 'ii' GIVEN ON THE FIRST 04 CARD IN A.NIP.

(TYPE04) - PERIODIC BOUNDARY CONDITIONS MUST OCCUR IN PAIRS ON OPPOSITE SIDES ON THE REACTOR. SEE BOUNDARY CONDITION VALUES 'ii' AND 'ii' ON TYPE 04 IN A.NIP.

(TYPE04) - X-Y GEOMETRIES DO NOT ALLOW WHITE BOUNDARY CONDITIONS - CARD TYPE 04 IN A.NIP.

(TYPE04) - BOUNDARY CONDITION TYPE NUMBERS ARE NOT ALL EITHER TRANSPORT OR DIFFUSION TYPE.

(TYPE04) - IN R-Z GEOMETRY, A WHITE BOUNDARY IS ONLY ALLOWED ON THE RIGHT HAND SIDE.

(TYPE04) - LEFT AND RIGHT BOUNDARIES CANNOT BE PERIODIC IN R-Z GEOMETRY.

Subroutine TYPE05 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Flag for error return

/CTYP05/

Variables used from this common block

Name	Type	Definition
NLX	I*4	Number of lower 'X' direction external boundary input cards

Name	Type	Definition
NUX	I*4	Number of upper 'X' external boundary input cards
NLY	I*4	Number of lower 'Y' external boundary input cards
NUY	I*4	Number of upper 'Y' external boundary input cards

3. Local Variables

Name	Type	Definition
N2	I*4	Number of full words needed for POINTR array requests
IBD	I*4	Pointer for array BD
DUMMY	R*8	Dummy
IEBCA	I*4	Pointer for array EBCA
IEBCB	I*4	Pointer for array EBCB
IIHEGN	I*4	Pointer for array IHEGN
IILEGN	I*4	Pointer for array ILEGN
IRLCOB	I*4	Pointer for array RLCOB
IRUCOB	I*4	Pointer for array RUCOB

4. Functions and tasks performed by this subroutine

CTYPE05 is always called by CDSCAN. It defines the quantities NLY, NUX, NLY, and NUY as zero. These represent the number of input cards to each boundary. When no type 05 cards are encountered in A.NIP the subroutine returns immediately to CDSCAN. Otherwise, it defines POINTR arrays used by subroutine TYP05A, BD, EBCA, EBCB, IHEGN, ILEGN, RLCOB, and RUCOB.

If an error occurs during a request for space, the number of words requested by the subroutine is printed, ISTOP is set to 2 and REPORT is called. This will then use the first or fatal error return.

After a successful space request, the pointers are requested and a call to TYP05A is made to do the actual reading of the type 05 data.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR) TYP05A
IPTERR (entry in POINTR) REPORT
PUTPNT (entry in POINTR)

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYP05) - ARRAY OVERFLOW. N2='ii'

Subroutine TYP05A (N1, BD, EBCA, EBCB, IHEGN, ILEGN, RLCOB, RUCOB, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
BD	R*8	N(5)	Array of boundary names
EBCA	R*8	N(5)	First constant input to boundary
EBCB	R*8	N(5)	Second constant input to boundary
IHEGN	I*4	N(5)	Highest energy group number affected by constant
ILEGN	I*4	N(5)	Lowest energy group number affected by constant
RLCOB	R*8	N(5)	Lower coordinate at end of given boundary
RUCOB	R*8	N(5)	Upper coordinate at end of given boundary
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

/CTYP04/

Variables used from this common block

Name	Type	Definition
NIEBC	I*4	Number of external boundary condition numbers given on card type 04

/CTYP05/

Variables used from this common block

Name	Type	Definition
NLX	I*4	Number of input cards to lower 'X' external boundary
NUX	I*4	Number of input cards to upper 'X' external boundary
NLY	I*4	Number of input cards to lower 'Y' external boundary
NUY	I*4	Number of input cards to upper 'Y' external boundary

3. Local Variables

Name	Type	Definition
I	I*4	Index of type 05 cards
J	I*4	Index on external boundary conditions from type 04 card
NN	I*4	NIEBC number of external boundaries on type 04 card
N2	I*4	Number of card type 05 being read
N3	I*4	Number of type 05 cards in A.NIP
N4	I*4	Type number of card being read
XL	R*8	'LOWER X' boundary name
XU	R*8	'UPPER X' boundary name
YL	R*8	'LOWER Y' boundary name
YU	R*8	'UPPER Y' boundary name
XBD	R*8	Equivalent for the 4 alphanumeric boundary names
XLOW	R*8	Name of alphanumeric data 'LOWER'
XHIGH	R*8	Name of alphanumeric data 'HIGHER'

4. Functions and tasks performed by this subroutine

TYP05A is called from subroutine TYPE05. It reads each type 05 card and checks the data for validity. It counts the number of input cards to each external boundary, redefining quantities NLX, NUX, NLY, NUY.

If a fatal error occurs at the time the card is read, a message is printed, ISTOP is set to 3 and subroutine REPORT is called. This then uses the first error return.

The quantities read into arrays from each card are

BD(I), the boundary name (may be 'XL', 'XU', 'YL', 'YU');

EBCA(I), the value of constant a;

EBCB(I), the value of constant b;

IHEGN(I), higher-energy group for which constants apply (must be greater than or equal to zero);

ILEGN(I), lower-energy group number for which constants apply (must be greater than or equal to zero, and must be greater than or equal to IHEGN(I));

RLCOB(I), the lower coordinate on the boundary for which the constants apply (must be greater than or equal to zero, and less than RUCOB(I) when RLCOB(I) is greater than zero. Both RLCOB(I) and RUCOB(I) may be zero); and

RUCOB(I), the upper coordinate on the boundary for which constants apply.

Any divergence from these standards will cause flag ISTOP to be set to one and subroutine REPORT to be called. The second error return is used.

5. Subprograms called by this subroutine

REPORT

6. Subprograms calling this subroutine

TYPE05

7. Error messages generated by this subroutine

Fatal:

(TYP05A) - CARD SUPPOSEDLY OF TYPE 05 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP05A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2='ii'

(TYP05A) - READ ERROR. ERROR IN DATA TRANSFER. N2='ii'

Serious:

(TYP05A) - INVALID BOUNDARY DESIGNATOR OF 'ii' GIVEN ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - NEGATIVE 'ii'-ENERGY GROUP NUMBER OF 'ii' GIVEN ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - THE LOWER ENERGY GROUP NUMBER IS LESS THAN THE HIGHER ENERGY GROUP NUMBER ON THE 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - NEGATIVE 'aaaaaa' COORDINATE OF 'ii' GIVEN ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - LOWER COORDINATE OF 'x.x' NOT LESS THAN HIGHER COORDINATE OF 'x.x' ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - IN A ONE DIMENSIONAL PROBLEM, Y BOUNDARIES ARE INCORRECTLY USED - CARD OF TYPE 05 - NUMBER 'ii' IN A.NIP.

Subroutine TYPE06 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Flag for error return

3. Local Variables

Name	Type	Definition
N2	I*4	Number of full words requested in POINTR array
IXL	I*4	Pointer for array XL
IXU	I*4	Pointer for array XU

Name	Type	Definition
IYL	I*4	Pointer for array YL
IYU	I*4	Pointer for array YU
DUMMY	R*8	Dummy
ILABEL	I*4	Pointer for array RLABEL
ININTX	I*4	Pointer for array NINTX
ININTY	I*4	Pointer for array NINTY

4. Functions and tasks performed by this subroutine

TYPE06 is always called by CDSCAN. Its purpose is to request space for the arrays which are read by TYP06A, RLABEL, XL, XU, NINTX, NINTY, YL and YU, and then to call subroutine TYP06A.

If no type 06 cards exist in A.NIP, an attempt is made to define the arrays. The error flag ISTOP is set to 1. Any error in the request for array space will cause ISTOP to be set to 2, the requested amount will be printed, REPORT will be called, and the first error return will be used. If no array error is encountered, the fact that ISTOP is equal to 1 will invoke a call to REPORT, with the second error return being used.

When there are type 06 cards in A.NIP, after a successful request for array space, the pointers for the arrays are requested and passed to subroutine TYP06A, where the cards are read.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP06A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE06) - ARRAY OVERFLOW. N2 = 'ii'.

Serious:

(TYPE06) - NO TYPE 06 CARDS SUPPLIED IN A.NIP.

Subroutine TYP06A (N1, RLABEL, XL, XU, NINTX, NINTY, YL, YU, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
RLABEL	R*8	N(6)	Array of region names given on type 06 cards
XL	R*8	N(6)	Lower 'X' coordinate defining left side of region
XU	R*8	N(6)	Upper 'X' coordinate defining right side of region
NINTX	I*4	N(6)	Number of intervals inserted in 'X' direction
NINTY	I*4	N(6)	Number of intervals inserted in 'Y' direction
YL	R*8	N(6)	Lower 'Y' coordinate defining bottom of region
YU	R*8	N(6)	Upper 'Y' coordinate defining top of region
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index of card type 06 being processed
N2	I*4	Number of card type 06 being read
N3	I*4	Number of type 06 cards in A.NIP
N4	I*4	Type number of card being read
BLANKS	R*8	Alphameric name for blank space

4. Functions and tasks performed by this subroutine

This routine is called from TYPE06 when there are type 06 cards in A.NIP. If a fatal error occurs while reading a card, ISTOP is set to 3, the appropriate message is printed and a call to REPORT is issued. The first error return is used.

The quantities read from the card into arrays and tested are the following:

RLABEL(I), region label (must be non-blank);

XL(I), 'X'-direction lower region boundary coordinate (must be greater than or equal to zero, but must also be less than XU(I));

XU(I), 'X'-direction upper region boundary coordinate;

NINTX(I), number of equal intervals to be inserted between the given XL(I) and XU(I);

NINTY(I), number of equal intervals to be inserted between the given YL(I) and YU(I) (may be blank);

YL(I), 'Y'-direction lower region boundary coordinate (must be greater than or equal to zero, and less than YU(I) when YU(I) is greater than zero); and

YU(I), 'Y'-direction upper region boundary coordinate (may be zero for one-dimensional problems).

When one of the above conditions is not satisfied by the input data, ISTOP is set to 1. This causes a call to REPORT after all cards are read and the second error return is used.

5. Subprograms called by this subroutine

SQUEZE
REPORT

6. Subprograms calling this subroutine

TYPE06

7. Error messages generated by this subroutine

Fatal:

(TYP06A) - CARD SUPPOSEDLY ON TYPE 06 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP06A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP06A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP06A) - BLANK REGION LABEL ON 'ii'-TH 06 CARD IN A.NIP.

(TYP06A) - INVALID X-COORDINATE REGION BOUNDARIES ON 'ii'-TH 06 CARD IN A.NIP.

(TYP06A) - NEGATIVE NUMBER OF INTERVALS GIVEN ON 'ii'-TH 06 CARD IN A.NIP.

(TYP06A) - INVALID Y-COORDINATE REGION BOUNDARIES ON 'ii'-TH 06 CARD IN A.NIP.

Subroutine TYPE07 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of array words requested
DUMMY	R*8	Dummy
IREGN	I*4	Pointer for array AREGN
ILABEL	I*4	Pointer for array ALABEL
JLABEL	I*4	Pointer for array RLABEL

4. Functions and tasks performed by this subroutine

TYPE07 is called from CDSCAN when there are type 07 cards present in A.NIP. Its purpose is to request space for the arrays ALABEL and AREGN, representing area names and area-region correspondence. Then it obtains the pointers for them and array RLABEL and passes them to subroutine TYP07A, where the data is read.

An error in the request to POINTR for space will result in ISTOP being set to 3. A message giving the number of words of array space requested is printed and subroutine REPORT is called. The first error return is used.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP07A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE07) - ARRAY OVERFLOW. N2 = 'ii'

Subroutine TYP07A (N1, RLABEL, ALABEL, AREGN, *, *)

1. Arguments

Name	Type	Dimension	Description
N1	I*4	--	A.NIP data set reference number
RLABEL	R*8	N(6)	Region label
ALABEL	R*8	N(7)	Area label
AREGN	R*8	10 * N(7)	Labels of regions contained in area

Name	Type	Dimension	Definition
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index of number of type 07 cards
J	I*4	Increment of index on region-area correspondence, AREGN
K	I*4	Index on region labels, KLABEL
N2	I*4	Number of card type 07 being read
N3	I*4	Number of type 07 cards existing in A.NIP
N4	I*4	Card type number on card currently being read
N5	I*4	10 * (I-1)
N6	I*4	Number of type 06 cards
N7	I*4	Index equal to N5 + J on AREGN
BLANKS	R*8	Blanks

4. Functions and tasks performed by this subroutine

TYP07A is called by TYPE07 to read the area specifications used in preparing the GEOM data set.

From each card is read an area label and up to 10 region labels. If a serious error occurs while reading, the number of the type 07 card being processed is printed in an appropriate message. The flag ISTOP is set equal to 3 and subroutine REPORT is called.

If no error occurs during reading, imbedded and leftmost blanks are removed from the labels and a check is made to see that the following conditions are met by the data.

ALABEL, the area label, must not be blank.

AREGN, the region label of this portion of an area, must match one of the region labels in RLABEL or must be blank.

When these conditions do not prevail, ISTOP is set to 1 after all type 07 cards are read and subroutine REPORT is called.

The second error return is then used.

5. Subprograms called by this subroutine

SQUEZE (system module)
REPORT

6. Subprograms calling this subroutine

TYPE07

7. Error messages generated by this subroutine

Fatal:

(TYP07A) - CARD SUPPOSEDLY ON TYPE 07 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP07A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP07A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

TYP07A) - BLANK AREA LABEL ON 'ii'-TH 07 CARD IN A.NIP.

TYP07A) - REGION LABEL OF 'aaaaaa' GIVEN ON 'ii'-TH TYPE 07 CARD IN A.NIP. DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD.

Subroutine TYPE08 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of words of array space requested
IZXL	I*4	Pointer for array ZXL
IZXU	I*4	Pointer for array ZXU

Name	Type	Definition
IZYL	I*4	Pointer for array ZYL
IZYU	I*4	Pointer for array ZYU
DUMMY	R*8	Dummy
ILABEL	I*4	Pointer for array ZLABEL

4. Functions and tasks performed by this subroutine

Subroutine TYPE08 is called by CDSCAN when type 08 cards exist in A.NIP. It requests space for the arrays ZLABEL, ZXL, ZXU, ZYL, and ZYU which are read from type 08 cards. If space is not available a message stating this is printed, ISTOP is set to 2, and REPORT is called. This returns via the first error return.

When space is available, the pointers are requested for the arrays and passed to subroutine TYP08A.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP08A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE08) - ARRAY OVERFLOW. N2 = 'ii'

Subroutine TYP08A (N1, ZLABEL, ZXL, ZXU, ZYL, ZYU, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
ZLABEL	R*8	N(8)	Zone label
ZXL	R*8	N(8)	Coordinate of lower 'X' boundary of section of zone
ZXU	R*8	N(8)	Coordinate of upper 'X' boundary of section of zone
ZYL	R*8	N(8)	Coordinate of lower 'Y' boundary of section of zone
ZYU	R*8	N(8)	Coordinate of upper 'Y' boundary of section of zone
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 08 card number
N2	I*4	Number of card type 08 being read
N3	I*4	Number of card type 08 cards in A.NIP

Name	Type	Definition
N4	I*4	Type number of card currently being read
BLANKS	R*8	Alphameric blanks

4. Functions and tasks performed by this subroutine

TYP08A is called by subroutine TYPE08. It reads the zone definitions used for data set GEOM.

From each card are obtained values for the arrays ZLABEL, ZXL, ZXU, ZYL, and ZYU. Any read error will cause ISTOP to be equal to 3 and the appropriate message to be printed. A call is issued to subroutine REPORT and the first error return is used.

When no read error occurs, a check for data confirmity is performed.

ZLABEL(I), the zone label must not be blank.

ZXL(I), must be greater than or equal to zero and must be less than ZXU(I) when ZXU(I) does not equal zero.

Any departure from these specifications causes ISTOP to be set to 1 and, after all type 08 cards have been processed, a call to REPORT to be made.

The second error return is then used.

If no specification error is found, the labels are left-justified and imbedded blanks are removed.

5. Subprograms called by this subroutine

SQUEZE (system module)
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYP08A) - CARD SUPPOSEDLY OF TYPE 08 FOUND TO BE OF TYPE 'ii' IN A.NIP

(TYP08A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP08A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP08A) - BLANK ZONE LABEL ON 'ii'-TH 08 CARD IN A.NIP.

(TYP08A) - INVALID X-COORDINATE ZONE BOUNDARY ON 'ii'-TH 08 CARD IN A.NIP.

(TYP08A) - INVALID Y-COORDINATE ZONE BOUNDARY ON 'ii'-TH 08 CARD IN A.NIP.

Subroutine TYPE09 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	Data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/CTYP09/

Variables used from this common block

Name	Type	Definition
NVX	I*4	Number of variable-mesh 'X' direction input quantities
NVY	I*4	Number of variable-mesh 'Y' direction input quantities

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of words requested in POINTR array
IUCX	I*4	Pointer to array UCX
IUCY	I*4	Pointer to array UCY
DUMMY	R*8	Dummy
ININVX	I*4	Pointer to array NINVX
ININVY	I*4	Pointer to array NINVY

4. Functions and tasks performed by this subroutine

TYPE09 is called from CDSCAN when there are type 09 cards present in A.NIP.

Its purpose is to request space for the variable mesh structure input arrays and to pass pointers for these arrays to subroutine TYP09A, which will read data into them.

If a request for space to POINTR is not successful, ISTOP is set to 2, and after printing a message TYPE09 calls REPORT. This returns through the first error return.

When space is available in the container array, the pointers to the arrays are extracted and passed to subroutine TYP09A. After a normal return, any unused arrays are wiped out.

When all arrays are used, NINVX and UCX are redefined to length NVX and arrays NINVY and UCY are redefined to length NVY by calls to REDEF.

5. Subprograms called by this subroutine

REDEF (entry in POINTR)
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP09A
WIPOUT
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE09) - ARRAY OVERFLOW. N2 = 'ii'

Subroutine TYP09A (N1, NINVX, UCX, NINVY, UCY, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
NINVX	I*4	NVX	Number of intervals to be inserted in 'X' direction
UCX	R*8	NVX	Upper 'X' coordinate of space being defined
NINVY	I*4	NVY	Number of intervals to be inserted in 'Y' direction
UCY	R*8	NVY	Upper 'Y' coordinate of space being defined
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYPO9/

Variables used from this common block

Name	Type	Definition
NVX	I*4	Number of input quantities in 'X' direction
NVY	I*4	Number of input quantities in 'Y' direction

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of type 09 card being processed
J	I*4	Index on position on card of data being processed
X	R*8	Representation of alphameric data 'X'
Y	R*8	Representation of alphameric data 'Y'
N2	I*4	Index number of card type 09 being processed
N3	I*4	Number of type 09 cards in A.NIP
N4	I*4	Type number on card being processed
UC(3)	R*8	Input upper coordinate of distance being divided

Name	Type	Definition
NINV(3)	I*4	Input number of intervals to be defined between coordinates
NNVX	I*4	Representation of data 'NVX'
NNVY	I*4	Representation of data 'NVY'
COORD	R*8	Input direction name of input coordinates and intervals

4. Functions and tasks performed by this subroutine

TYP09A is called by TYPE09. It performs the reading of the type 09 cards and separates 'X' direction input from 'Y' direction input.

A coordinate direction, either 'X' or 'Y', is read from each card, and up to three pairs of the values NINV(J) and UC(J).

A fatal reading error causes an explanatory message to be printed and indicator ISTOP to be set to 3. Subroutine REPORT is then called and the first error return is employed.

Each card is read and the quantity COORD(I) is tested to determine the input direction, which must be 'X' or 'Y'. The rest of the input on the card is placed in appropriate X or Y arrays: that is, if 'X' is the value of COORD(I), then NINV(I) and UC(I) will become NINVX (NVX+J) and UCX (NVX+J), where J indicates the increment per card (J=1,2,3). As each 'X' card is encountered, NVX increases by 3.

A similar process distributes data from the 'Y' direction cards into the arrays NINVY (NVY+J) and UCY (NVY+J).

These data arrays are then checked for conformity to certain specifications.

If there is 'X' direction input, UCX(1) must be greater than zero, NINVX(1) must be greater than zero for I = 2, NVX, NINVX(I) must be greater than zero, and UCX(I) must be greater than UCX(I-1).

If both NINVX(I) and UCX(I) are zero it indicates that no data are present and is not an error.

The same checks are made for the 'Y' direction data arrays.

Any deviations cause the flag ISTOP to be set to 1 and an explanatory message to be printed. After all processing, if ISTOP is greater than zero, REPORT is called and use is made of the second error return.

5. Subprograms called by this subroutine

REPORT

6. Subprograms calling this subroutine

TYPE09

7. Error messages generated by this subroutine

Fatal:

(TYP09A) - CARD SUPPOSEDLY OF TYPE 09 TO BE OF TYPE 'ii' IN A.NIP.

(TYP09A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP09A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP09A) - INVALID COORDINATE OF 'a' GIVEN ON 'ii'-TH 09 CARD IN A.NIP.

(TYP09A) - DISCREPANCY IN NUMBER OF {X/Y}-DIRECTION 09 CARDS IN A.NIP. N(9) = 'ii' BUT {NIX/NIY} = 'ii'

(TYP09A) - THE ZERO OR NEGATIVE VALUE 'x.x' WAS GIVEN FOR THE FIRST LOWER COORDINATE FOR THE {X/Y}-DIRECTION ON A 09 CARD IN A.NIP.

(TYP09A) - THE ZERO OR NEGATIVE VALUE 'ii' WAS GIVEN AS THE FIRST NUMBER OF INTERVALS ON {X/Y}-DIRECTION 09 CARD IN A.NIP.

(TYP09A) - THE NEGATIVE VALUE 'ii' WAS GIVEN AS A NUMBER OF INTERVALS FOR THE {X/Y}-DIRECTION ON A 09 CARD IN A.NIP.

(TYP09A) - THE UPPER COORDINATES FOR THE {X/Y}-DIRECTION ON THE 09 CARDS IN A.NIP ARE NOT MONOTONICALLY INCREASING. THE INCORRECT VALUE IS 'x.x'

(TYP09A) - TWO SUCCESSIVE UPPER COORDINATES FOR THE {X/Y}-DIRECTION ON THE 09 CARDS IN A.NIP ARE EQUAL. THE INCORRECT VALUE IS 'x.x'

Subroutine TYPE10 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/CTYP10/

Variables used from this common block

Name	Type	Definition
NIX	I*4	Number of type 10 'X' direction internal boundary input
NIY	I*4	Number of type 10 'Y' direction internal boundary input

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Array space requested in full words
DUMMY	R*8	Dummy
IXLEI	I*4	Pointer to array XLEI
IXUEI	I*4	Pointer to array XUEI
IXYIC	I*4	Pointer to array XYIC
IYLEI	I*4	Pointer to array YLEI
IYUEI	I*4	Pointer to array YUEI
IYXIC	I*4	Pointer to array YXIC
IICONX	I*4	Pointer to array ICONX
IICONY	I*4	Pointer to array ICONY
ILABEL	I*4	Pointer to array XILABL
JLABEL	I*4	Pointer to array YILABL

4. Functions and tasks performed by this subroutine

TYPE10 is called by CDSCAN when there are type 10 cards in A.NIP.

It requests space for the input arrays associated with data set BC and passes pointers for these arrays to subroutine TYP10A.

Arrays defined are XILABL, ICONX, XYIC, XLEI, XUEI, YILABL, ICONY, YXIC, YLEI, and YUEI. If any error occurs during the request for space, a message giving the total number of words requested is printed and ISTOP is set equal to 2. A call is issued to subroutine REPORT and the first error return is used.

When no POINTR error occurs, the pointers for the arrays are extracted and passed in the call to TYP10A. After a normal return the unused arrays are wiped out. In the case where all arrays are used, the 'X' direction arrays are redefined to length NIX and the 'Y' direction arrays redefined to the length NIY.

5. Subprograms called by this subroutine

REDEF (entry in POINTR)
GETPNT (entry in POINTR)

IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP10A
WIPOUT
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYP10) - ARRAY OVERFLOW, N2 = 'ii'

Subroutine TYP10A (N1, XILABL, ICONX, XYIC, XLEI, XUEI, YILABL, ICONY, YXIC, YLEI, YUEI, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
XILABL	R*8	NIX	Label of 'X' direction internal interface
ICONX	I*4	NIX	Internal condition on 'X' interface
XYIC	R*8	NIX	'Y' coordinate along which 'X' direction interface lies
XLEI	R*8	NIX	Lower coordinate of end of 'X' interface
XUEI	R*8	NIX	Upper coordinate of end of 'X' interface
YILABL	R*8	NIY	Label of 'Y' direction internal interface
ICONY	I*4	NIY	Internal condition on 'Y' interface
YXIC	R*8	NIY	'X' coordinate along which 'Y' direction interface lies

Name	Type	Dimension	Definition
YLEI	R*8	NIY	Lower coordinate of end of 'Y' interface
YUEI	R*8	NIY	Upper coordinate of end of 'Y' interface
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP10/

Variables used from this common block

Name	Type	Definition
NIX	I*4	Number of type 10 'X' direction internal boundary input
NIY	I*4	Number of type 10 'Y' direction internal boundary input

/CTYP03/

Variables used from this common block

Name	Type	Definition
IGTYPE	I*4	Geometry type

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 10 card
J	I*4	Index on 'X' direction input; also index on 'Y' direction input
X	R*8	Representation of data name 'X'
Y	R*8	Representation of data name 'Y'
Z	R*8	Coordinate along which boundary lies
N2	I*4	Index number of card type 10 being processed
N3	I*4	N(10)
N4	I*4	Card type number on card being read in A.NIP
ICON	I*4	Input internal condition on interface
ZLEI	R*8	Input coordinate at lower end of internal interface
ZUEI	R*8	Input coordinate at upper end of internal interface
COORD	R*8	Input coordinate direction
BLANKS	R*8	Name for alphameric blanks
ZLABEL	R*8	Input label of internal interface

4. Functions and tasks performed by this subroutine

TYP10A is called from subroutine TYPE10. It takes the data input and separates 'X' direction input from 'Y' direction input, filling the proper arrays and counting the input to each coordinate direction.

A fatal error in card processing causes an explanatory message to be printed and flag ISTOP to be set to 3. Subroutine REPORT is then called and the first error return is used.

If the card in A.NIP is read properly the data is then checked for agreement with the following.

ZLABEL, the interface label, must not be blank and must be different from any other interface label.

ICON, the internal condition number, must lie in the range from 0 to 6, inclusive.

COORD, the coordinate direction, must be 'X' or 'Y'.

Z, the coordinate of the axis along which the interface lies, must be non-negative.

ZLEI, the lower end coordinate of the interface, must be non-negative and less than ZUEI, the upper end coordinate of the interface, when ZUEI is greater than zero.

ZUEI, must be non-negative.

If COORD equals 'X', IGTYPE must be 30 or greater, for an 'X' direction internal interface implies a two-dimensional geometry.

Any deviation from these standards causes ISTOP to be set to 1 but processing continues.

At this point the data from cards having COORD = 'X' are placed in the following arrays. The count of NIX is incremented by 1. XILABL (NIX) = ZLABEL, ICONX (NIX) = ICON, XYIC (NIX) = Z, XLEI (NIX) = ZLEI, and XUEI (NIX) = ZUEI.

Similarly, cards having COORD = 'Y' are defined using 'Y'-direction arrays.

If ISTOP has been set to 1 a message will have been printed at the time the error occurred. Subroutine REPORT is now called and the second error return is used.

5. Subprograms called by this subroutine

SQUEZE (system module)
REPORT

6. Subprograms calling this subroutine

TYPE10

7. Error messages generated by this subroutine

Fatal:

(TYP10A) - CARD SUPPOSEDLY OF TYPE 10 FOUND TO BE TYPE 'ii' IN A.NIP.

(TYP10A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP10A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP10A) - BLANK LABEL GIVEN ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - INVALID COORDINATE DIRECTION OF 'a' GIVEN ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - INVALID INTERNAL CONDITION OF 'ii' GIVEN ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - NEGATIVE VALUE OF 'x.x' GIVEN IN COLUMNS 25-36 ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - NEGATIVE VALUE OF 'x.x' GIVEN IN COLUMNS 37-48 on 'ii'-TH CARD IN A.NIP.

(TYP10A) - LOWER INTERFACE COORDINATE NOT LESS THAN UPPER INTERFACE COORDINATE ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - THE SAME INTERNAL INTERFACE LABEL, 'aaaaaa' WAS USED ON MORE THAN ONE TYPE 10 CARD IN A.NIP.

(TYP10A) - X COORDINATE DIRECTION INPUT INDICATES A TWO-DIMENSIONAL PROBLEM. GEOMETRY HAS BEEN STATED AS ONE-DIMENSIONAL. CHECK CARD NUMBER 'ii' OF TYPE 10 IN A.NIP.

Subroutine TYPE11 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of words of space requested for arrays
DUMMY	R*8	Dummy
IBCA1	I*4	Pointer to array BCA1
IBCA2	I*4	Pointer to array BCA2
IBCB1	I*4	Pointer to array BCB1
IBCB2	I*4	Pointer to array BCB2
IIHBGN	I*4	Pointer to array IBHEGN
IILBGN	I*4	Pointer to array IBLEGN
ILABEL	I*4	Pointer to array ULABEL
JLABEL	I*4	Pointer to array XILABL
KLABEL	I*4	Pointer to array YILABL

4. Functions and tasks performed by this subroutine

TYPE11 is called from CDSCAN after type 10 cards have been successfully processed. It sets up arrays for the constants input to the internal interfaces named on the type 10 cards.

If there are no type 11 cards an error message is printed and ISTOP is set to 1. Subroutine REPORT is called and use is made of the first

error return.

When type 11 cards exist, a request for array space is made for the arrays used in preparing data set BC; ULABEL, BCA1, BCB1, BCA2, BCB2, IBHEGN, and IBLEGN. When space has been successfully allocated, the pointers to the needed arrays are extracted and passed in a call to TYP11A.

If there is an error detected after the request for space, ISTOP is set to 2 and the message printed indicates the total space asked for. Subroutine REPORT is called and use is made of the first error return.

5. Subprograms called by this subroutine

SPACE
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP11A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE11) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE11) - TYPE 10 CARDS SUPPLIED IN A.NIP BUT NO TYPE 11 CARDS SUPPLIED.

Subroutine TYP11A (N1, ULABEL, XILABL, YILABL, BCA1, BCB1, BCA2, BCB2, IBHEGN, IBLEGN, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
ULABEL	R*8	N(11)	Internal interface label

Name	Type	Dimension	Definition
XILABL	R*8	NIX	'X' direction internal interface label
YILABL	R*8	NIY	'Y' direction internal interface label
BCA1	R*8	N(11)	First input constant to internal interface
BCB1	R*8	N(11)	Second input constant to internal interface
BCA2	R*8	N(11)	Third input constant to internal interface
BCB2	R*8	N(11)	Fourth input constant to internal interface
IBHEGN	I*4	N(11)	Higher-energy group number for which constants apply
IBLEGN	I*4	N(11)	Lower-energy group number for which constants apply
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP03/

Variables used from this common block

Name	Type	Definition
IGTYPE	I*4	Geometry type

/CTYP10/

Variables used from this common block

Name	Type	Definition
NIX	I*4	Number of type 10 card 'X' direction internal interface input
NIY	I*4	Number of type 10 card 'Y' direction internal interface input

Name	Type	Definition
NX	I*4	Number of different 'X' direction internal interface type 11 input cards
NY	I*4	Number of different 'Y' direction internal interface type 11 input cards

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 11 card
J	I*4	Index on 'X' input type 10 cards Index on 'Y' input type 10 cards
N1	I*4	A.NIP data set reference number
N2	I*4	Number of words requested in array
N3	I*4	Number of type 11 card being processed
N4	I*4	Type number of card being read
XLOW	R*8	Representation of data 'HIGHER'
XHIGH	R*8	Representation of data 'LOWER'

4. Functions and tasks performed by this subroutine

TYP11A is called by subroutine TYPE11. Its purpose is to read in boundary condition constants and the numbers of energy groups affected by those constants for the internal interfaces defined on type 10 cards.

A fatal error at read time causes ISTOP to be set to 3, the appropriate error message to be printed, and subroutine REPORT to be called. The first error return is then used.

The data must conform to certain limits.

ULABEL(I), the boundary label, must match a label on a type 10 card.

IBHEGN(I) must be greater than or equal to zero, and IBLEGN(I) must be greater than or equal to IBHEGN(I) if it is greater than zero.

For IGTYPE less than 30, IBHEGN must be greater than zero.

If unacceptable data is detected, ISTOP is set equal to 1 and processing continues.

After processing, if ISTOP is greater than zero, subroutine REPORT is called and the second error return is used.

As each label is read, the number of different boundaries in each direction is counted to set NX and NY.

5. Subprograms called by this subroutine

SQUEZE (system module)
REPORT

6. Subprograms calling this subroutine

TYPE11

7. Error messages generated by this subroutine

Fatal:

(TYP11A) - CARD SUPPOSEDLY OF TYPE 11 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP11A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP11A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP11A) - INTERNAL INTERFACE LABEL OF 'aaaaaa' GIVEN ON 'ii'-TH TYPE CARD IN A.NIP. DOES NOT MATCH LABEL ON ANY TYPE 10 CARD.

(TYP11A) - NEGATIVE 'aaaaaa'-ENERGY GROUP NUMBER OF 'ii' GIVEN ON 'ii'-TH 11 CARD IN A.NIP.

(TYP11A) - THE LOWER ENERGY GROUP NUMBER IS LESS THAN THE HIGHER ENERGY GROUP NUMBER OF THE 'ii'-TH 11 CARD IN A.NIP.

(TYP11A) - GEOMETRY INDICATED IS ONE DIMENSIONAL. BOUNDARY CONDITION CONSTANTS MAY NOT APPLY TO ALL GROUPS. CHECK CARD NUMBER 'ii' OF TYPE 11 IN A.NIP.

Subroutine TYPE12 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of full words requested for arrays
DUMMY	R*8	Dummy
IATHD1	I*4	Pointer to array ATHD1
IATHD2	I*4	Pointer to array ATHD2
ILABEL	I*4	Pointer to array DLABEL
ITEXD1	I*4	Pointer to array TEXD1
ITEXD2	I*4	Pointer to array TEXD2
JLABEL	I*4	Pointer to array RLABEL

4. Functions and tasks performed by this subroutine

TYPE12 is called from CDSCAN.

Its purpose is to set up array space for the arrays DLABEL, ATHD1, TEXD2, ATHD2, and TEXD2 which are needed for finite geometry data, then to request the pointers to these arrays and pass them in a call to TYP12A where they are processed.

An error during a request for space causes an error message to be printed and ISTOP to be set to 2. Subroutine REPORT is then called and return is made through the first error return.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP12A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE12) - ARRAY OVERFLOW N2 = 'ii'

Subroutine TYP12A (N1, RLABEL, DLABEL, ATHD1, TEXD1, ATHD2, TEXD2, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
RLABEL	R*8	N(6)	Region label from type 06 card
DLABEL	R*8	N(12)	Region label on type 12 card
ATHD1	R*8	N(12)	Actual transverse half distance in first direction
TEXD1	R*8	N(12)	Transverse extrapolation distance in first direction
ATHD2	R*8	N(12)	Actual transverse half distance in second direction for a finite slab
TEXD2	R*8	N(12)	Transverse extrapolation distance in second direction for a finite slab
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 12 cards
J	I*4	Index on type 06 cards
N2	I*4	Index number of type 12 card being processed
N3	I*4	Number of type 06 cards in A.NIP
N4	I*4	Number of type 12 cards in A.NIP
N5	I*4	Type number of card being read
FIRST	R*8	Representation of data 'FIRST'
BLANKS	R*8	Alphameric blanks
SECOND	R*8	Representation of data 'SECOND'

4. Functions and tasks performed by this subroutine

TYP12A handles the input arrays which are later used to write the type 8 record of data set GEOM. It is called by subroutine TYPE12.

When a fatal error occurs at the time a card is read, the appropriate explanatory message is written and quantity ISTOP is set to 3. Subroutine REPORT is called and the program returns via the fatal error return.

The data on each card must conform to certain limits. When it does not, flag ISTOP is set to 1 and card type 12 processing continues.

DLABEL(I) must match a region label from card type 06 (RLABEL). If DLABEL(I) is blank, there must only be one type 12 card input; then the input applies to all regions.

ATHD1(I), the actual transverse half height, must be greater than zero.

If TEXD1(I) is negative, its absolute value must be less than ATHD1(I).

ATHD2(I) must be either all greater than or all equal to zero.

If all ATHD2(I) are greater than zero, each TEXD2(I), if negative must have an absolute value less than the corresponding ATHD2(I).

5. Subprograms called by this subroutine

SQUEZE (system module)
WIPOUT (entry in POINTR)
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYP12A) - CARD SUPPOSEDLY OF TYPE 12 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP12A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP12A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP12A) - REGION LABEL OF 'aaaaaa' ON 'ii'-TH TYPE 12 CARD IN A.NIP. DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD.

(TYP12A) - NEGATIVE VALUE OF 'x.x' GIVEN FOR THE {FIRST, SECOND} TRANSVERSE EXTRAPOLATION DISTANCE ON THE 'ii'-TH TYPE 12 CARD IN A.NIP. EXCEEDS THE MAGNITUDE OF {FIRST, SECOND} ACTUAL TRANSVERSE HALF-DISTANCE ON CARD.

(TYP12A) - NEGATIVE VALUE OF 'x.x' GIVEN FOR THE SECOND ACTUAL TRANSVERSE HALF-DISTANCE ON THE 'ii'-TH TYPE 12 CARD IN A.NIP.

(TYP12A) - THE SECOND ACTUAL TRANSVERSE HALF-DISTANCE GIVEN ON THE TYPE 12 CARDS IN A.NIP ARE NEITHER ALL ZERO (BLANK) NOR ALL POSITIVE, NON-ZERO QUANTITIES.

(TYP12A) - REGION LABEL IN A TYPE 12 CARD IS BLANK. IN THIS CASE ONLY ONE TYPE 12 CARD IS ALLOWED. THERE ARE 'ii' TYPE 12 CARDS IN A.NIP.

Subroutine TYPE13 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of full words requested for arrays
DUMMY	R*8	Dummy
IIADEN	I*4	Pointer to array RIADEN
ILABEL	I*4	Pointer to array RMLABL
JLABEL	I*4	Pointer to array RILABL

4. Functions and tasks performed by this subroutine

TYPE13 is called from subroutine CDSCAN.

When type 13 cards exist space is requested for arrays RMLABL, RILABL, RIADEN, and the pointers for these arrays are requested and passed to subroutine TYP13A.

If array space is not available, the total space asked for is printed and ISTOP is set to 2. Subroutine REPORT is called and the first error is used.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP13A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE13) - ARRAY OVERFLOW. N2 = 'ii'

Subroutine TYP13A (N1, ITYPE, XLABEL, YLABEL, CONC, RMLABL, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
ITYPE	I*4	--	Type number of cards being processed
XLABEL	R*8	3*N(13) or 3*N(14)	Material label on card type 13 or composition label on card type 14
YLABEL	R*8	3*N(13) or 3*N(14)	Isotope label on card type 13 or material label on card type 14
CONC	R*8	3*N(13) or 3*N(14)	Isotope atom density on card type 13 or material volume fraction on card type 14
RMLABL	R*8	N(13)	Material labels from type 13 cards
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag
IWARN	I*4	Warning error flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on input data on type 13 or type 14 cards
J	I*4	Increment used in reading type 13 cards; also index on number of type 13 cards
N2	I*4	Index number of card being read of type 13 or 14
N3	I*4	Number of cards of type being read in A.NIP
N4	I*4	Number of input isotope labels on card type 13 or input material labels on card type 14
N5	I*4	Index base used for card reading, equal to 3*(I-1)
N6	I*4	Type number of card being read
N7	I*4	Number of type 13 cards in A.NIP
ITYPE	I*4	Type number of subroutine that called TYP13A
BLANKS	R*8	Alphameric blank data

4. Functions and tasks performed by this subroutine

TYP13A is a double-use subroutine which is called from TYPE13 and TYPE14. In each case, the cards are read into their predefined arrays and the data is checked for conformity to standards. As in other routines, fatal errors cause explanatory messages, ISTOP is set to 3, REPORT is called and the first error is used.

Improper data causes printing of an explanatory message. ISTOP is set to 1, but processing continues. After processing the card type, REPORT is called and the second error return is used to return to CDSCAN.

The quantities read into arrays from cards are XLABEL(I) and the paired quantities YLABEL(N5+J) and CONC(N5+J), where N5 represents 3*(I-1) or the base for the index of the data pairs, and J is the increment of 1 to 3 on each card.

The XLABEL(I) quantities must not be blank. They represent material labels on type 13 cards or composition labels on type 14 cards.

The YLABEL(N5+J) quantities represent isotope labels on type 13 cards or material labels on type 14 cards. Those on type 14 cards must match a material label from a type 13 card when both card types are present.

The CONC(N5+J) quantities represent isotope atom densities on type 13 cards or material volume fractions on type 14 cards. These are allowed to be negative, but a message is printed to indicate that a negative quantity has been input. IWARN is set to 1.

If IWARN is greater than zero, REPORT is called to indicate that negative volume fractions are present but there is a normal return.

5. Subprograms called by this subroutine

SQUEZE (system module)
REPORT

6. Subprograms calling this subroutine

TYPE

7. Error messages generated by this subroutine

Fatal:

(TYP13A) - CARD SUPPOSEDLY OF TYPE 'ii' FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP13A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP13A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP13A) - BLANK MATERIAL LABEL ON 'ii'-TH 13 CARD IN A.NIP.

(TYP13A) - BLANK COMPOSITION LABEL ON 'ii'-TH 14 CARD IN A.NIP.

(TYP13A) - FIRST ISOTOPE LABEL ON THE 'ii'-TH TYPE 13 CARD IN A.NIP IS BLANK.

(TYP13A) - FIRST MATERIAL LABEL ON THE 'ii'-TH TYPE 14 CARD IN A.NIP IS BLANK.

(TYP13A) - THE MATERIAL LABEL 'aaaaaa' GIVEN ON A TYPE 14 CARD IN A.NIP DOES NOT MATCH ANY MATERIAL LABEL FROM A TYPE 13 CARD.

Informative:

(TYP13A) - NEGATIVE ATOM DENSITY OF 'x.x' GIVEN ON 'ii'-TH TYPE 13 IN A.NIP.

(TYP13A) - NEGATIVE VOLUME FRACTION OF 'x.x' GIVEN ON 'ii'-TH TYPE 14 IN A.NIP.

Subroutine TYPE14 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of full words requested for arrays
DUMMY	R*8	Dummy
IIADEN	I*4	Pointer to array RIADEN
ILABEL	I*4	Pointer to array CLABEL
IMVOLF	I*4	Pointer to array CMVOLF
JLABEL	I*4	Used as pointer to either array RILABL or array CMLABL
KLABEL	I*4	Pointer to array RMLABL

4. Functions and tasks performed by this subroutine

TYPE14 is called from CDSCAN when type 14 cards are present. Its purpose is to define arrays CLABEL, RILABL, and RIADEN if no type 13 cards were supplied. If type 13 cards were present, it defines the arrays CLABEL, CMLABL, and CMVOLF.

Any error on a space request causes a message to be printed along with the amount of space requested. ISTOP is set to 2 and the first error return is used.

When no POINTR error occurs, in the case where type 13 cards are absent the pointers to CLABEL, RILABL, and RIADEN are passed in a call to TYP13A along with a dummy argument of the CLABEL pointer which will not be used.

Where type 13 cards are present, the pointers to CLABEL, CMLABL, CMVOLF, and RMLABL are extracted and passed to subroutine TYP13A. After either call, TYPE14 returns to CDSCAN.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP13A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE14) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE14) - NO TYPE 14 CARDS SUPPLIED IN A.NIP.

Subroutine TYPE15 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

/XSCM/

Variables used from this common block

Name	Type	Definition
NGROUP	I*4	Number of energy groups
NCMP	I*4	Number of compositions
MINFLG	I*4	Flag = 1 if old XS.C.MIN was read
NDUM	I*4	Dummy for alignment
CNAME(199)	R*8	Array of composition names

3. Local Variables

Name	Type	Definition
N2	I*4	Number of full words requested for arrays
DUMMY	R*8	Dummy
ICREGN	I*4	Pointer to array CREGN
ILABEL	I*4	Pointer to array CRLABL
JLABEL	I*4	Pointer to array CLABEL
KLABEL	I*4	Pointer to array RLABEL

4. Functions and tasks performed by this subroutine

Subroutine TYPE15 is always called by CDSCAN. It defines the arrays CRLABL and CREGN in the large container if there are type 15 cards in A.NIP.

When no type 15 card exist, ISTOP is set to 1 and an indicative message is printed. After a call to subroutine REPORT, the second error return is used.

If an error occurs during a PUTPNT space request, a message is printed and ISTOP is set to 2. REPORT is then called and the first error return is used. If there are no errors, the pointers to CRLABL, CREGN, CLABEL, and RLABEL are extracted and passed to subroutine TYP15A.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP15A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE15) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE15) - NO TYPE 15 CARDS SUPPLIED IN A.NIP.

Subroutine TYP15A (N1, CRLABL, CREGN, CLABEL, RLABEL, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
CRLABL	R*8	N(15)	Composition label
CREGN	R*8	10*N(15)	Label of regions containing specified composition
CLABEL	R*8	N(14) or NCMP	Composition label from card type 14 or from old XS.C.MIN data set if present
RLABEL	R*8	N(6)	Region label from card type 06
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

/XSCM/

Variables used from this common block

Name	Type	Definition
NCMP	I*4	Number of compositions in XS.C.MIN data set
MINFLG	I*4	Flag: 0 when XS.C.MIN is not present 1 when XS.C.MIN is read
CNAME(199)	R*8	List of composition names

3. Local Variables

Name	Type	Definition
I	I*4	(1) Index on type 15 cards (2) Index on type 15 card data
J	I*4	(1) Index on type 14 cards (2) Increment of 1 to 10 added to N6 for type 15 card (3) Index on type 15 card data
K	I*4	Index on regions from type 06 card

Name	Type	Definition
N2	I*4	Index number of type 15 card being processed
N3	I*4	(1) Number of type 06 cards in A.NIP (2) Number of data given on type 15 card, 10*N(15)
N4	I*4	(1) Number of type 14 cards (2) Equal to N3-1 or (10*N(15)) -1 = I, used to index CREGN
N5	I*4	(1) Number of type 15 cards (2) Equal to I+1 (I from previous line)
N6	I*4	Base index equal to 10*(I-1) used for indexing CREGN
N7	I*4	(1) Type number of card being read (2) Index increment equal to N6+J on CREGN
BLANKS	R*8	Alphameric Blank data

4. Functions and tasks performed by this subroutine

TYP15A is called from TYPE15 to define the composition-region assignments for use later in defining record type 5 of the GEOM output data set.

Each card contains a composition label and up to 10 region labels which contain this composition. A read error will cause an appropriate error message to be printed and quantity ISTOP to be set to 3. Subroutine REPORT will then be called and the first error return will be taken.

If no error occurs, all labels are left justified, with imbedded blanks removed, and checked for conformity.

The following limits apply.

CRLABL(I), the composition on type 15 card, must duplicate some composition label CLABEL(J) from a type 14 card or CLABEL(J) from old XS.C.MIN.

Defining N7 as (10*(I-1) + J), for J = 1, 2, . . ., 10, the region containing this composition CREGN(N7) must be blank or the same as some RLABEL(K) from a type 06 card. (N7 represents the index on the input region labels on card type 15).

Each CREGN(N7) must be unique.

If a deviation occurs, ISTOP is set to 1 and after all cards have been processed subroutine REPORT is called and the second error return is used.

5. Subprograms called by this subroutine

SQUEZE (system module)
REPORT

6. Subprograms calling this subroutine

TYPE15

7. Error messages generated by this subroutine

Fatal:

(TYP15A) - CARD SUPPOSEDLY OF TYPE 15 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP15A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

Serious:

(TYP15A) - THE COMPOSITION LABEL 'aaaaaa' GIVEN ON A TYPE 15 CARD IN A.NIP DOES NOT MATCH ANY COMPOSITION LABEL GIVEN ON A TYPE 14 CARD.

(TYP15A) - THE 'ii'-TH REGION LABEL 'aaaaaa' GIVEN ON THE 'ii'-TH TYPE 15 CARD IN A.NIP DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD.

(TYP15A) - THE REGION LABEL 'aaaaaa' APPEARS MORE THAN ONCE IN THE SET OF TYPE 15 CARDS SUPPLIED IN A.NIP.

(TYP15A) - THE COMPOSITION LABEL 'aaaaaa' GIVEN ON A TYPE 15 CARD IN A.NIP DOES NOT MATCH ANY COMPOSITION LABEL GIVEN IN THE OLD XS.C.MIN DATA SET.

Subroutine TYPE16 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block - None

3. Local Variables - None

4. Functions and tasks performed by this subroutine

TYPE16 is called by CDSCAN. At present this is a dummy routine and any type 16 cards are skipped by a call to SPACE. Return is made to CDSCAN. Initial fission source or flux specifications are not implemented at present in GNIP.

5. Subprograms called by this subroutine

SPACE

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine - None

Subroutine TYPE17 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables - None

4. Functions and tasks performed by this subroutine

TYPE17 is called from CDSCAN. It skips any type 17 cards by means of a call to SPACE, and returns to CDSCAN. Use of distributed initial fission source or flux data is not implemented in GNIP.

5. Subprograms called by this subroutine

SPACE

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine - None

Subroutine TYPE18 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables - None

4. Functions and tasks performed by this subroutine

TYPE18 is called from CDSCAN. It skips any type 18 cards by means of a call to SPACE, and returns to CDSCAN. Use of distributed initial fission source or flux data is not implemented to GNIP.

5. Subprograms called by this subroutine

SPACE

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine - None

Subroutine TYPE19 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Index number of card being read
DUMMY	R*8	Dummy
ICODE	I*4	Pointer to array IICODE
IIHEGN	I*4	Pointer to array ISHEGN
IILEGN	I*4	Pointer to array ISLEGN
IISPEC	I*4	Pointer to array ISPECN
ILABEL	I*4	Pointer to array SILABL
IRLISC	I*4	Pointer to array RLISC

Name	Type	Definition
IRUISC	I*4	Pointer to array RUISC
ISOURC	I*4	Pointer to array SOURCE
JLABEL	I*4	Pointer to array RLABEL
KLABEL	I*4	Pointer to array ALABEL
LLABEL	I*4	Pointer to array ZLABEL

4. Functions and tasks performed by this subroutine

TYPE19 is called from CDSCAN if type 19 cards exist in A.NIP. It requests array space for the arrays used in preparing the distributed external source data set ES.D1D or ES.D2D. These are IICODE, SILABL, ISHEGN, ISLEGN, SOURCE, RLISC, RUISC, and ISPECN.

The pointers for needed arrays are extracted and passed to subroutine TYP19A, where the processing is carried out.

If a POINTR error occurs, the amount of space requested is printed and ISTOP is set equal to 2. Subroutine REPORT is called and it returns via the first error return.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP19A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE19) - ARRAY OVERFLOW. N2 = 'ii'

Subroutine TYP19A (N1, IICODE, SILABL, ISHEGN, ISLEGN, SOURCE, ISPECN, RLISC, RUISC, RLABEL, ALABEL, ZLABEL, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
IICODE	I*4	N(19)	Flag which indicates how data is input
SILABL	R*8	N(19)	Label of region, area, or zone
ISHEGN	I*4	N(19)	Initial energy group number to which source applies
ISLEGN	I*4	N(19)	Final energy group number to which source applies
SOURCE	R*8	N(19)	Isotropic source value in specified mesh interval for the given range
ISPECN	I*4	N(19)	Spectrum distribution flag
RLISC	R*8	N(19)	Lower 'X' direction coordinate of mesh interval containing this source
RUISC	R*8	N(19)	Upper 'Y' direction coordinate of mesh interval containing this source
RLABEL	R*8	N(6)	Region LABEL
ALABEL	R*8	N(7)	Area LABEL
ZLABEL	R*8	N(8)	Zone LABEL
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 19 cards
N2	I*4	I, the index number of card 19 being processed
N3	I*4	Number of type 19 cards in A.NIP
N4	I*4	Type number on the card being processed
BLANKS	R*8	Alphameric blank data

4. Functions and tasks performed by this subroutine

TYP19A is called by subroutine TYPE19.

As each card is read, the arrays IICODE, SILABL, ISPECN, ISHEGN, ISLEGN, SOURCE, RLISC, and RUISC are filled. Read errors cause an appropriate message to be printed and the flag ISTOP to be set to 3. After a call to REPORT, the routine uses the first error return.

Data must conform to the following specifications.

ISPECN(I) must be zero or 1.

ISHEGN(I) must be positive and less than ISLEGN(I) if ISLEGN(I) is greater than zero.

ISLEGN(I) must be zero or greater.

If input is not by mesh interval, i.e. if SILABL(I) is not blank, the SILABL array is left-justified, with imbedded blanks removed and passed to subroutine TYP19B, where further checking is done on arrays IICODE and SILABL.

In the coordinate arrays, RLISC(I), the lower 'X' coordinate of interval containing SOURCE(I), must be positive, and RUISC(I), the lower 'Y' coordinate of interval containing SOURCE(I), must be greater than RLISC(I) or must be blank.

If data does not conform, ISTOP is set equal to 1, an error message is issued and subroutine REPORT is called after all cards are processed. Return is by means of the second error return.

5. Subprograms called by this subroutine

SQUEZE (system module)
TYP19B
REPORT

6. Subprograms calling this subroutine

TYPE19

7. Error messages generated by this subroutine

Fatal:

(TYP19A) - CARD SUPPOSEDLY OF TYPE 19 FOUND TO BE TYPE 'ii' IN A.NIP.

(TYP19A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP19A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP19A) - THE NEGATIVE VALUE 'ii' WAS GIVEN FOR AN ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 19 CARD IN A.NIP.

(TYP19A) - THE HIGHER ENERGY GROUP NUMBER IS GREATER THAN THE LOWER ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 19 CARD IN A.NIP.

(TYP19A) - THE NEGATIVE VALUE 'x.x' WAS GIVEN FOR A LOWER COORDINATE ON THE 'ii'-TH TYPE 19 CARD IN A.NIP.

(TYP19A) - ERROR ON CARD NO. 'ii' OF TYPE 19 IN A.NIP. VALUE GIVEN WAS 'ii'. ZERO OR 1 IS REQUIRED.

(TYP19A) - ERROR ON CARD NUMBER 'ii' ON TYPE 19 IN A.NIP. GROUP DEPENDENT INPUT MAY NOT HAVE A GROUP WEIGHTING FLAG OF 1.

Subroutine TYP19B (I, ICODE, SLABEL, RLABEL, ALABEL, ZLABEL, ITYPE)

1. Arguments

Name	Type	Dimension	Definition
I	I*4	--	Index number of card type 19 being processed
ICODE	I*4	--	Input type code
SLABEL	R*8	N(19)	Label of input entity containing source
RLABEL	R*8	N(6)	Region LABEL
ALABEL	R*8	N(7)	Area LABEL
ZLABEL	R*8	N(8)	Zone LABEL
ITYPE	I*4	--	Card type being processed

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
J	I*4	Index on type 6, 7, or 8 cards
N6	I*4	Number of type 06 cards in A.NIP
N7	I*4	Number of type 07 cards in A.NIP
N8	I*4	Number of type 08 cards in A.NIP
SIX	R*8	Representation of alphameric data 'SIX'
AREA	R*8	Representation of alphameric data 'AREA'
ZONE	R*8	Representation of alphameric data 'ZONE'
EIGHT	R*8	Representation of alphameric data 'EIGHT'
SEVEN	R*8	Representation of alphameric data 'SEVEN'
REGION	R*8	Representation of alphameric data 'REGION'

4. Functions and tasks performed by this subroutine

Subroutine TYP19B is called from a card processing loop in subroutine TYP19A. It checks the labels given on type 19 cards to see that they match labels that have been previously defined.

ICODE(I) must be 0, 1, or 2.

If ICODE(I) = 1, N(7) must be greater than zero and SLABEL(I) must match some ALABEL(J) from a type 07 card.

If ICODE(I) = 2, N(8) must be greater than zero and SLABEL(I) must match some ZLABEL(J) from a type 08 card.

If ICODE(I) = 0, then SLABEL(I) must match some RLABEL(J) on a type 06 card.

When these standards are not met, an explanatory message is printed and flag ISTOP is set to 1. No action is taken until after return to TYP19A, and after all cards are processed.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

TYP19A

7. Error messages generated by this subroutine

Serious:

(TYP19B) - THE CONTENTS ON COLS. 7-12 ON THE 'ii'-TH TYPE 'ii' CARD IN A.NIP IS NOT 0, 1, OR 2.

(TYP19B) - THE 'ii'-TH TYPE 'ii' CARD IN A.NIP HAS THE VALUE 'ii' IN COLS. 7-12. THIS MEANS THAT COLS. 13-18 OF THE CARD MUST CONTAIN A 'aaaaaa' LABEL. HOWEVER, NO TYPE 'aaaaaa' CARDS WERE INCLUDED IN A.NIP.

(TYP19B) - THE LABEL ON THE 'ii'-TH TYPE 'ii' CARD IN A.NIP DOES NOT MATCH THE 'aaaaaa' LABEL GIVEN ON ANY TYPE 'aaaaaa' CARD.

Subroutine TYPE20 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables - None

4. Functions and tasks performed by this subroutine

TYPE20 is called from CDSCAN. It skips any type 20 cards by means of a call to SPACE and returns to CDSCAN. Use of distributed initial fission source or flux data is not implemented in GNIP.

5. Subprograms called by this subroutine

SPACE

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine - None

Subroutine TYPE21 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP21/

Variables used from this common block

Name	Type	Definition
ISSPEC	I*4	Search type specification
MAXNSP	I*4	Maximum number of search passes
PKOA	R*8	Prescribed k-effective or initial alpha
EPSLON	R*8	Convergence criterion epsilon for k-eff or alpha

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Index number of type 21 card being processed; also N(21)-1
N3	I*4	Type number on card being processed

4. Functions and tasks performed by this subroutine

TYPE21 is called by CDSCAN when type 21 cards exist in A.NIP. Only one type 21 card is used. Any others are skipped by a call to SPACE. Any error during the card reading causes an appropriate explanatory message to be printed and flag ISTOP to be set to 3. Subroutine REPORT is called and the fatal error return is used.

The following conditions must be met by the data on the card.

ISSPEC, the search type specification, must be zero or 1.

MAXNSP, the maximum number of seach passes, must be positive.

If ISSPEC = 0, indicating a search to obtain a prescribed KEFF, PKOA, the prescribed k-eff, must be non-zero. PKOA is set to 1.0 when input is blank.

EPSLON, the convergence criterion, must be positive; otherwise, a default of 10^{-4} is supplied.

If these conditions are not met, flag ISTOP is set to 1 and after the indicative message is printed, subroutine REPORT is called and the second error return is used.

The data from this card is used in the preparation of output data set SP.CRIT.

5. Subprograms called by this subroutine

SPACE
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE21) - CARD SUPPOSEDLY OF TYPE 21 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYPE21) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYPE21) - READ ERROR. ERROR IN DATA TRANSFER, N2 = 'ii'

Serious:

(TYPE21) - CONTENTS OF COLS. 13-18 ON THE TYPE 21 CARD IN A.NIP ARE NOT 0 OR 1.

(TYPE21) - MAXIMUM NUMBER OF SEARCH PASSES IN COLS. 19-24 ON THE TYPE 21 CARD IN A.NIP IS NEGATIVE.

(TYPE21) - THE VALUE OF THE PRESCRIBED K-EFFECTIVE IN COLS. 25-36 OF THE TYPE 21 CARD IN A.NIP IS NEGATIVE.

Informative:

(TYPE21) - THE CONVERGENCE CRITERION IN COLS. 37-48 OF THE TYPE 21 CARD IN A.NIP IS NOT POSITIVE. DEFAULT OF 1.0E-04 HAS BEEN SUBSTITUTED.

Subroutine TYPE22 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP22/

Variables used from this common block

Name	Type	Definition
SIVX	R*8	Initial value of X, the search parameter
SSVX	R*8	Second value of X
SLBX	R*8	Lower bound for X
SUBX	R*8	Upper bound for X
SDRX	R*8	Derivative of keff or alpha with respect to X

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Index number of type 22 card being processed; also N(22)-1
N3	I*4	Type number on card being processed

4. Functions and tasks performed by this subroutine

TYPE 22 is always called by CDSCAN if a type 21 card has been supplied. If no type 22 card then occurs in A.NIP, error flag ISTOP is set to 1 and an explanatory message is printed. Subroutine REPORT is called and return is made using the second error return.

Only one type 22 card is used. Any others are bypassed by means of a call to subroutine SPACE.

Quantities SIVX, SSVX, SLBX, SUBX, and SDRX are read from the card. If a read error occurs, the appropriate error message is printed and ISTOP is set to 3. Subroutine REPORT is called and it returns via the first error return.

5. Subprograms called by this subroutine

SPACE
REPORT

6. Subprograms called this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE22) - CARD SUPPOSEDLY OF TYPE 22 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYPE22) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYPE22) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYPE22) - A TYPE 21 CARD WAS SUPPLIED IN A.NIP BUT NO TYPE 22 CARD WAS SUPPLIED.

Subroutine TYPE23 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of words of array space requested
DUMMY	R*8	Dummy
ILABEL	I*4	Pointer to array SLABL1

Name	Type	Definition
JLABEL	I*4	Pointer to array SLABL2
KLABEL	I*4	Pointer to array CLABEL

4. Functions and tasks performed by this subroutine

Subroutine TYPE23 is called by CDSCAN when type 23 cards exist in A.NIP. Its purpose is to set up the array space required for the reading of type 23 cards for the preparation of data set SP.CRIT.

If there are cards to type 24, 25, or 26, in addition to the type 23 cards, flag ISTOP is set equal to 1 and subroutine REPORT is called after the informative message has been printed. This returns via the second error return.

Space is requested for arrays SLABL1 and SLABL2. When this space is available, the pointers for these arrays and array CLABEL are extracted and passed to subroutine TYP23A.

If space is not available, a message stating this is printed and ISTOP is set equal to 2. REPORT is called and the first error return is used.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP23A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

(TYPE23) - ARRAY OVERFLOW. N2 - 'ii'

Serious:

(TYPE23) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25, 26 WERE SUPPLIED IN A.NIP.

Subroutine TYP23A (N1, SLABL1, SLABL2, CLABEL, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
SLABL1	R*8	N(23)	Composition label of modifier composition
SLABL2	R*8	10*N(23)	Composition label of compositions with which modifier is used
CLABEL	R*8	N(14)	Composition label from card type 14
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

/XSCM/

Variables used from this common block

Name	Type	Definition
NCMP	I*4	Number of compositions in XS.C.MIN data set
MINFLG	I*4	Flag = 0 if XS.C.MIN is not present = 1 if XS.C.MIN has been read

3. Local Variables

Name	Type	Definition
I	I*4	Index on SLABL1
J	I*4	Index increment to N5 equals 1 to 10
N2	I*4	Index of type 22 card being processed
N3	I*4	Number of type 23 cards present in A.NIP
N4	I*4	Number of type 14 cards present in A.NIP or NCMP from old data set XS.C.MIN
N5	I*4	Equal to 10*I-10, the quantity which is incremented by J to make the index of SLABL2
N6	I*4	Type number of card being processed
X712	R*8	Representation of data '7-12'
X1372	R*8	Representation of data '13-72'
BLANKS	R*8	Alphameric blanks

4. Functions and tasks performed by this subroutine

TYP23A is called by TYPE23 after it successfully sets up array space.

One composition name, SLABL(1), and up to ten SLABL2(I) composition names are read from each card.

As the reading proceeds, any error will cause ISTOP to be set to 1 after the proper explanatory message has been printed. Subroutine REPORT is then called and the first error return system is used.

The quantities on the cards must conform to the following.

SLABL1(I), the composition used as the modifier composition, must duplicate some composition label, CLABEL(J).

SLABL2(1) the compositions which are to be modified, must either be blank or match some CLABEL(J).

If a discrepancy is found, the index number of the erroneous card and the SLABL2(I) are printed and ISTOP is set to 1.

When ISTOP is greater than zero, TYP23A calls subroutine REPORT and the second error return is used.

5. Subprograms called by this subroutine

SQUEZE (system module)
REPORT

6. Subprograms calling this subroutine

TYPE23

7. Error messages generated by this subroutine

Fatal:

(TYP23A) - CARD SUPPOSEDLY OF TYPE 23 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP23A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP23A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP23A) - THE QUANTITY IN COLS. {7-13/13-72} ON THE 'ii'-TH TYPE 23 CARD IN A.NIP 'aaaaaa' DOES NOT DUPLICATE ANY COMPOSITION LABEL GIVEN ON A TYPE 14 CARD OR ON THE OLD XS.C.MIN DATA SET.

Subroutine TYPE24 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block·

Name	Type	Definition
A(1)	R*8	Large container array

/CTYP24/

Variables used from this common block

Name	Type	Definition
NMX	I*4	Number of 'X'-direction mesh input modifiers in criticality search
NMY	I*4	Number of 'Y'-direction mesh input modifiers in criticality search

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Number of words requested in array
DUMMY	R*8	Dummy
IXMESH	I*4	Pointer to array XMESHM
IXMODL	I*4	Pointer to array XMODLC
IXMODU	I*4	Pointer to array XMODUC
IYMESH	I*4	Pointer to array YMESHM
IYMODL	I*4	Pointer to array YMODLC
IYMODU	I*4	Pointer to array YMODUC

4. Functions and tesks performed by this subroutine

TYPE24 is called by subroutine CDSCAN when there are type 24 cards in A.NIP.

If there are also cards of type 23, 25, or 26 present an error message is printed and ISTOP is set to 1.

The routine requests space for the arrays XMESHM, YMODLC, YMODUC, and YMESHM. If not enough space is available, the amount requested is printed in an explanatory message and ISTOP is set to 2. REPORT is then called and use is made of the first error return.

After a successful PUTPNT request, the pointers to the arrays are extracted and passed to subroutine TYP24A.

If both 'X' and 'Y' direction input was supplied, the arrays XMODLC, XMODUC, and XMESHM are redefined to length NMX, and arrays YMODLC, YMODUC, and YMESHM are redefined to length NMY.

If ISTOP is greater than zero, REPORT will be called and return is via the second error return system.

5. Subprograms called by this subroutine

REDEF (entry in POINTR)
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP24A
WIPOUT (entry in POINTR)
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE24) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE24) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25 or 26 WERE SUPPLIED IN A.NIP.

Subroutine TYP24A (N1, XMODLC, XMODUC, XMESHM, YMODLC, YMODUC, YMESHM, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
XMODLC	R*8	NMX	'X' direction lower coordinate of mesh interval range being modified
XMODUC	R*8	NMX	'X' direction upper coordinate of mesh interval range being modified
XMESHM	R*8	NMX	Dimensionless mesh modifier for the above 'X' interval range
YMODLC	R*8	NMY	'Y' direction lower coordinate of mesh interval range being modified
YMODUC	R*8	NMY	'Y' direction upper coordinate of mesh interval range being modified
YMESHM	R*8	NMY	Dimensionless mesh modifier for the above 'Y' interval range
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP24/

Variables used from this common block

Name	Type	Definition
NMX	I*4	Number of 'X'-direction mesh modifiers input in criticality search
NMY	I*4	Number of 'Y'-direction mesh modifiers input in criticality search

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on number of type 24 cards
X	I*4	Representation of alphameric data 'X'
Y	I*4	Representation of alphameric data 'Y'
N2	I*4	Index number of card type 24 being processed
N3	I*4	Number of type 24 cards in A.NIP
N4	I*4	Type number of card being processed
RLC	R*8	Input lower coordinate of lower interval defining range to be modified
RMM	R*8	Mesh modifier to be applied in given range of intervals
RUC	R*8	Input upper coordinate of upper interval defining range to be modified
COORD	R*8	Input coordinate direction name on card being processed

4. Functions and tasks performed by this subroutine

Subroutine TYP24A is called by TYPE24 after it successfully sets up the needed arrays for a dimension search.

As each card is read, any fatal reading error causes an explanatory error message to be printed and the flag ISTOP to be set to 3. Subroutine REPORT is called and the first error return system is used.

The quantities read from the cards are COORD, RLC, RUC, and RMM. In order to be valid, the data must comply with the following.

RLC must be greater than or equal to zero and less than RUC.

COORD must either be 'X' or 'Y'.

If this is true the following are defined.

If COORD is equal to 'X', NMX = NMX+1, XMODLC(NMX) = RLC, XMODUC(NMX) = RUC, and XMESHM(NMX) = RMM.

Similarly, if COORD = 'Y', NMY = NMY+1, YMODLC(NMY) = RLC, YMODUC(NMY) = RUC, and YMESHM(NMY) = RMM.

If ISTOP is greater than zero after all cards are processed, subroutine REPORT is called and the second error returns are used.

These arrays are used to construct the second record of data set SP.CRIT.

5. Subprograms called by this subroutine

REPORT

6. Subprograms calling this subroutine

TYPE24

7. Error messages generated by this subroutine

Fatal:

(TYP24A) - CARD SUPPOSEDLY OF TYPE 24 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP24A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP24A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP24A) - THE LOWER COORDINATE GIVEN ON THE 'ii'-TH TYPE 24 CARD IN A.NIP IS NEGATIVE.

(TYP24A) - THE LOWER COORDINATE GIVEN ON THE 'ii'-TH TYPE 24 CARD IN A.NIP IS GREATER THAN THE UPPER COORDINATE GIVEN ON THE CARD.

(TYP24A) - THE COORDINATE DESIGNATOR GIVEN IN COL. 12 ON THE 'ii'-TH TYPE 24 CARD IN A.NIP IS NOT X OR Y.

Subroutine TYPE25 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	3*N(25), amount of array space requested
DUMMY	R*8	Dummy
IBMOD1	I*4	Pointer to array BMOD1
IBMOD2	I*4	Pointer to array BMOD2

Name	Type	Definition
ILABEL	I*4	Pointer to array SLABLR
JLABEL	I*4	Pointer to array RLABEL

4. Functions and tasks performed by this subroutine

TYPE25 is called from CDSCAN when type 25 cards exist in A.NIP.

Its purpose is to create the arrays used in forming the output data set SP.CRIT which are filled in subroutine TYP25A.

The routine sets ISTOP equal to 1 when there are more than one of card types 23, 24, 25 or 26 present in A.NIP. After indicating this condition with a message, it calls subroutine REPORT and uses the second error return.

Otherwise the request is made for space to PUTPNT for arrays SLABLR, BMOD1, and BMOD2. An insufficient amount of available space causes an explanatory message to be printed and ISTOP to be set to 2. Subroutine REPORT is then called and the first error return is used.

In the event of no POINTR error, the pointers to arrays SLABLR, BMOD1, BMOD2, and RLABEL are extracted and passed to subroutine TYP25A.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
TYP25A
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE25) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE25) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25, 26 WERE SUPPLIED IN A.NIP.

Subroutine TYP25A (N1, SLABLR, BMOD1, BMOD2, RLABEL, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
SLABLR	R*8	N(25)	Input label of region being modified
BMOD1	R*8	N(25)	Input buckling modifier in first transverse direction
BMOD2	R*8	N(25)	Input buckling modifier in second transverse direction
RLABEL	R*8	N(6)	Region label from type 06 cards
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 25 cards
J	I*4	Index on type 06 cards
N2	I*4	Index number of type 25 card being processed

Name	Type	Definition
N3	I*4	Number of type 06 cards in A.NIP
N4	I*4	Number of type 25 cards in A.NIP
N5	I*4	Type number on card being processed
BLANKS	R*8	Alphameric blanks

4. Functions and tasks performed by this subroutine

TYP25A is called by subroutine TYPE25 to read the buckling modifiers for a buckling search.

Each card read contains a region label SLABLR(I) and buckling modifiers BMOD1(I) and BMOD2(I).

A serious reading error will cause an explanation to be printed and quantity ISTOP to be set to 3. Subroutine REPORT will be called and the fatal error return will be used.

The input data must conform to the following specifications.

SLABLR(I) must be either blank or the same as some RLABEL(J), a region label from card type 06.

All SLABLR(I) must be different.

ISTOP is set to 1 whenever these requirements are not met.

After processing type 25 cards, subroutine REPORT is called if ISTOP is greater than zero. Return in that case is made through the second error return.

5. Subprograms called by this subroutine

SQUEZE (system module)

6. Subprograms calling this subroutine

TYPE25

7. Error messages generated by this subroutine

Fatal:

(TYP25A) - CARD SUPPOSEDLY OF TYPE 25 FOUND TO BE OF TYPE 'ii'
IN A.NIP.

(TYP25A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP25A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP25A) - THE LABEL GIVEN ON THE 'ii'-TH TYPE 25 CARD IN A.NIP DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD

(TYP25A) - THE REGION LABEL 'aaaaaa' APPEARS ON MORE THAN ONE TYPE 25 CARD IN A.NIP.

Subroutine TYPE26 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/CTYP26/

Variables used from this common block

Name	Type	Definition
ALPHAM	R*8	Alpha modifier for a criticality search

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Index number of card type 26 being processed; also number of remaining type 26 cards
N3	I*4	Type number on card being processed

4. Functions and tasks performed by this subroutine

TYPE26 is called by CDSCAN when a type 26 card is detected in A.NIP. There should be only one type 26 card. Any others are skipped in a call to subroutine SPACE.

If a type 23, 24, or 25 card is present, a message is printed and ISTOP is set to 1.

ALPHAM is read from the card. If a reading error occurs a message is printed and ISTOP is set to 3.

If ISTOP is greater than zero, REPORT is called and the return used depends on the value of ISTOP. If ISTOP = 1, the second error return is used. If ISTOP = 3, the first error return is used.

5. Subprograms called by this subroutine

SPACE
REPORT

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE26) - CARD SUPPOSEDLY OF TYPE 25 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYPE26) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYPE26) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYPE26) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25, 26 WERE SUPPLIED IN A.NIP.

Subroutine TYPE27 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Amount of array space requested
DUMMY	R*8	Dummy
ICSORS	I*4	Pointer to array CISORS
IIHEGN	I*4	Pointer to array IAHEGN
IILEGN	I*4	Pointer to array IALEGN
ISLABL	I*4	Pointer to array SSLABL
IXSORS	I*4	Pointer to array AXSORS
IYSORS	I*4	Pointer to array AYSORS

4. Functions and tasks performed by this subroutine

TYPE27 is called by CDSCAN when type 27 cards exist in A.NIP. The purpose of the routine is to allocate arrays necessary for reading the surface source specifications used in writing data set ES.D1SH and ES.D2SH.

These arrays are SSLABL, IAHEGN, IALEGN, CISORS, AXSORS, and AYSORS.

If space is available the pointers to the arrays are extracted and passed to subroutine TYP27A where the array contents are read.

If a POINTR error occurs an error message is printed, flag ISTOP is set to 2, and subroutine REPORT is called. The first error return is used.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
REPORT
TYP27A

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE27) - ARRAY OVERFLOW. N2 = 'ii'

Subroutine TYP27A (N1, SSLABL, IALEGN, IAHEGN, CISORS, AXSORS, AYSORS, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
SSLABL	R*8	N(27)	Inhomogeneous surface source label
IALEGN	I*4	N(27)	Lower energy group number to which source applies
IAHEGN	I*4	N(27)	Higher energy group number to which source applies
CISORS	R*8	N(27)	Isotropic component of source
AXSORS	R*8	N(27)	Anisotropic 'X' direction component of source
AYSORS	R*8	N(27)	Anisotropic 'Y' direction component of source
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/CTYP27/

Variables used from this common block

Name	Type	Definition
NORDER	I*4	Anisotropy flag for data sets ES.D1SH and ES.D2SH

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
I	I*4	Index of type 28 card being processed
N2	I*4	I minus the index of card being processed
N3	I*4	N(27) number of type 27 cards in A.NIP
N4	I*4	Type number on card being processed
BLANKS	R*8	Alphameric blanks

4. Functions and tasks performed by this subroutine

TYP27A is called after successful array allocation from TYPE27. The quantities SSLABL, IAHEGN, IALEGN, CISORS, AXSORS, and AYSORS are read from each type 27 card. A read error causes an explanatory message, and after ISTOP is set to 2 REPORT is called and return is made by means of the second error return system.

The data items are checked for validity as follows:

SSLABL(I) must equal 0 when IAHEGN = 0,

IALEGN(I) must be > 0, and

IALEGN(I) must be $\geq$ IAHEGN(I) when IAHEGN > zero.

At least one of CISORS(I), AXSORS(I) or AYSORS(I) must be > 0.0.

Then an item does not meet the requirements, counter ISTOP is set to 1. After all cards of TYPE27 are processed, subroutine REPORT is called and the second error return is used.

5. Subprograms called by this subroutine

REPORT
SQUEZE (system module)

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYP27A) - CARD SUPPOSEDLY OF TYPE 27 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP27A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP27A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP27A) - THE NEGATIVE VALUE 'ii' WAS GIVEN FOR AN ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 27 CARD IN A.NIP.

(TYP27A) - THE HIGHER ENERGY GROUP NUMBER IS GREATER THAN THE LOWER ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 27 CARD IN A.NIP.

(TYP27A) - ALL THREE SOURCE VALUES IN COLS. 37-72 of the 'ii'-TH TYPE 27 CARD IN A.NIP ARE ZERO OR BLANK.

(TYP27A) - THERE IS NO LABEL GIVEN IN COLS. 7-12 ON CARD NO. 'ii' OF TYPE 27 IN A.NIP.

Subroutine TYPE28 (N1, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag

3. Local Variables

Name	Type	Definition
N2	I*4	Amount of array space requested
DUMMY	R*8	Dummy
ICODE	I*4	Pointer to array CORDIR
IALABL	I*4	Pointer to array SALABL
IDNCOR	I*4	Pointer to array DNCORD

Name	Type	Definition
IONCOR	I*4	Pointer to array ONCORD
IUPCOR	I*4	Pointer to array UPCORD
ISLABL	I*4	Pointer to array SSLABL

4. Functions and tasks performed by this subroutine

Subroutine TYPE28 allocates array space for additional arrays used in writing the ES.D1SH and ES.D2SH data set.

It is called when type 27 cards are present. Type 28 cards must be present when type 27 cards are present.

A POINTR call requests space for arrays SALABL, CORDIR, DNCORD, UPCORD, and ONCORD. The pointers to these arrays are then passed to subroutine TYP28A along with the pointer to array SSLABL.

If a POINTR error occurs, a message is printed, flag ISTOP is set to 2, and subroutine REPORT is called. Control is returned to the calling program by means of the first error return system.

5. Subprograms called by this subroutine

GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
REPORT
TYP28A

6. Subprograms calling this subroutine

CDSCAN

7. Error messages generated by this subroutine

Fatal:

(TYPE28) - ARRAY OVERFLOW. N2 = 'ii'

Subroutine TYP28A (N1, SALABL, CORDIR, DNCORD, UPCORD, ONCORD, SSLABL, *, *)

1. Arguments

Name	Type	Dimension	Definition
N1	I*4	--	A.NIP data set reference number
SALABL	R*8	N(28)	Source label
CORDIR	R*8	N(28)	Coordinate direction of face
DNCORD	R*8	N(28)	Lower end coordinate of specified source on face
UPCORD	R*8	N(28)	Upper end coordinate of specified source on face
ONCORD	R*8	N(28)	'Y' coordinate along which 'X' direction face lies 'X' coordinate along which 'Y' direction face lies
SSLABL	R*8	N(27)	Inhomogeneous surface source label
*	--	--	Fatal error return
*	--	--	Serious error return

2. Common Blocks

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Number of cards of each type present in A.NIP

/IERR/

Variables used from this common block

Name	Type	Definition
ISTOP	I*4	Error return flag
IQUIT	I*4	Path termination flag

3. Local Variables

Name	Type	Definition
I	I*4	Index on type 28 cards
J	I*4	Index on type 27 cards
X	R*8	Data name 'X'
Y	R*8	Data name 'Y'
N2	I*4	Index of card being processed
N3	I*4	N(28)
N4	I*4	Type number on card being processed
N5	I*4	N(27)
DN1	R*8	Dummy used for re-reading DNCORD field in T format
DN2	R*8	Dummy used for re-reading DNCORD field in T format
ON1	R*8	Dummy used for re-reading ONCORD field in T format
ON2	R*8	Dummy used for re-reading ONCORD field in T format
UP1	R*8	Dummy used for re-reading UPCORD field in T format
UP2	R*8	Dummy used for re-reading UPCORD field in T format
BLANKS	R*8	Alphameric blanks

4. Functions and tasks performed by this subroutine

TYP28A is called by TYPE28 to read the data on the type 28 cards.

Quantities read are SALABL, CORDIR, DNCORD, UPCORD, and ONCORD.

Requirements are as follows.

SALABL(I) must be the same as some SSLABL(I) on a type 27 card and must be non-blank.

CORDIR(I) must be 'X' or 'Y'.

DNCORD(I) must be less than UPCORD(I) when UPCORD(I) is greater than 0.0.

If a label is blank, ISTOP is set to 3 and IQUIT is set to 1. Card 28 reading continues, but return is made after a call to REPORT via the first error return system.

Quantities DNCORD(I), UPCORD(I) and ONCORD(I) are re-read in T format and compared with blanks. Any one of these is set to -1 if it contains a blank.

If any of the other requirements are not met, ISTOP is set to 1 and REPORT is called after all type 28 cards have been read. The second error return is used.

5. Subprograms called by this subroutine

REPORT
SQUEZE (system module)

6. Subprograms calling this subroutine

TYPE28

7. Error messages generated by this subroutine

Fatal:

(TYP28A) - CARD SUPPOSEDLY OF TYPE 28 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP28A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP28A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP28A) - LABEL ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH ANY LABEL ON A TYPE 27 CARD.

(TYP28A) - THE LOWER COORDINATE GIVEN ON THE 'ii'-TH TYPE 28 CARD IN A.NIP HAS A NEGATIVE VALUE.

(TYP28A) - THE UPPER COORDINATE GIVEN ON THE 'ii'-TH TYPE 28 CARD IN A.NIP HAS A NEGATIVE VALUE.

(TYP28A) - COORDINATE DIRECTION GIVEN ON THE 'ii'-TH TYPE 28 CARD IN A.NIP IS NOT X OR Y.

(TYP28A) - THERE IS NO LABEL GIVEN IN COLS. 7-12 ON CARD NO. 'ii' OF TYPE 28 IN A.NIP.

(TYP28A) - LOWER COORDINATE IS NOT LESS THAN UPPER COORDINATE GIVEN ON THE 'ii'-TH CARD IN A.NIP.

(TYP28A) - THE COORDINATE IN COLUMNS 60 TO 72 ON THE 'ii'-TH CARD IN A.NIP IS NOT POSITIVE.

Subroutine ZONEDR (*)

1. Arguments

Name	Type	Dimension	Definition
*	--	--	Error return

2. Common Blocks

/ARRA/

Variables used from this common block

Name	Type	Definition
A(1)	R*8	Large container array

/NCARDS/

Variables used from this common block

Name	Type	Definition
N(99)	I*4	Array of number of cards of each type present in A.NIP

/GEOM1/

Variables used from this common block

Name	Type	Definition
NDIM	I*4	Problem dimension
NZONE	I*4	Number of zones
LZONE	I*4	Length of record type 7 in GEOM data set

3. Local Variables

Name	Type	Definition
NZ	I*4	Pointer to array ZONE
BLANK	R*8	Representation of alphameric blank data
DUMMY	R*8	Dummy

4. Functions and tasks performed by this subroutine

ZONEDR is called by FGEOM to define GEOM data set quantities NZONE and LZONE and array ZONE.

When no type 07 cards exist in A.NIP, array ZONE is filled with blanks, the number of zones, NZONE, is set to 1, and the length of record 7, LZONE, is set to 0.

Otherwise ZONEDR calls DZONE after requesting array space.

If a POINTR error occurs while requesting array space, a message is printed to indicate array overflow and return is made by the fatal error return.

5. Subprograms called by this subroutine

IGET (entry in POINTR)
DZONE
GETPNT (entry in POINTR)
IPTERR (entry in POINTR)
PUTPNT (entry in POINTR)
WIPOUT (entry in POINTR)

6. Subprograms calling this subroutine

FORMDS

7. Error messages generated by this subroutine

Fatal:

(ZONEDR) - ARRAY OVERFLOW, N2 = 'ii'

5. GNIP (NUI002) Common Block Descriptions

This section provides more detailed information about the common blocks used by the GNIP (NUI002) 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.

A special case is the /ARRA/ common block, which contains the container array A 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.

/ARRA/

Name	Type	Subprograms where values are set or modified	Definition
A(30000)	R*8	GNIP	Large container array used by POINTR

/BLNAME/

Name	Type	Subprograms where values are set or modified	Definition
BLNAME	R*8	BCDDS MAIN	Name of data block being read or processed

/CTYP02/

Name	Type	Subprograms where values are set or modified	Definition
IOFLAG	I*4	CDSCAN	GNIP output print flag 0 - suppress printing ≠0 - print edits

/CTYP03/

Name	Type	Subprograms where values are set or modified	Definition
ALPHA	R*8	TYP03A	Initial alpha or fixed alpha
IGTYPE	I*4	TYP03A	Geometry type number
IPTYPE	I*4	TYP03A	Problem type number
FIXEDK	R*8	TYP03A	Fixed k-effective
ICTYPE	I*4	TYP03A	Calculation type number

/CTYP04/

Name	Type	Subprograms where values are set or modified	Definition
IEBC(4,4)	I*4	TYPE04	External boundary condition number
NIEBC	I*4	TYPE04	Number of external boundary condition numbers given

/CTYP05/

Name	Type	Subprograms where values are set or modified	Definition
NLX	I*4	TYP05A	Number of lower 'X' direction external boundary input cards
NUX	I*4	TYP05A	Number of upper 'X' direction external boundary input cards
NLY	I*4	TYP05A	Number of lower 'Y' direction external boundary input cards
NUY	I*4	TYP05A	Number of upper 'X' direction external boundary input cards

/CTYP09/

Name	Type	Subprograms where values are set or modified	Definition
NVX	I*4	TYP09A	Number of variable-mesh 'X' direction input quantities
NVY	I*4	TYP09A	Number of variable-mesh 'Y' direction input quantities

/CTYP10/

Name	Type	Subprograms where values are set or modified	Definition
NIX	I*4	TYP10A	Number of type 10 'X' direction internal interface input cards
NIY	I*4	TYP10A	Number of type 10 'Y' direction internal interface input cards
NX	I*4	TYP11A	Number of different 'X' direction internal interface input cards
NY	I*4	TYP11A	Number of different 'Y' direction internal interface input cards

/CTYP21/

Name	Type	Subprograms where values are set or modified	Definition
ISSPEC	I*4	TYPE21	Search type specification
MAXNSP	I*4	TYPE21	Maximum number of search passes
PKOA	R*8	TYPE21	Prescribed k-effective or initial alpha
EPSLON	R*8	TYPE21	Convergence criterion epsilon for k_{eff} or α

/CTYP22/

Name	Type	Subprograms where values are set or modified	Definition
SIVX	R*8	TYPE22	Initial value of X, the search parameter
SSVX	R*8	TYPE22	Second value of X
SLBX	R*8	TYPE22	Lower bound for X
SUBX	R*8	TYPE22	Upper bound for X
SDRX	R*8	TYPE22	Derivative of k_{eff} with respect to X

/CTYP24/

Name	Type	Subprograms where values are set or modified	Definition
NMX	I*4	TYP24A	Number of 'X'-direction mesh input modifiers in criticality search
NMY	I*4	TYP24A	Number of 'Y'-direction mesh input modifiers in criticality search

/CTYP26/

Name	Type	Subprograms where values are set or modified	Definition
ALPHAM	R*8	TYPE26	Alpha modifier for a criticality search

/CTYP27/

Name	Type	Subprograms where values are set or modified	Definition
NORDER	I*4	TYP27A	Anisotropy flag for data sets ES.D1SH and ES.D2SH 0 - source is isotropic only 1 - source is anisotropic in 'X' direction for a 1-D sphere or slab - source is anisotropic in 'X' and 'Y' directions for 2-D geometry 2 - source is anisotropic in 'X' and 'Y' directions for a 1-D cylinder

/GEOM1/

Name	Type	Subprograms where values are set or modified	Definition
NDIM	I*4	GMTYPE	Number of dimensions in geometry
NGEOM	I*4	GMTYPE	Type of geometry
NPTI	I*4	MESHDR MSHORD	Number of mesh lines in 'X' direction
NPTJ	I*4	MESHDR MSHORD MESHC	Number of mesh lines in 'Y' direction
NINTI	I*4	MESHV MESHC	Number of mesh intervals in 'X' direction
NINTJ	I*4	MESHV MESHC	Number of mesh intervals in 'Y' direction
NREG	I*4	COMREG	Number of regions
NCMP	I*4	COMREG	Number of compositions
NAREA	I*4	AREADR ARIA	Number of areas defined
LAREA	I*4	AREADR ARIA	Length of record 6 in data set GEOM
NZONE	I*4	ZONEDR DZONE	Number of zones
LZONE	I*4	ZONEDR DZONE	Length of record type 7 in GEOM
NFIN	I*4	GMTYPE	Number of finite geometry type
NIL	I*4	GMTYPE	Number of integration limits
LREGA	I*4	ARIA	Length of vector NRA
LSECT	I*4	DZONE	Length of vector NSB

/IERR/

Name	Type	Subprograms where values are set or modified	Definition
ISTOP	I*4	All TYPEnn and TYPEnnA subroutines	Error return flag
IWARN	I*4	REPORT TYP13A	Warning flag
IQUIT	I*4	REPORT TYP28A	Path termination flag
IGOOF	I*4	REPORT	Warning counter

/LOCATE/

Name	Type	Subprograms where values are set or modified	Definition
LSTBLK	I*4	Module POINTR	First available location in container array
MAXBLK	I*4	Module POINTR	Size of last array allocation in container
IPRINT	I*4	TYPE02	Print edit indicator for POINTR

/NCARDS/

Name	Type	Subprograms where values are set or modified	Definition
MAXN	I*4	GNIP	Number of highest card type present in A.NIP
N(99)	I*4	GNIP	Number of cards of each type present in A.NIP

/XSCM/

Name	Type	Subprograms where values are set or modified	Definition
NGROUP	I*4	GNIP	Number of energy groups
NCMP	I*4	GNIP	Number of compositions
MINFLG	I*4	GNIP	Flag = 1 if old XS.C.MIN was read
NDUM	I*4	--	Dummy variable for alignment
CNAME(199)	R*8	GNIP	Array of composition names

6. GNIP (NUI002) Dynamic Storage Array Allocation

This section describes all arrays which may be defined by this module in common block /ARRA/ A, 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. Note that N(nn) is the number of cards of type nn present in data set A.NIP.

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
ALABEL	R*8	N(7)	TYPE07(PUTPNT)	TYP07A	Area label input to GNIP
			AREADR(WIPOUT)		
AREA	R*8	1	AREADR(PUTPNT)	AREADR	Area name with no TYPE07 card input
		N(7)	AREADR(PUTPNT)	ARIA	Input area names
AREGN	R*8	10*N(7)	TYPE07(PUTPNT)	TYP07A	Label of regions contained in area in ALABEL
			AREADR(WIPOUT)		
ATHD1	R*8	N(12)	TYPE12(PUTPNT)	TYP12A	Actual transverse half-distance in first direction
			BUCKDR(WIPOUT)		
ATHD2	R*8	N(12)	TYPE12(PUTPNT)	TYP12A	Actual transverse half-distance in second direction for a finite slab
			BUCKDR(WIPOUT)		
			TYP12A(WIPOUT)		
AXSORS	R*8	N(27)	TYPE27(PUTPNT)	TYP27A	Anisotropic 'X' direction component of surface source
			FESD1H(WIPOUT)		
			FESD2H(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
AYSORS	R*8	N(27)	TYPE27(PUTPNT)	TYP27A	Anisotropic 'Y' direction component of surface source
			FESD1H(WIPOUT)		
			FESD2H(WIPOUT)		
BD	R*8	N(5)	TYPE05(PUTPNT)	TYP05A	Boundary name on type 05 card
			FBC(PUTPNT)	FBC	Used when no type 05 cards are input
			FBC(WIPOUT)		
BCA1	R*8	N(11)	TYPE11(PUTPNT)	TYP11A	First constant input to internal interface
			FBC(WIPOUT)		
BCA2	R*8	N(11)	TYPE11(PUTPNT)	TYP11A	Third constant input to internal interface
			FBC(WIPOUT)		
BCB1	R*8	N(11)	TYPE11(PUTPNT)	TYP11A	Second constant input to internal interface
			FBC(WIPOUT)		
BCB2	R*8	N(11)	TYPE11(PUTPNT)	TYP11A	Fourth constant input to internal interface
			FBC(WIPOUT)		
BMOD1	R*8	N(25)	TYPE25(PUTPNT)	TYP25A	Buckling modifier in first transverse direction
			FSPCRT(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
BMOD2	R*8	N(25)	TYPE25(PUTPNT)	TYP25A	Buckling modifier in second transverse direction
			FSPCRT(WIPOUT)		
BSQ	R*8	NREG*NIL	BUCKDR(PUTPNT)	BUCK	Buckling by region and direction
CISORS	R*8	N(27)	TYPE27(PUTPNT)	TYP27A	Isotropic component of surface source
			FESD1H(WIPOUT)		
			FESD2H(WIPOUT)		
CLABEL	R*8	N(13)	TYPE13(PUTPNT)	TYP13A	Material label when card type 13 is input
		N(14)	TYPE14(PUTPNT)	TYP13A	Composition label when card type 14 is input
		NCMP	TYPE15(PUTPNT)	TYP15A	Composition label from data set XS.C.MIN
CMLABL	R*8	3*N(14)	TYPE14(PUTPNT)	TYP13A	If type 13 cards present, material label
			TYPE14(WIPOUT)		If no type 13 card present, isotope label
CMVOLF	R*8	3*N(14)	TYPE14(PUTPNT)	TYP13A	If type 13 cards present, volume fraction of CMLABL
			TYPE14(WIPOUT)		If no type 13 cards present, atom density of CMLABL
CNAME	R*8	N(14)	CRCDR(PUTPNT)	COMREG	Composition names

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
CORDIR	R*8	N(28)	TYPE28(PUTPNT)	TYP28A	Coordinate direction of face containing surface source
			FESD1H(WIPOUT)		
			FESD2H(WIPOUT)		
CRLABL	R*8	N(15)	TYPE15(PUTPNT)	TYP15A	Composition labels of composition assigned to region CREGN
			CRCDR(WIPOUT)		
CREGN	R*8	10*N(15)	TYPE15(PUTPNT)	TYP15A	Labels of region containing compositions specified in CRLABL
			CRCDR(WIPOUT)		
C1	R*8	NBC	FBC(PUTPNT)	EXTBND	First constant in output data set BC
			FBC(WIPOUT)	INTBND	
C2	R*8	NBC	FBC(PUTPNT)	EXTBND	Second constant in output data set BC
			FBC(WIPOUT)	INTBND	
C3	R*8	NBC	FBC(PUTPNT)	EXTBND	Third constant in output data set BC
			FBC(WIPOUT)	INTBND	
C4	R*8	NBC	FBC(PUTPNT)	EXTBND	Fourth constant in output data set BC
			FBC(WIPOUT)	INTBND	

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
DIMOD	R*8	NINTI	FSPCRT(PUTPNT)	CRITSH	'X' direction modifiers for dimension searches
			FSPCRT(WIPOUT)		
DJMOD	R*8	NINTJ	FSPCRT(PUTPNT)	CRITSH	'Y' direction modifiers for dimension searches
			FSPCRT(WIPOUT)		
DLABEL	R*8	N(12)	TYPE12(PUTPNT)	TYP12A	Input region label
			BUCKDR(WIPOUT)		
DMODB	R*8	NREG*NIL	FSPCRT(PUTPNT)	BUCSRH	Buckling modifiers
			FSPCRT(WIPOUT)		
DNCORD	R*8	N(28)	TYPE28(PUTPNT)	TYP28A	Lower end coordinate of the specified surface source face
			FESD1H(WIPOUT)		
			FESD2H(WIPOUT)		
EBCA	R*8	N(5)	TYPE05(PUTPNT)	TYP05A	First input constant to external boundary
		N(5)	FBC(PUTPNT)		Used when no type 05 input exists
			FBC(WIPOUT)		
EBCB	R*8	N(5)	TYPE05(PUTPNT)	TYP05A	Second input constant to external boundary
		N(5)	TYPE05(PUTPNT)		Used when no type 05 input exists
			FBC(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
FSEXT	R*8	NINTI*NGROUP	FESD1D(PUTPNT)	I1SORD	1-D external inhomogeneous surface sources
			FESD1D(WIPOUT)		
	R*8	NINTI*NINTJ	FESD2D(PUTPNT)	I2SORD	2-D external inhomogeneous surface sources
			FESD2D(WIPOUT)		
H	R*8	NIL*NREG	BUCKDR(PUTPNT)	BUCK	Actual half-height for each region, in each finite direction
			DSGEM(WIPOUT)		
HSUR1	R*8	NPTJ*NGROUP	FESD1H(PUTPNT)	INITZE	Isotropic component of surface source on 'X' direction face
			FESD2H(PUTPNT)	ANISO1	
			FESD1H(WIPOUT)	ANISO2	
			FESD2H(WIPOUT)		
HSUR2	R*8	NPTJ*NGROUP	FESD1H(PUTPNT)	INITZE	Anisotropic 'X' direction component of 'X' direction surface source
			FESD2H(PUTPNT)	ANISO1	
			FESD1H(WIPOUT)	ANISO2	
			FESD2H(WIPOUT)		
HSUR3	R*8	NPTJ*GROUP	FESD1H(PUTPNT)	INITZE	Anisotropic 'Y' direction component of 'X' direction surface source
			FESD2H(PUTPNT)	ANISO1	
			FESD1H(WIPOUT)	ANISO2	
			FESD2H(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
IAHEGN	I*4	N(27)	TYPE27(PUTPNT)	TYP27A	Higher-energy group number to which given surface source applies
			FESD1H(WIPOUT)		
			FESD2H(WIPOUT)		
IALEGN	I*4	N(27)	TYPE27(PUTPNT)	TYP27A	Lower-energy group number to which surface source applies
			FESD1H(WIPOUT		
			FESD2H(WIPOUT)		
IBHEGN	I*4	N(11)	TYPE11(PUTPNT)	TYP11A	Higher-energy group number to which internal boundary input constants apply
			FBC(WIPOUT)		
IBLEGN	I*4	N(11)	TYPE11(PUTPNT)	TYP11A	Lower-energy group number to which internal boundary input constants apply
			FBC(WIPOUT)		
ICONX	I*4	N(10)	TYPE10(PUTPNT)	TYP10A	Internal condition of 'X' direction interface
			FBC(WIPOUT)		
			TYPE10(WIPOUT) if unused		
		NIX	TYPE10(REDEF)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
ICONY	I*4	N(10)	TYPE10(PUTPNT)	TYP10A	Internal condition of 'Y' direction interface
			FBC(WIPOUT)		
			TYPE10(WIPOUT) if unused		
		NIX	TYPE10(REDEF)		
IFLAG	I*4	NINTI	FESD1D(PUTPNT)	I1SORD	Output 1-D fission spectrum weighting flag for external source
			FESD2D(WIPOUT)		
	I*4	NINTI*NINTJ	FESD2D(PUTPNT)	I2SORD	Output 2-D fission spectrum weighting flag for external source
			FESD2D(WIPOUT)		
IHEGN	I*4	N(5)	TYP05(PUTPNT)	TYP05A	Input higher energy group number to which constants apply on external boundary
			FBC(WIPOUT)		
IHORZ	I*4	NPTJ*NINTI	FESD1H(PUTPNT)	INITZE	Pointer to position of surface source in source arrays HSUR1, HSUR2, HSUR3
			FESD2H(PUTPNT)	ANISO1	
			FESD1H(WIPOUT)	ANISO2	
			FESD2H(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
IICODE	I*4	N(19)	TYPE19(PUTPNT)	TYP19A	Input type code for external source data
			FESD1D(WIPOUT)		
			FESD2D(WIPOUT)		
ILEGN	I*4	N(5)	TYPE05(PUTPNT)	TYP05A	Input lower-energy group number to which constants apply on external boundary
			FBC(WIPOUT)		
ISHEGN	I*4	N(19)	TYPE19(PUTPNT)	TYP19A	Input higher-energy group number to which external source applies
			FESD1D(WIPOUT)		
			FESD2D(WIPOUT)		
ISLEGN	I*4	N(19)	TYPE19(PUTPNT)	TYP19A	Input lower-energy group number to which external source applies
			FESD1D(WIPOUT)		
			FESD2D(WIPOUT)		
ISPECN	I*4	N(19)	TYPE19(PUTPNT)	TYP19A	Input fission spectrum weighting flag for external source
			FESD1D(WIPOUT)		
			FESD2D(WIPOUT)		
IVERT	I*4	NPTI*NINTJ	FESD1H(PUTPNT)	INITZE	Pointer to position of surface source in source arrays VSUR1, VSUR2, VSUR3
			FESD2H(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
MC	I*4	NINTI*NINTJ	RCMDR(PUTPNT)	RCMESH	Composition index for each mesh interval
MCOMPN	I*4	N(23)	FSPCRT(PUTPNT)	CONSCH	Modifies composition on number for a concentration search
			FSPCRT(WIPOUT)		
MFACE	I*4	NBC	FBC(PUTPNT)	EXTBND	Mesh line number and side indicator of boundary to which record applies for record type 2 in BC
			FBC(WIPOUT)	INTBND	
MR	I*4	NINTI*NINTJ	RCMDR(PUTPNT)	RCMESH	Region index for each mesh interval
M1	I*4	NBC	FBC(PUTPNT)	EXTBND	Mesh line number defining beginning of interval along face MFACE
			FBC(WIPOUT)	INTBND	
M2	I*4	NBC	FBC(PUTPNT)	EXTBND	Mesh line number defining end of interval along face MFACE
JI	I*4	NBC	FBC(PUTPNT)	EXTBND	Output initial higher-energy group number of which set of boundary condition constants applies
			FBC(WIPOUT)	INTBND	

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
JF	I*4	NBC	FBC(PUTPNT)	EXTBND	Output final lower-energy number to which set of boundary condition constants applies
			FBC(WIPOUT)	INTBND	
NBIT	I*4	NBC	FBC(PUTPNT)	EXTBND	Type number of boundary or interface condition
			FBC(WIPOUT)		
NC	I*4	N(6)	CRCDR(PUTPNT)	COMREG	Composition index of regions
NG1	I*4	N(19)	FESD2D(PUTPNT)	GRPSOR	Initial group number for which given external source applies
			FESD2D(WIPOUT)		
NG2	I*4	N(19)	FESD2D(PUTPNT)	GRPSOR	Final group number for which given external source applies
			FESD2D(WIPOUT)		
NINTX	I*4	N(6)	TYPE06(PUTPNT)	TYP06A	Number of intervals inserted in region in 'X' direction
			MESHDR(WIPOUT)		
NINTY	I*4	N(6)	TYPE06(PUTPNT)	TYP06A	Number of intervals inserted in region in 'Y' direction
			MESHDR(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
NINVX	I*4	3*N(9)	TYPE09(WIPOUT) if used	TYP09A	Number of intervals inserted in variable mesh in 'X' direction
		NVX	TYPE09(REDEF) if used		
			MESHDR(WIPOUT)		
NINVY	I*4	3*N(9)	TYPE09(PUTPNT)	TYP09A	Number of intervals inserted in variable mesh in 'Y' direction
			TYPE09(WIPOUT if unused		
		NVY	TYPE09(REDEF) if used		
			MESHDR(WIPOUT)		
NRA	I*4	10*N(7)	AREADR(PUTPNT)	ARIA	Region index of portion of area
NREGA	I*4	N(7)	AREADR(PUTPNT)	ARIA	Number of regions in each area
NMC	I*4	NCMP	FSPCRT(PUTPNT)	CONSCH	Modifier compositions for given compositions
			FSPCRT(WIPOUT)		
NSECTZ	I*4	N(8)	ZONEDR(PUTPNT)	DZONE	Number of sections in each zone
NSB	I*4	N(8)*2*NDIM	ZONEDR(PUTPNT)	DZONE	Section boundaries
NUMC	I*4	N(15)	CRCDR(PUTPNT)	COMREG	Temporary composition number
			RCMDR(WIPOUT)		
NUMR	I*4	N(6)	CRCDR(PUTPNT)	COMREG	Temporary region number
			RCMDR(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
ONCORD	R*8	N(28)	TYPE28(PUTPNT)	TYP28A	'Y' coordinate along which 'X' direction face lies
			FESD1H(WIPOUT)		'X' coordinate along which 'Y' direction face lies
			FESD2H(WIPOUT)		
REG	R*8	N(6)	CRCDR(PUTPNT)	COMREG	Region names
REGVOL	R*8	NREG	FGEOM(PUTPNT)	REC10	Region volumes
			DSGEM(WIPOUT)		
RIADEN	R*8	3*N(13)	TYPE13(PUTPNT)	TYP13A	Type 13 card atom density
			TYPE13(WIPOUT)		
		3*N(14)	TYPE14(PUTPNT)	TYP13A	Volume fraction or, if no type 13 cards, material label
			TYPE14(WIPOUT)		
RILABL	R*8	3*N(13)	TYPE13(PUTPNT)	TYP13A	Isotope label from card 13
			TYPE14(WIPOUT) if no type 13 cards		
		3*N(14)	TYPE14(PUTPNT)		Material label or, if no type 13 cards, isotope label
			TYPE14(WIPOUT)		
RLABEL	R*8	N(6)	TYPE06(PUTPNT)	TYP06A	Input region label for region definition
			CRCDR(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
RMLABL	R*8	N(13)	TYPE13(PUTPNT)	TYP13A	Material label
RLCOB	R*8	N(5)	TYPE05(PUTPNT)	TYP05A	Input lower end coordinate on external boundary
			FBC(PUTPNT) if no type 05 cards		
			FBC(WIPOUT)		
RLISC	R*8	N(19)	TYPE19(PUTPNT)	TYP19A	Lower 'X' direction coordinate of mesh interval containing given source
			FESD1D(WIPOUT)		
			FESD2D(WIPOUT)		
RUCOB	R*8	N(5)	TYPE05(PUTPNT)	TYP05A	Input upper end coordinate on external boundary
			FBC(PUTPNT) if no type 05 cards		
			FBC(WIPOUT)		
RUISC	R*8	N(19)	TYPE19(PUTPNT)	TYP19A	Lower 'Y' direction coordinate of mesh interval containing given source
			FESD1D(WIPOUT)		
			FESD2D(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
SALABL	R*8	N(28)	TYPE28(PUTPNT) FESD1H(WIPOUT) FESD2H(WIPOUT)	TYP28A	Inhomogeneous surface source label
SILABL	R*8	N(19)	TYPE19(PUTPNT) FESD1D(WIPOUT) FESD2D(WIPOUT)	TYP19A	Input label of entity to which input source data applies
SLABLR	R*8	N(25)	TYPE25(PUTPNT) FSPCRT(WIPOUT)	TYP25A	Region label to which buckling modifier applies
SLABL1	R*8	N(23)	TYPE23(PUTPNT) FSPCRT(WIPOUT)	TYP23A	Composition label of modifier composition
SLABL2	R*8	10*N(23)	TYPE23(PUTPNT) FSPCRT(WIPOUT)	TYP23A	Composition label of compositions with which modifier is used
SOURCE	R*8	N(19)	TYPE19(PUTPNT) FESD1D(WIPOUT) FESD2D(WIPOUT)	TYP19A	Input isotropic external source value

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
SSLABL	R*8	N(27)	TYPE27(PUTPNT) FESD1H(WIPOUT) FESD2H(WIPOUT)	TYP27A	Inhomogeneous surface source label
TITLE	R*8	11*N(1)	TYPE01(PUTPNT)	TYP01A	Problem title
TEXD1	R*8	N(12)	TYPE12(PUTPNT) BUCKDR(WIPOUT)	TYP12A	Transverse extrapolation distance in first direction
TEXD2	R*8	N(12)	TYPE12(PUTPNT) TYP12A(WIPOUT) if not used BUCKDR(WIPOUT)	TYP12A	Transverse extrapolation distance in second direction (for a finite slab)
UCX	R*8	3*N(9)	TYPE09(PUTPNT) TYPE09(WIPOUT) if not used	TYP09A	Upper 'X' coordinate of space being defined in variable mesh
		NVX	TYPE09(REDEF) MESHDR(WIPOUT)		
UCY	R*8	3*N(9)	TYPE09(PUTPNT) TYPE09(WIPOUT) if not used	TYP09A	Upper 'Y' coordinate of space being defined in variable mesh
		NVY	TYPE09(REDEF) MESHDR(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
UPCORD	R*8	N(28)	TYPE28(PUTPNT)	TYP28A	Upper end coordinate of specified surface source face
			FESD1H(WIPOUT)		
			FESD2H(WIPOUT)		
VSUR1	R*8	NPTI*NGROUP	FESD2H(PUTPNT)	INITZE	Isotropic 'Y' direction surface source component
			FESD2H(WIPOUT)	ANISO2	
VSUR2	R*8	NPTI*NGROUP	FESD2H(PUTPNT)	INITZE	Anisotropic 'X' direction component of 'Y' direction surface source
			FESD2H(WIPOUT)	ANISO2	
VSUR3	R*8	NPTI*NGROUP	FESD2H(PUTPNT)	INITIZE	Anisotropic 'Y' direction component of 'Y' direction surface source
			FESD2H(WIPOUT)	ANISO2	
X	R*8	NIL*NREG	BUCKDR(PUTPNT)	BUCK	Extrapolated half-height for each region in each finite direction
XILABL	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	Label of 'X' direction internal interface
			TYPE10(WIPOUT) if not used		
		NIX	TYPE10(REDEF)		
			FBC(WIPOUT)		
XL	R*8	N(6)	TYPE06(PUTPNT)	TYP06A	Lower 'X' coordinate defining left side of a region
			RCMDR(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or mofified	Definition
XLEI	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	Lower end coordinate of 'X' direction interface
			TYPE10(REDEF)		
			TYPE10(WIPOUT) if not used		
			FBC(WIPOUT)		
XMESH	R*8	NPTI	MESHDR(PUTPNT)	MESHC	'X' direction coordinate of constant mesh output in GEOM data set
				MESHORD	
				MESHV	'X' direction coordinate of variable mesh
XMESHM	R*8	N(24)	TYPE24(PUTPNT)	TYP24A	'X' direction mesh modifier for a dimension search
			TYPE24(WIPOUT) if not used		
		NMX	TYPE24(REDEF)		
			FSPCRT(WIPOUT)		
XMODLC	R*8	N(24)	TYPE24(PUTPNT)	TYP24A	'X' direction lower coordinate of range to be modified
		NMX	TYPE24(REDEF)		
			FSPCRT(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
XMODUC	R*8	N(24)	TYPE24(PUTPNT)	TYP24A	'X' direction upper coordinate of range to be modified
			TYPE24(WIPOUT) if not used		
		NMX	TYPE24(REDEF)		
			FSPCRT(WIPOUT)		
XU	R*8	N(6)	TYPE06(PUTPNT)	TYP06A	Upper 'X' coordinate defining right side of region
			RCMDR(WIPOUT)		
XUEI	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	Upper end coordinate of 'X' direction interface
			TYPE10(REDEF)		
			TYPE10(WIPOUT) if not used		
			FBC(WIPOUT)		
XYIC	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	'Y' direction coordinate along which 'X' direction interface lies
			TYPE10(REDEF)		
			TYPE10(WIPOUT) if not used		
			FBC(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
YMODLC	R*8	N(24)	TYPE24(PUTPNT)	TYP24A	'Y' direction lower coordinate of range to be modified
			TYPE24(WIPOUT) if not used		
		NMY	TYPE24(REDEF)		
			FSPCRT(WIPOUT)		
YMODUC	R*8	N(24)	TYPE24(PUTPNT)	TYP24A	'Y' direction upper coordinate of range to be modified
			TYPE24(WIPOUT) if not used		
		NMY	TYPE24(REDEF)		
			FSPCRT(WIPOUT)		
YU	R*8	N(6)	TYPE06(PUTPNT)	TYP06A	Upper 'Y' coordinate defining top of region
			RCMDR(WIPOUT)		
YUEI	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	Upper end coordinate along which 'Y' direction interface lies
		NIY	TYPE10(REDEF)		
			FBC(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
YXIC	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	'X' direction coordinate along which 'Y' direction interface lies
			TYPE10(WIPOUT) if not used		
		NIY	TYPE10(REDEF)		
			FBC(WIPOUT)		
YILABL	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	Label of 'Y' direction internal interface
			TYPE10(WIPOUT)		
		NIY	TYPE10(REDEF)		
			FBC(WIPOUT)		
YL	R*8	N(6)	TYPE06(PUTPNT)	TYP06A	Lower 'Y' coordinate defining bottom of region
			RCMDR(WIPOUT)		
YLEI	R*8	N(10)	TYPE10(PUTPNT)	TYP10A	Lower end coordinate along which 'Y' direction interface lies
			TYPE10(WIPOUT) if not used		
		NIY	TYPE10(REDEF)		
			FBC(WIPOUT)		

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
YMESHM	R*8	N(24)	TYPE24(PUTPNT)	TYP24A	'Y' direction mesh modifier for a dimension search
			TYPE24(WIPOUT) if not used		
		NMY	TYPE24(REDEF)		
			FSPCRT(WIPOUT)		
ZLABEL	R*8	N(8)	TYPE09(PUTPNT)	TYP08A	Input zone definition label
			ZONEDR(WIPOUT)		
ZONE	R*8	N(8)	ZONEDR(PUTPNT)	DZONE	Zone names output to GECM data set
		1	ZONEDR(PUTPNT)	DZONE	Used when N(8)=0
ZXL	R*8	N(8)	TYPE08(PUTPNT)	TYP08A	Coordinate of lower 'X' boundary of section of zone
			ZONEDR(WIPOUT)		
ZXU	R*8	N(8)	TYPE08(PUTPNT)	TYP08A	Coordinate of upper 'X' boundary of section of zone
			ZONEDR(WIPOUT)		
ZYL	R*8	N(8)	TYPE08(PUTPNT)	TYP08A	Coordinate of lower 'Y' boundary of section of zone
			ZONEDR(WIPOUT)		
ZYU	R*8	N(8)	TYPE08(PUTPNT)	TYP08A	Coordinate of upper 'Y' boundary of section of zone
			ZONEDR(WIPOUT)		

7. GNIP (NUI002) 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. the names of subprograms in which the data set is read and/or written, along with a list of the record types which are read or written within each subprogram and the name of the variable which contains the reference number of the data set.

Complete descriptions 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.NIP, General Input for Neutronics Calculations

Record structure - ARC System data set

Read in subprograms	Card types	Reference variable
GNIP	BCDDS specification record	N1
SPACE	Spaces past unneeded cards	
TYP01A	Problem Title (type 01)	
TYPE02	Input Processing Specifications (type 02)	
TYPE03	Basic Problem Identification (type 03)	
TYPE04	External Boundary Conditions (type 04)	
TYP05A	External Boundary Condition Constants (type 05)	
TYP06A	Region Boundary Coordinates and Constant Mesh Structure (type 06)	
TYP07A	Area Specifications (type 07)	
TYP08A	Zone Specifications (type 08)	
TYP09A	Variable Mesh Structure (type 09)	
TYP10A	Internal Boundary or Interface Conditions (type 10)	
TYP11A	Internal Boundary or Interface Condition Constants (type 11)	
TYP12A	Finite Geometry Transverse Distances (type 12)	
TYP13A	Material Specifications (type 13) Composition Specifications (type 14)	
TYP15A	Composition and Region Assignments (type 15)	
TYP19A	Distributed Isotropic Inhomogeneous Source Data (type 19)	
TYPE21	Search Specifications (type 21)	

Read in subprograms	Card types	Reference variable
TYPE22	Search Parameters (type 22)	
TYP23A	Composition Modifiers for Criticality Search (type 23)	
TYP24A	Mesh Modifiers for Criticality Search (type 24)	
TYP25A	Buckling Modifiers for Criticality Search (type 25)	
TYPE26	Alpha Modifier for Criticality Search (type 26)	
TYP27A	Inhomogeneous Surface Source Spectrum Specification (type 27)	
TYP28A	Inhomogeneous Source Geometry Data (type 28)	

Written in subprograms - None

Name or description of data set - BC, Boundary Condition Specifications

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
FBC	One of Specifications (type 1)	N1
BCRITE	NBC of Boundary Conditions (type 2)	N1

Name or description of data set - ES.D1D, One-dimensional Distributed External Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
I1SORD	One of Specifications (type 1)	N1
	One of One-dimensional Distributed External Source (type 2)	

Name or description of data set - ES.D1SH, One-dimensional Surface Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
ANISO1	One of Specifications (type 1)	NVOL
	One of Surface Source Mesh Organization (type 2)	
	One of One-dimensional Surface Source Components (type 3)	

Name or description of data set - ES.D2D, Two-dimensional Distributed External Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
I2SORD	One of Specifications (type 1)	N1
	NGROUP of Two-dimensional Distributed External Source (type 2)	

Name or description of data set - ES.D2SH, Two-dimensional Surface Source

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
ANISO2	One of Specifications (type 1)	NVOL
	One of Surface Source Mesh Organization (type 2)	
	One of Two-dimensional Surface Source Components (type 3)	

Name or description of data set - FA.D1, One-dimensional Adjoint Group Fluxes

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
FFAD1	One of Specifications (type 1)	NFA
	One of One-dimensional Adjoint Group Fluxes (type 2)	

Name or description of data set - FA.D2, Two-dimensional Adjoint Group Fluxes

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
FFAD2	One of Specifications (type 1)	NFA
	NGROUP of Two-dimensional Adjoint Group Fluxes (type 2)	

Name or description of data set - FR.D1, One-dimensional Real Group Fluxes

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
FFRD1	One of Specifications (type 1)	NFR
	One of One-dimensional Real Group Fluxes (type 2)	

Name or description of data set - FR.D2, Two-dimensional Real Group Fluxes

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
FFRD2	One of Specifications (type 1)	NFR
	NGROUP of Two-dimensional Real Group Fluxes (type 2)	

Name or description of data set - GEOM, Geometry Data

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
FGEOM	One of Specifications (type 1)	N1

Written in subprograms	Record types	Reference variable
DSGEM	One of Specifications (type 1)	N1
	One of Boundaries (type 2)	
	One of Region - Interval Correspondence (type 3)	
	One of Composition - Interval Correspondence (type 4)	

Written in subprograms	Record types	Reference variable
	One of Composition - Region Correspondence (type 5)	
	One of Area Definitions (type 6) if NAREA>0	
	One of Zone Definitions (type 7) if NZONE>0	
	One of Finite Geometry Data (type 8) if NIL>0	
	One of Labels (type 9)	
REC10	One of Region Volumes (type 10)	N1

Name or description of data set - SP.CICN, Module-independent Data

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
FSPCCN	One of Specifications (type 1)	M

Name or description of data set - SP.CRIT, Criticality Search Specifications

Record structure - ARC System data set

Read in subprograms - None

Written in subprograms	Record types	Reference variable
FSPCRT	One of Specifications (type 1)	M
CONSCH	One of Concentration Search Modifiers (type 2)	M
CRITSH	One of Dimension Search Modifiers (type 3)	M

Written in subprograms	Record types	Reference variable
BUCSRH	One of Buckling Search Modifiers (type 4) Only one of record types 2, 3, or 4 will be written	M

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
GNIP	One of Specifications (type 1)	N1

Written in subprograms - None

Name or description of data set - XS.ISO, Microscopic Group Cross-sections

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
GNIP	One of Specifications (type 1)	N1

Written in subprograms - None

Name or description of data set - XS.ISO2, Problem-dependent Microscopic Group Cross-sections

Record structure - ARC System data set

Read in subprograms	Record types	Reference variable
GNIP	One of Specifications (type 1)	N1

Written in subprograms - None

8. GNIP (NUI002) Error Messages

This section lists the error messages generated by the GNIP module by subprogram name, where the subprogram names appear in the same order as in section 4.

Where messages are listed with letters inside quotation marks, 'aaaaaa' indicates alphameric characters, 'x.xx' indicates a floating point number, and 'ii' indicates an integer.

The symbol N2 represents the number of POINTR container array space requests in a given subprogram. N2 is also used to indicate the index number of the card being read in the TYPEnn subroutine. This is fully explained in the TYP01A subroutine description.

A fatal error is one which causes card processing to halt immediately; an error return is made to the calling program.

A serious error does not interfere with card processing but data sets cannot be written; after all card images are processed, an error return is made to the calling program.

A minor error is one which does not affect card processing but is simply noted for the user's benefit.

AREADR

Fatal:

ARRAY OVERFLOW - SUBROUTINE AREADR, N2 = 'ii'

BUCKDR

Fatal:

ARRAY OVERFLOW - SUBROUTINE BUCKDR, N2 = 'ii'

(BUCKDR) - NO TYPE 12 CARDS FOR FINITE GEOMETRY WERE INPUT.

BUCSRH

Fatal:

(BUCSRH) - NUMBER OF INTEGRATION LIMITS IS NOT CONSISTENT WITH GEOMETRY TYPE.

CDSCAN

Fatal:

(CDSCAN) - TYPE 21 AND 22 CARDS WERE SUPPLIED IN A.NIP, BUT NO CARDS OF TYPES 23, 24, 25, OR 26 WERE SUPPLIED.

COMREG

Fatal:

(COMREG) - NO REGION LABEL ON TYPE 15 CARDS AGREES WITH REGION LABEL 'aaaaaa' ON TYPE 06 CARDS.

CRCDR

Fatal:

ARRAY OVERFLOW - SUBROUTINE CRCDR, N2 = 'ii'

CRITSH

Fatal:

(CRITSH) - THERE IS NO MESH POINT CORRESPONDING TO THE 'ii' CARD-(TYPE 24).

DZONE

Fatal:

(DZONE) - COORDINATE IN ZONE 'aaaaaa' DOES NOT CORRESPOND TO ANY MESH LINE.

EXTBND

Fatal:

(EXTBND) - TYPE 6 BOUNDARY CONDITION NOT IMPLEMENTED.

(EXTBND) - INVALID BOUNDARY CONDITION TYPE NUMBER 'ii' GIVEN ON TYPE 04 CARD. WHEN NO TYPE 05 CARDS ARE PRESENT, TYPE NUMBER MUST BE 2, 3, 6, 9, 10, 11, OR 12.

FACEN

Fatal:

(FACEN) - THE INPUT IN COLS. 20-36 ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH THE MESH STRUCTURE.

(FACEN) - THE INPUT IN COLS. 37-48 ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH THE MESH STRUCTURE.

(FACEN) - THE INPUT IN COLS. 49-60 ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH THE MESH STRUCTURE.

FBC

Fatal:

ARRAY OVERFLOW - IN FBC, N2 = 'ii'

(FBC) - THERE ARE NO BOUNDARY CONDITION NUMBERS GIVEN - TYPE 04 CARD MAY NOT BE CORRECT.

FESD1D

Fatal:

ARRAY OVERFLOW - IN FESD1D, N2 = 'ii'

FESD1H

Fatal:

(FESD1H) - ARRAY OVERFLOW. N2 = 'ii'

FESD2D

Fatal:

ARRAY OVERFLOW - IN FESD2D, N2 = 'ii'

FESD2H

Fatal:

(FESD2H) - ARRAY OVERFLOW. N2 = 'ii'

FGEOM

Fatal:

(FGEOM) - OLD AND NEW GEOM QUANTITIES NUMBER 'iiii' IN FIRST RECORD DO NOT AGREE, ERROR IS ASSUMED.

FORMDS

Fatal:

(FORMDS) - ERROR IN SNIFF. SUB-BLOCK = 'aaaaaaaa' WAS LOCATED IN DSNAME ARRAY BUT DOES NOT YET EXIST. ON CALL TO SNIFF TO OBTAIN DATA SET REFERENCE NUMBER TO WRITE THE DATA SET, ITS NAME COULD NOT BE FOUND IN DSNAME.

(FORMDS) - ERROR IN SNIFF. SUB-BLOCK = 'aaaaaaaa' WAS LOCATED IN DSNAME ARRAY BUT DOES NOT YET EXIST. ON CALL TO SNIFF TO OBTAIN DATA SET REFERENCE NUMBER TO WRITE THE DATA SET, A REFERENCE NUMBER OF ZERO WAS RETURNED.

FSPCRT

Fatal:

ARRAY OVERFLOW - SUBROUTINE FSPCRT, N2 = 'ii'

(FSPCRT) - THERE IS NO SEARCH DATA GIVEN - NO CARDS OF TYPES 23, 24, 25, OR 26 ARE PRESENT.

GNIP

Fatal:

(GNIP) - ERROR IN SNIFF. CANNOT MATCH DSNAME ENTRY BLOCK = 'aaaaaaaa', SUB-BLOCK = 'aaaaaaaa'.

(GNIP) - READ ERROR. ERROR IN DATA TRANSFER, DSNAME = 'aaaaaaaa'.

(GNIP) - XS.C.MIN HAS BEEN DECLARED AS OLD FOR GEOMETRY/NEUTRONICS INPUT PREPARATION BUT DOES NOT EXIST.

(GNIP) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. DSNAME = 'aaaaaaaa'

(GNIP) - INPUT ERRORS HAVE MADE IT IMPOSSIBLE TO WRITE OUTPUT DATA SETS. TIME IN NUI002 WAS 'x.xx' MINUTES.

(GNIP) - A.NIP IS NEEDED FOR GEOMETRY/NEUTRONICS INPUT PREPARATION BUT DOES NOT EXIST.

I1SORD

Fatal:

(I1SORD) - DATA ON CARD NUMBER 'ii' OF TYPE 19 IN A.NIP DOES NOT MATCH MESH STRUCTURE.

I2SORD

Fatal:

(I2SORD) - DATA ON CARD NUMBER 'ii' OF TYPE 19 IN A.NIP DOES NOT MATCH MESH STRUCTURE.

INTBND

Fatal:

(INTBND) - COORDINATE INPUT LINE NUMBER 'ii' ON CARD TYPE 10 DOES NOT AGREE WITH MESH STRUCTURE.

(INTBND) - ERROR ON CARD NUMBER 'ii' OF TYPE 11, 2D BOUNDARY CONSTANTS CANNOT APPLY TO ALL GROUPS WHEN INTERNAL BOUNDARY EXTENDS THE WIDTH OF THE REACTOR.

(INTBND) - MATRIX 'ii' HAS NO INVERSE.

MESHDR

Fatal:

ARRAY OVERFLOW - SUBROUTINE MESHDR, N2 = 'ii'

RCMDR

Fatal:

ARRAY OVERFLOW - SUBROUTINE RCMDR, N2 = 'ii'

RCMESH

Fatal:

(RCMESH) - MESH 'ii', 'ii' DID NOT GET A REGION ASSIGNMENT - CHECK INPUT TO A.NIP.

(RCMESH) - ERROR IN REGION SPECIFICATIONS HAS MADE IT POSSIBLE TO CONTINUE WITH FGEOM.

ERROR XL, 'K', 'I', 'J', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'
ERROR XU, 'K', 'I', 'J2', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'
ERROR YL, 'K', 'I', 'J', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'
ERROR YU, 'K', 'I', 'J2', 'XL(K)', 'XU(K)', 'YL(K)', 'YU(K)'

The above 4 messages contain the values of the variables given in quotation marks, as defined in subroutine RCMESH.

REPORT

Fatal:

(REPORT) - AN ERROR HAS OCCURRED DURING A CALL TO POINTER WHILE READING CARD TYPE 'ii'. NO ATTEMPT TO READ FURTHER WILL BE MADE.

(REPORT) - AN ERROR OCCURRED WHILE READING CARD TYPE 'ii' WHICH MAKES IT IMPOSSIBLE TO CONTINUE READING - RETURN WILL BE MADE TO THE CALLING PATH. EXECUTION WILL BE TERMINATED.

Serious:

(REPORT) - AN ERROR DETECTED WHILE READING CARD TYPE 'ii' HAS MADE IT IMPOSSIBLE TO WRITE GNIP OUTPUT DATA SETS.

Minor:

(REPORT) - MINOR ERROR NUMBER 'ii' TYPE 'ii' CARD.

SPACE

Fatal:

(SPACE) - CARD SUPPOSEDLY OF TYPE 'ii' FOUND TO BE OF TYPE 'ii' IN A.NIP.

(SPACE) - DISCREPANCY IN NUMBER OF TYPE 'ii' CARDS. N2 = 'ii' BUT NUMBER OF CARDS TO BE SPACED PAST = 'ii'

(SPACE) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(SPACE) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

TEST

Fatal:

(TEST) - THERE IS NO MESH POINT WHICH CORRESPONDS WITH INPUT COORDINATE 'xx.xxxxx'

TYPE01

(TYPE01) - ARRAY OVERFLOW, N2 = 'ii'

TYP01A

Fatal:

(TYP01A) - CARD SUPPOSEDLY OF TYPE 01 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP01A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED, N2 = 'ii'

(TYP01A) - READ ERROR. ERROR IN DATA TRANSFER.

TYPE02

Fatal:

(TYPE02) - CARD SUPPOSEDLY OF TYPE 02 FOUND TO BE OF TYPE 'ii' in A.NIP.

(TYPE02) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED, N2 = 'ii'

(TYPE02) - READ ERROR. ERROR IN DATA TRANSFER, N2 = 'ii'

TYPE03

Fatal:

(TYPE03) - CARD SUPPOSEDLY OF TYPE 03 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYPE03) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED, N2 = 'ii'

(TYPE03) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYPE03) - NO TYPE 03 CARD SUPPLIED IN A.NIP.

(TYPE03) - INVALID GEOMETRY TYPE OF 'ii' GIVEN IN FIRST 03 CARD IN A.NIP.

(TYPE03) - INVALID CALCULATION TYPE OF 'ii' GIVEN ON FIRST 03 CARD IN A.NIP.

(TYPE03) - INVALID PROBLEM TYPE OF 'ii' GIVEN ON FIRST 03 CARD IN A.NIP.

(TYPE03) - INVALID PROBLEM TYPE OF 2 GIVEN FOR A CALCULATION TYPE OF 1 ON FIRST 03 CARD IN A.NIP.

TYPE04

Fatal:

(TYPE04) - NO TYPE 04 CARD SUPPLIED IN A.NIP.

(TYPE04) - CARD SUPPOSEDLY OF TYPE 04 IN A.NIP FOUND TO BE OF TYPE 'ii'

(TYPE04) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'.

(TYPE04) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYPE04) - INVALID BOUNDARY CONDITION OF 'ii' GIVEN ON THE FIRST 04 CARD IN A.NIP.

(TYPE04) - PERIODIC BOUNDARY CONDITIONS MUST OCCUR IN PAIRS ON OPPOSITE SIDES ON THE REACTOR. SEE BOUNDARY CONDITION VALUES 'ii' AND 'ii' ON TYPE 04 IN A.NIP.

(TYPE04) - X-Y GEOMETRIES DO NOT ALLOW WHITE BOUNDARY CONDITIONS - CARD TYPE 04 IN A.NIP.

(TYPE04) - BOUNDARY CONDITION TYPE NUMBERS ARE NOT ALL EITHER TRANSPORT OR DIFFUSION TYPE.

(TYPE04) - IN R-Z GEOMETRY, A WHITE BOUNDARY IS ONLY ALLOWED ON THE RIGHT HAND SIDE.

(TYPE04) - LEFT AND RIGHT BOUNDARIES CANNOT BE PERIODIC IN R-Z GEOMETRY.

TYPE05

Fatal:

(CTYP05) - ARRAY OVERFLOW. N2 = 'ii'

TYP05A

Fatal:

(TYP05A) - CARD SUPPOSEDLY OF TYPE 05 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP05A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP05A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP05A) - INVALID BOUNDARY DESIGNATOR OF 'ii' GIVEN ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - NEGATIVE 'ii'-ENERGY GROUP NUMBER OF 'ii' GIVEN ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - THE LOWER ENERGY GROUP NUMBER IS LESS THAN THE HIGHER ENERGY GROUP NUMBER ON THE 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - NEGATIVE 'aaaaaa' COORDINATE OF 'ii' GIVEN ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - LOWER COORDINATE OF 'x.x' NOT LESS THAN HIGHER COORDINATE OF 'x.x' ON 'ii'-TH 05 CARD IN A.NIP.

(TYP05A) - IN A ONE DIMENSIONAL PROBLEM, Y BOUNDARIES ARE INCORRECTLY USED - CARD OF TYPE 05 - NUMBER 'ii' IN A.NIP.

TYPE06

Fatal:

(TYPE06) - ARRAY OVERFLOW. N2 = 'ii'.

Serious:

(TYPE06) - NO TYPE 06 CARDS SUPPLIED IN A.NIP.

TYP06A

Fatal:

(TYP06A) - CARD SUPPOSEDLY ON TYPE 06 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP06A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP06A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP06A) - BLANK REGION LABEL ON 'ii'-TH 06 CARD IN A.NIP.

(TYP06A) - INVALID X-COORDINATE REGION BOUNDARIES ON 'ii'-TH 06 CARD in A.NIP.

(TYP06A) - NEGATIVE NUMBER OF INTERVALS GIVEN ON 'ii'-TH 06 CARD IN A.NIP.

(TYP06A) - INVALID Y-COORDINATE REGION BOUNDARIES ON 'ii'-TH 06 CARD IN A.NIP.

TYPE07

Fatal:

(TYPE07) - ARRAY OVERFLOW. N2 = 'ii'

TYP07A

Fatal:

(TYP07A) - CARD SUPPOSEDLY ON TYPE 07 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP07A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP07A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP07A) - BLANK AREA LABEL ON 'ii'-TH 07 CARD IN A.NIP.

(TYP07A) - REGION LABEL OF 'aaaaaa' GIVEN ON 'ii'-TH TYPE 07 CARD IN A.NIP DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD.

TYPE08

Fatal:

(TYPE08) - ARRAY OVERFLOW. N2 = 'ii'

TYP08A

Fatal:

(TYP08A) - CARD SUPPOSEDLY OF TYPE 08 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP08A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP08A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP08A) - BLANK ZONE LABEL ON 'ii'-TH 08 CARD IN A.NIP.

(TYP08A) - INVALID X-COORDINATE ZONE BOUNDARY ON 'ii'-TH 08 CARD IN A.NIP.

(TYP08A) - INVALID Y-COORDINATE ZONE BOUNDARY ON 'ii'-TH 08 CARD IN A.NIP.

TYPE09

Fatal:

(TYPE09) - ARRAY OVERFLOW. N2 = 'ii'

TYP09A

Fatal:

(TYP09A) - CARD SUPPOSEDLY OF TYPE 09 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP09A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP09A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP09A) - INVALID COORDINATE OF 'a' GIVEN ON 'ii'-TH 09 CARD IN A.NIP.

(TYP09A) - DISCREPANCY IN NUMBER OF $\left\{\begin{matrix}X\\Y\end{matrix}\right\}$ -DIRECTION 09 CARDS IN A.NIP. N(9) = 'ii' BUT $\left\{\begin{matrix}NIX\\NIY\end{matrix}\right\}$ = 'ii'

(TYP09A) - THE ZERO OR NEGATIVE VALUE 'x.x' WAS GIVEN FOR THE FIRST LOWER COORDINATE FOR THE $\left\{\begin{matrix}X\\Y\end{matrix}\right\}$ -DIRECTION ON A 09 CARD IN A.NIP.

(TYP09A) - THE ZERO OR NEGATIVE VALUE 'ii' WAS GIVEN AS THE FIRST NUMBER OF INTERVALS ON A $\left\{\begin{matrix}X\\Y\end{matrix}\right\}$ -DIRECTION 09 CARD IN A.NIP.

(TYP09A) - THE NEGATIVE VALUE 'ii' WAS GIVEN AS A NUMBER OF INTERVALS FOR THE $\left\{\begin{matrix}X\\Y\end{matrix}\right\}$ -DIRECTION ON A 09 CARD IN A.NIP.

(TYP09A) - THE UPPER COORDINATES FOR THE $\left\{\begin{matrix}X\\Y\end{matrix}\right\}$ -DIRECTION ON THE 09 CARDS IN A.NIP ARE NOT MONOTONICALLY INCREASING. THE INCORRECT VALUE IS 'x.x'

(TYP09A) - TWO SUCCESSIVE UPPER COORDINATES FOR THE $\left\{\begin{matrix}X\\Y\end{matrix}\right\}$ -DIRECTION ON THE 09 CARDS IN A.NIP ARE EQUAL. THE INCORRECT VALUE IS 'x.x'

TYPE10

Fatal:

(TYPE10) - ARRAY OVERFLOW, N2 = 'ii'

TYP10A

Fatal:

(TYP10A) - CARD SUPPOSEDLY OF TYPE 10 FOUND TO BE TYPE 'ii' IN A.NIP.

(TYP10A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP10A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP10A) - BLANK LABEL GIVEN ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - INVALID COORDINATE DIRECTION OF 'a' GIVEN ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - INVALID INTERNAL CONDITION OF 'ii' GIVEN ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - NEGATIVE VALUE OF 'x.x' GIVEN IN COLUMNS 25-36 ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - NEGATIVE VALUE OF 'x.x' GIVEN IN COLUMNS 37-48 ON 'ii'-TH CARD IN A.NIP.

(TYP10A) - LOWER INTERFACE COORDINATE NOT LESS THAN UPPER INTERFACE COORDINATE ON 'ii'-TH 10 CARD IN A.NIP.

(TYP10A) - THE SAME INTERNAL INTERFACE LABEL, 'aaaaaa' WAS USED ON MORE THAN ONE TYPE 10 CARD IN A.NIP.

(TYP10A) - X COORDINATE DIRECTION INPUT INDICATES A TWO-DIMENSIONAL PROBLEM. GEOMETRY HAS BEEN STATED AS ONE-DIMENSIONAL. CHECK CARD NUMBER 'ii' OF TYPE 10 IN A.NIP.

Serious:

(TYPE11) - TYPE 10 CARDS SUPPLIED IN A.NIP BUT NO TYPE 11 CARDS SUPPLIED.

TYP11A

Fatal:

(TYP11A) - CARD SUPPOSEDLY OF TYPE 11 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP11A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP11A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP11A) - INTERNAL INTERFACE LABEL OF 'aaaaaa' GIVEN ON 'ii'-TH TYPE 11 CARD IN A.NIP. DOES NOT MATCH LABEL ON ANY TYPE 10 CARD.

(TYP11A) - NEGATIVE 'aaaaaa'-ENERGY GROUP NUMBER OF 'ii' GIVEN ON 'ii'-TH 11 CARD IN A.NIP.

(TYP11A) - THE LOWER ENERGY GROUP NUMBER IS LESS THAN THE HIGHER ENERGY GROUP NUMBER OF THE 'ii'-TH 11 CARD IN A.NIP.

(TYP11A) - GEOMETRY INDICATED IS ONE DIMENSIONAL. BOUNDARY CONDITION CONSTANTS MAY NOT APPLY TO ALL GROUPS. CHECK CARD NUMBER 'ii' OF TYPE 11 IN A.NIP.

TYPE12

Fatal:

(TYP12) - ARRAY OVERFLOW N2 = 'ii'

TYP12A

Fatal:

(TYP12A) - CARD SUPPOSEDLY OF TYPE 12 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP12A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP12A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP12A) - REGION LABEL OF 'aaaaaa' ON 'ii'-TH TYPE 12 CARD IN A.NIP. DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD.

(TYP12A) - NEGATIVE VALUE OF 'x.x' GIVEN FOR THE {FIRST / SECOND} TRANSVERSE EXTRAPOLATION DISTANCE ON THE 'ii'-TH TYPE 12 CARD IN A.NIP. EXCEEDS THE MAGNITUDE OF {FIRST / SECOND} ACTUAL TRANSVERSE HALF-DISTANCE ON CARD.

(TYP12A) NEGATIVE VALUE OF 'x.x' GIVEN FOR THE SECOND ACTUAL TRANSVERSE HALF-DISTANCE ON THE 'ii'-TH TYPE 12 CARD IN A.NIP.

(TYP12A) - THE SECOND ACTUAL TRANSVERSE HALF-DISTANCE GIVEN ON THE TYPE 12 CARDS IN A.NIP ARE NEITHER ALL ZERO (BLANK) NOR ALL POSITIVE, NON-ZERO QUANTITIES.

(TYP12A) - REGION LABEL IN A TYPE 12 CARD IS BLANK. IN THIS CASE ONLY ONE TYPE 12 CARD IS ALLOWED. THERE ARE 'ii' TYPE 12 CARDS IN A.NIP.

TYPE13

Fatal:

(TYPE13) - ARRAY OVERFLOW. N2 = 'ii'

TYP13A

Fatal:

(TYP13A) - CARD SUPPOSEDLY OF TYPE 'ii' FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP13A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP13A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP13A) - BLANK MATERIAL LABEL ON 'ii'-TH 13 CARD IN A.NIP.

(TYP13A) - BLANK COMPOSITION LABEL ON 'ii'-TH 14 CARD IN A.NIP.

(TYP13A) - FIRST ISOTOPE LABEL ON THE 'ii'-TH TYPE 13 CARD IN A.NIP IS BLANK.

(TYP13A) - FIRST MATERIAL LABEL ON THE 'ii'-TH TYPE 14 CARD IN A.NIP IS BLANK.

(TYP13A) - THE MATERIAL LABEL 'aaaaaa' GIVEN ON A TYPE 14 CARD IN A.NIP DOES NOT MATCH ANY MATERIAL LABEL FROM A TYPE 13 CARD.

Informative:

(TYP13A) - NEGATIVE ATOM DENSITY OF 'x.x' GIVEN ON 'ii'-TH TYPE 13 IN A.NIP.

(TYP13A) - NEGATIVE VOLUME FRACTION OF 'x.x' GIVEN ON 'ii'-TH TYPE 14 IN A.NIP.

TYPE14

Fatal:

(TYPE14) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE14) - NO TYPE 14 CARDS SUPPLIED IN A.NIP.

TYPE15

Fatal:

(TYPE15) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE15) - NO TYPE 15 CARDS SUPPLIED IN A.NIP.

TYP15A

Fatal:

(TYP15A) - CARD SUPPOSEDLY OF TYPE 15 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP15A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP15A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP15A) - THE COMPOSITION LABEL 'aaaaaa' GIVEN ON A TYPE 15 CARD IN A.NIP DOES NOT MATCH ANY COMPOSITION LABEL GIVEN ON A TYPE 14 CARD.

(TYP15A) - THE 'ii'-TH REGION LABEL 'aaaaaa' GIVEN ON THE 'ii'-TH TYPE 15 CARD IN A.NIP DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD.

(TYP15A) - THE REGION LABEL 'aaaaaa' APPEARS MORE THAN ONCE IN THE SET OF TYPE 15 CARDS SUPPLIED IN A.NIP.

(TYP15A) - THE COMPOSITION LABEL 'aaaaaa' GIVEN ON A TYPE 15 CARD IN A.NIP DOES NOT MATCH ANY COMPOSITION LABEL GIVEN IN THE OLD XS.C.MIN DATA SET.

TYPE19

Fatal:

(TYPE19) - ARRAY OVERFLOW. N2 = 'ii'

TYP19A

Fatal:

(TYP19A) - CARD SUPPOSEDLY OF TYPE 19 FOUND TO BE TYPE 'ii' IN A.NIP.

(TYP19A) - READ ERROR. UNEXPECTED END IF FILE ENCOUNTERED. N2 = 'ii'

(TYP19A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP19A) - THE NEGATIVE VALUE 'ii' WAS GIVEN FOR AN ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 19 CARD IN A.NIP.

(TYP19A) - THE HIGHER ENERGY GROUP NUMBER IS GREATER THAN THE LOWER ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 19 CARD IN A.NIP.

(TYP19A) - THE NEGATIVE VALUE 'x.x' WAS GIVEN FOR A LOWER COORDINATE ON THE 'ii'-TH TYPE 19 CARD IN A.NIP.

(TYP19A) - ERROR ON CARD NO. 'ii' OF TYPE 19 IN A.NIP. VALUE GIVEN WAS 'ii' ZERO OR 1 IS REQUIRED.

(TYP19A) - ERROR ON CARD NUMBER 'ii' OF TYPE 19 IN A.NIP. GROUP DEPENDENT INPUT MAY NOT HAVE A GROUP WEIGHTING FLAG OF 1.

TYP19B

Serious:

(TYP19B) - THE CONTENTS ON COLS. 7-12 ON THE 'ii'-TH TYPE 'ii' CARD IN A.NIP IS NOT 0, 1, or 2.

(TYP19B) - THE 'ii'-TH TYPE 'ii' CARD IN A.NIP HAS THE VALUE 'ii' IN COLS. 7-12. THIS MEANS THAT COLS. 13-18 OF THE CARD MUST CONTAIN A 'aaaaaa' LABEL. HOWEVER, NO TYPE 'aaaaaa' CARDS WERE INCLUDED IN A.NIP.

(TYP19B) - THE LABEL ON THE 'ii'-TH TYPE 'ii' CARD IN A.NIP DOES NOT MATCH THE 'aaaaaa' LABEL GIVEN ON ANY TYPE 'aaaaaa' CARD.

TYPE21

Fatal:

(TYPE21) - CARD SUPPOSEDLY OF TYPE 21 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYPE21) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYPE21) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYPE21) - CONTENTS OF COLS. 13-18 ON THE TYPE 21 CARD IN A.NIP ARE NOT 0 OR 1.

(TYPE21) - MAXIMUM NUMBER OF SEARCH PASSES IN COLS. 19-24 ON THE TYPE 21 CARD IN A.NIP IS NEGATIVE.

(TYPE21) - THE VALUE OF THE PRESCRIBED K-EFFECTIVE IN COLS. 25-36 OF THE TYPE 21 CARD IN A.NIP IS NEGATIVE.

Informative:

(TYPE21) - THE CONVERGENCE CRITERION IN COLS. 37-48 OF THE TYPE 21 CARD IN A.NIP IS NOT POSITIVE. DEFAULT OF 1.0E-04 HAS BEEN SUBSTITUTED.

TYPE22

Fatal:

(TYPE22) - CARD SUPPOSEDLY OF TYPE 22 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYPE22) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYPE22) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYPE22) - A TYPE 21 CARD WAS SUPPLIED IN A.NIP BUT NO TYPE 22 CARD WAS SUPPLIED.

TYPE23

Fatal:

(TYPE23) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE23) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25, 26 WERE SUPPLIED IN A.NIP.

TYP23A

Fatal:

(TYP23A) - CARD SUPPOSEDLY OF TYPE 23 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP23A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP23A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP23A) - THE QUANTITY IN COLS. $\left\{\begin{matrix}7\text{-}13\\13\text{-}72\end{matrix}\right\}$ ON THE 'ii'-TH TYPE 23 CARD IN A.NIP 'aaaaaa' DOES NOT DUPLICATE ANY COMPOSITION LABEL GIVEN ON A TYPE 14 CARD OR ON THE OLD XS.C.MIN DATA SET.

TYPE24

Fatal:

(TYPE24) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE24) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25, OR 26 WERE SUPPLIED IN A.NIP.

TYP24A

Fatal:

(TYP24A) - CARD SUPPOSEDLY OF TYPE 24 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP24A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP24A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP24A) - THE LOWER COORDINATE GIVEN ON THE 'ii'-TH TYPE 24 CARD IN A.NIP IS NEGATIVE.

(TYP24A) - THE LOWER COORDINATE GIVEN ON THE 'ii'-TH TYPE 24 CARD IN A.NIP IS GREATER THAN THE UPPER COORDINATE GIVEN ON THE CARD.

(TYP24A) - THE COORDINATE DESIGNATOR GIVEN IN COL. 12 ON THE 'ii'-TH TYPE 24 CARD IN A.NIP IS NOT X OR Y.

TYPE25

Fatal:

(TYPE25) - ARRAY OVERFLOW. N2 = 'ii'

Serious:

(TYPE25) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25, 26 WERE SUPPLIED IN A.NIP.

TYP25A

Fatal:

(TYP25A) - CARD SUPPOSEDLY OF TYPE 25 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP25A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP25A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP25A) - THE LABEL GIVEN ON THE 'ii'-TH TYPE 25 CARD IN A.NIP DOES NOT MATCH ANY REGION LABEL GIVEN ON A TYPE 06 CARD.

(TYP25A) - THE REGION LABEL 'aaaaaa' APPEARS ON MORE THAN ONE TYPE 25 CARD IN A.NIP.

TYPE26

Fatal:

(TYPE26) - CARD SUPPOSEDLY OF TYPE 25 FOUND TO BE OF TYPE 'ii' in A.NIP.

(TYPE26) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYPE26) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYPE26) - CARDS OF MORE THAN ONE OF THE TYPES 23, 24, 25, 26 WERE SUPPLIED IN A.NIP.

TYPE27

Fatal:

(TYPE27) - ARRAY OVERFLOW. N2 = 'ii'

TYP27A

Fatal:

(TYP27A) - CARD SUPPOSEDLY OF TYPE 27 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP27A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP27A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP27A) - THE NEGATIVE VALUE 'ii' WAS GIVEN FOR AN ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 27 CARD IN A.NIP.

(TYP27A) - THE HIGHER ENERGY GROUP NUMBER IS GREATER THAN THE LOWER ENERGY GROUP NUMBER ON THE 'ii'-TH TYPE 27 CARD IN A.NIP.

(TYP27A) - ALL THREE SOURCE VALUES IN COLS. 37-72 OF THE 'ii'-TH TYPE 27 CARD IN A.NIP ARE ZERO OR BLANK.

(TYP27A) - THERE IS NO LABEL GIVEN IN COLS. 7-12 ON CARD NO. 'ii' OF TYPE 27 in A.NIP.

TYPE28

Fatal:

(TYPE28) - ARRAY OVERFLOW. N2 = 'ii'

TYP28A

Fatal:

(TYP28A) - CARD SUPPOSEDLY OF TYPE 28 FOUND TO BE OF TYPE 'ii' IN A.NIP.

(TYP28A) - READ ERROR. UNEXPECTED END OF FILE ENCOUNTERED. N2 = 'ii'

(TYP28A) - READ ERROR. ERROR IN DATA TRANSFER. N2 = 'ii'

Serious:

(TYP28A) - LABEL ON THE 'ii'-TH TYPE 28 CARD IN A.NIP DOES NOT MATCH ANY LABEL ON A TYPE 27 CARD.

(TYP28A) - THE LOWER COORDINATE GIVEN ON THE 'ii'-TH TYPE 28 CARD IN A.NIP HAS A NEGATIVE VALUE

(TYP28A) - THE UPPER COORDINATE GIVEN ON THE 'ii'-TH TYPE 28 CARD IN A.NIP HAS A NEGATIVE VALUE.

(TYP28A) - COORDINATE DIRECTION GIVEN ON THE 'ii'-TH TYPE 28 CARD IN A.NIP IS NOT X OR Y.

(TYP28A) - THERE IS NO LABEL GIVEN IN COLS. 7-12 ON CARD NO. 'ii' OF TYPE 28 IN A.NIP.

(TYP28A) - LOWER COORDINATE IS NOT LESS THAN UPPER COORDINATE GIVEN ON THE 'ii'-TH CARD IN A.NIP.

(TYP28A) - THE COORDINATE IN COLUMNS 60 to 72 ON THE 'ii'-TH CARD IN A.NIP IS NOT POSITIVE.

ZONEDR

Fatal:

(ZONEDR) - ARRAY OVERFLOW, N2 = 'ii'

9. GNIP (NUI002) Timing Information

For a one-dimensional external source, 6-group, 260 mesh interval problem with 6 regions and 3 compositions, running time was .15 minutes. For 22-group problems, 40 or 80 intervals, 4 regions, 4 compositions, average time was .09 minutes.

Two-dimensional synthesis runs for a 6-group, 2-region, 1-composition problem took .18 minutes for runs having 10 × 110 intervals, .39 minutes for problems having 10 × 1100 intervals, and .15 minutes for problems having 10 × 220 intervals.

Problems consisting mainly of 2-region, 2-composition problems, with 3, 16, and 26 groups and from 30 to 50 mesh intervals showed an average time of .13 minutes.

Generally, time appears to be directly proportional to number of mesh points and number of groups.

10. GNIP (NUI002) Sample Output

The material below is representative of the GNIP module printed output. The output is described in Section 2d.

```
*****************************************************************************************************************

 ENTERING GNIP MODULE

*****************************************************************************************************************
SNIFF***********XS.C.MIN HAS NOT BEEN WRITTEN YET AND YOU HAVE REQUESTED TO READ IT.
TYPE 03 CARD EDIT.
GEOMETRY TYPE =     51
PROBLEM TYPE =      0
CALCULATION TYPE =      0
INITIAL ALPHA =  0.0


TYPE 04 CARD EDIT.
THE FOUR BOUNDARY CONDITIONS ARE     3     2     3     2

 EDIT OF DATA SET GEOM

  NDIM  NGEOM NPTI  NPTJ  NINTI NINTJ NREG  NCMP  NAREA LAREA NZONE LZONE NFIN  NIL
    2     8    11    15    10    14    12    11     1     0     1     0     0    0

 MESH POINTS
       0.0                  0.127000D 01         0.254000D 01         0.241300D 02         0.457200D 02         0.459785D 02
       0.462369D 02         0.535985D 02         0.609600D 02         0.685800D 02         0.762000D 02
       0.0                  0.144000D 01         0.288000D 01         0.109100D 02         0.189400D 02         0.224100D 02
       0.258800D 02         0.309600D 02         0.360400D 02         0.435400D 02         0.510400D 02         0.547900D 02
       0.585400D 02         0.622900D 02         0.660400D 02

 COMPOSITION REGION ASSIGNMENT
   11    11    11     6     2     6     3     6     6     6     5     5

 REGION LABELS
R1      R2      R3      R4      R5      R6      R7      R8      R9      R10
R11     R12

 COMPOSITION LABELS
C1      C2      C3      C4      C5      C6      C7      C8      C9      C10
C11

 AREA LABELS


 ZONE LABELS


REGION VOLUMES

        REGION   R1       0.66186D 02


        REGION   R2       0.11252D 04


        REGION   R3       0.50333D 05


        REGION   R4       0.58253D 03


        REGION   R5       0.44677D 05


        REGION   R6       0.50511D 03
```

```
REGION   R7       0.90361D 05

REGION   R8       0.10216D 04

REGION   R9       0.25015D 06

REGION   R10      0.60078D 05

REGION   R11      0.33633D 06

REGION   R12      0.20881D 06
```

MESH REGION COMPOSITION ASSIGNMENT

66.0400									
R R11	R11	R11	R11	R11	R11	R11	R11	R12	R12
C C5	C5	C5	C5	C5	C5	C5	C5	C5	C5
62.2900									
R R11	R11	R11	R11	R11	R11	R11	R11	R12	R12
C C5	C5	C5	C5	C5	C5	C5	C5	C5	C5
58.5400									
R R11	R11	R11	R11	R11	R11	R11	R11	R12	R12
C C5	C5	C5	C5	C5	C5	C5	C5	C5	C5
54.7900									
R R11	R11	R11	R11	R11	R11	R11	R11	R12	R12
C C5	C5	C5	C5	C5	C5	C5	C5	C5	C5
51.0400									
R R9	R9	R9	R9	R9	R9	R9	R9	R12	R12
C C6	C6	C6	C6	C6	C6	C6	C6	C5	C5
43.5400									
R R9	R9	R9	R9	R9	R9	R9	R9	R12	R12
C C6	C6	C6	C6	C6	C6	C6	C6	C5	C5
36.0400									
R R7	R7	R7	R7	R8	R8	R10	R10	R12	R12
C C3	C3	C3	C3	C6	C6	C6	C6	C5	C5
30.9600									
R R7	R7	R7	R7	R8	R8	R10	R10	R12	R12
C C3	C3	C3	C3	C6	C6	C6	C6	C5	C5
25.8800									
R R5	R5	R5	R5	R6	R6	R10	R10	R12	R12
C C2	C2	C2	C2	C6	C6	C6	C6	C5	C5
22.4100									
R R5	R5	R5	R5	R6	R6	R10	R10	R12	R12
C C2	C2	C2	C2	C6	C6	C6	C6	C5	C5
18.9400									
R R3	R3	R3	R3	R4	R4	R10	R10	R12	R12
C C11	C11	C11	C11	C6	C6	C6	C6	C5	C5
10.9100									
R R3	R3	R3	R3	R4	R4	R10	R10	R12	R12
C C11	C11	C11	C11	C6	C6	C6	C6	C5	C5
2.8800									
R R1	R1	R2	R2	R4	R4	R10	R10	R12	R12
C C11	C11	C11	C11	C6	C6	C6	C6	C5	C5
1.4400									
R R1	R1	R2	R2	R4	R4	R10	R10	R12	R12
C C11	C11	C11	C11	C6	C6	C6	C6	C5	C5
0.0									

1	2	3	4	5	6	7	8	9	10	11
0.0	1.2700	2.5400	24.1300	45.7200	45.9785	46.2369	53.5985	60.9600	68.5800	76.2000

EDIT OF DATASET BC

NBCL	NBCR	NVB	NBCB	NBCT	NHB
1	1	2	1	1	2

BOUNDARY CONDITIONS-TYPE 2 RECORDS

BOUNDARY 1

MFACE	M1	M2	NBIT	JI	JF	C1	C2	C3	C4
1	1	15	1	1	22	0.10000D 01	0.0	0.0	0.0

BOUNDARY 2

MFACE	M1	M2	NBIT	JI	JF	C1	C2	C3	C4
-11	1	15	1	1	22	0.0	0.10000D 01	0.0	0.0

BOUNDARY 3

MFACE	M1	M2	NBIT	JI	JF	C1	C2	C3	C4
1	1	11	1	1	22	0.10000D 01	0.0	0.0	0.0

BOUNDARY 4

MFACE	M1	M2	NBIT	JI	JF	C1	C2	C3	C4
-15	1	11	1	1	22	0.0	0.10000D 01	0.0	0.0

DATA SET SP.CICN

MKRTHY	NGEOM	NDIM	NINTI	NINTJ	NGROUP	IPROB
0	8	2	10	14	22	2

CONVCR	X1	X2	ALPHA	FIXEDK
0.0	0.0	0.0	0.0	0.10000D 01

****
TIME IN GNIP WAS 0.0642 MINUTES
GNIP MODULE HAS BEEN UTILIZED
****
****
****

PART TWO.

GEOMETRY DATA SET (GEOM) COPIER, CPGEOM (AJC006)

Programmed by - D. E. Neal

Documented by - D. E. Neal

1. CPGEOM (AJC006) Introduction

a. Description of Capabilities

This module copies the geometry data set GEOM from one data set reference number to another.

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

	Modules Utilized by Each Path	Paths Utilizing This Module	
		STP006 1-D Diffusion Theory Perturbation	STP007 1-D Diffusion Theory Perturbation
	DELDATA (CSI007) Delayed Neutron Data Preparation	X	X
	INHOMG (NUI001) Multigroup Cross Section Homog. Specifications	X	X
	HOMOG (NUC001) Multigroup Cross Section Homog.	X	X
	GNIP (NUI002) Neutronics Input Processor	X	X
DARC1D	DIF1D (NUC002) 1-D Diffusion Theory k_{eff} and Source Calculations	X	
DARC1D	D1DSCH (NUC004) 1-D Diffusion Theory Crit. Search	X	
DARC2D	DIF2D (NUC005) 2-D Diffusion Theory k_{eff} and Source Calculations		X
DARC2D	D2DSCH (NUC006) 2-D Diffusion Theory Crit. Search		X

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

Modules Utilized by Each	Paths Utilizing This Module	
	STP006 1-D Diffusion Theory Perturbation	STP007 1-D Diffusion Theory Perturbation
MODXSISO (NUI006) Run-Time Microscopic Cross-Section Modification	X	X
PARC1D (AJC003) 1-D Diffusion Theory Perturbation	X	
PARC2D (AJC004) 2-D Diffusion Theory Perturbation		X
OUTMAN1D (NUE001) 1-D Neutronics Output Manipulation	X	
INVENT1D (AJC001) 1-D Neutron Inventory	X	
INVENT2D (AJC002) 2-D Neutron Inventory		X
CPGEOM (AJC006) Geometry Data Set (GEOM) Copier	X	X

c. Data Sets Utilized by This Module

Upper-interface Data Sets	Records Read
GEOM, Geometry Data	All present

Lower-interface Data Sets	Records Written
GEOM, Geometry Data	All present in GEOM

2. CPGEOM (AJC006) Description of Problem and Method of Solution

Before computing a new or modified geometry data set using module GNIP (NUI002), it may be necessary to save the geometry data produced by another module or by an earlier call to GNIP. CPGEOM copies the geometry data set GEOM from one data set reference number to another, so that the previous GEOM can be saved while a new or modified GEOM is being created.

The only normal printed output from this module consists of statements that the module has been entered, that GEOM has been copied, and that the module has been left.

3. CPGEOM (AJC006) 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 CPGEOM module segments required by this module. This is followed by a list of the subprograms in the CPGEOM module segment.

Section b. illustrates the flow of logic through the subprograms of the module, and section c. describes this flow in terms of the module's functions.

a. Overlay Structure Diagram

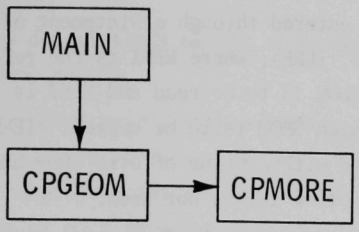
```
MAIN
CPGEOM
SYSTEM
LINK
LOAD
POINTR
SQUEZE
IBCOM
```

Subprograms contained in the CPGEOM module segment

CPGEOM

MAIN
CPGEOM
CPMORE

b. Diagrammatic Subprogram Logic Flow



c. Descriptive Subprogram Logic Flow

Module CPGEOM is entered through a statement of the form CALL LINK ('AJC006', NDS1, NDS2, IERR), where NDS1 is the reference number of the data set from which GEOM is to be read and NDS2 is the reference number of the data set to which GEOM is to be copied. IERR is an error flag, which will be returned with a value of 0 if GEOM has been successfully copied and a value of 1 if it has not been.

CPGEOM is entered at MAIN, which calls subroutine CPGEOM. CPGEOM initializes POINTR (ARC System module) and copies the first record, Specifications, of GEOM. For each of the next four records, CPGEOM sets up in the POINTR container array the storage needed, calls CPMORE to copy the record, and wipes out the storage allocated for the array. This process is then repeated for whichever of record types 6 (Area Definitions), 7 (Zone Definitions), and 8 (Finite Geometry Data) are present, and then for record types 9 (Labels) and 10 (Region Volumes). CPGEOM then returns to MAIN, which returns to the program which linked to the module.

If any error is encountered in reading GEOM, if an unexpected end-of-file is encountered, or if dynamic storage space is exceeded, an error message is printed and execution is terminated.

4. CPGEOM (AJC006) Subprogram Descriptions

This section contains descriptions of the three subprograms which are part of the CPGEOM (AJC006) module.

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 dimension 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.

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 dimension, if any, follows the variable name in parentheses.

Subsection 3 contains a listing of important variables which are local to the subprogram, in a format like that of 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 this subprogram or any of its entry points; subsection 6 lists the subprograms which call this subprogram or any of its entry points.

Subsection 7 describes error messages which can be generated by this 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.

Main Program MAIN

1. Arguments - None

2. Common blocks - None

3. Local variables - None

4. Functions and tasks performed by this subroutine

This is the main program for module CPGEOM. It serves to pass the arguments of the module from the calling program to subroutine CPGEOM, and to return the error flag from subroutine CPGEOM to the calling program.

5. Subprograms called by this program

CPGEOM

6. Subprograms calling this program

Paths or programs which link to this module

7. Error messages generated by this program - None

Subroutine CPGEOM (NDS1,NDS2,IERR)

1. Arguments

Name	Type	Dimension	Definition
NDS1	I*4	--	Reference number of data set from which GEOM is to be read
NDS2	I*4	--	Reference number of data set to which GEOM is to be copied
IERR	I*4	--	Error flag - 0 - execution successful 1 - execution terminated

2. Common blocks

/BLKSTR/ BLK(2000) Large container array used by POINTR

3. Local variables

Name	Type	Definition
IREC	I*4	Number of GEOM record type being read
NDIM	I*4	Number of dimensions from record type 1
NGEOM	I*4	Number of geometry type from record type 1
NPTI	I*4	Number of x-direction mesh lines from record type 1
NPTJ	I*4	Number of y-direction mesh lines from record type 1
NINTI	I*4	Number of x-direction mesh intervals from record type 1
NINTJ	I*4	Number of y-direction mesh intervals from record type 1
NREG	I*4	Number of regions from record type 1
NCOMP	I*4	Number of compositions from record type 1
NAREA	I*4	Number of areas from record type 1
LAREA	I*4	Length of area record (type 6) from record type 1
NZONE	I*4	Number of zones from record type 1
LZONE	I*4	Length of zone record (type 7) from record type 1
NFIN	I*4	Number of finite direction type from record type 1
NIL	I*4	Number of integration limits from record type 1
NPT	I*4	Twice the total number of mesh lines, 2*(NPTI+NPTJ)
NINT	I*4	Total number of mesh intervals, NINTI*NINTJ
NLGTH	I*4	Length of record type 8, 9 or 10 in fullwords

4. Functions and tasks performed by this subroutine

CPGEOM initializes the large container array in common block /BLKSTR/ by calling ARC System module POINTR. Then, for each record type present in GEOM, it sets up an array of fullwords (four bytes) long enough to hold the record contents, and calls CPMORE to read the record into the array from NDS1 and write the contents of the array on NDS2. The array space is then wiped out of the container array.

5. Subprograms called by this subroutine

CPMORE

6. Subprograms calling this subroutine

Main program

7. Error messages generated by this subroutine

Message: *****ERROR IN READING RECORD TYPE ii OF DATA SET GEOM

Significance: An error was encountered in attempting to read the record type whose number is given.

Message: *****UNEXPECTED END-OF-FILE ENCOUNTERED IN ATTEMPTING TO READ RECORD TYPE ii OF DATA SET GEOM

Significance: An end-of-file was encountered in attempting to read the record type whose number is given.

Message: *****ERROR ENCOUNTERED IN ALLOCATING POINTR ARRAY SPACE FOR READING RECORD TYPE ii OF DATA SET GEOM

Significance: There was not sufficient space in the dynamic storage array BLK to hold the array which was to contain the contents of the record type whose number is given.

These messages are each followed by the message

GEOM COULD NOT BE COPIED FROM DSRN ii TO DSRN ii

if the error occurs in reading the first record; otherwise, they are followed by the message

ONLY THE FIRST ii RECORDS OF GEOM COULD BE COPIED FROM DSRN ii TO DSRN ii

Action taken: In all cases, those records that could be copied up to the point of error are copied. Execution is terminated.

Subroutine CPMORE (ARRAY,LGTH,NDS1,NDS2,*,*,*)

1. Arguments

Name	Type	Dimension	Definition
ARRAY	R*4	LGTH	Array to be read from NDS1 and written on NDS2
LGTH	I*4	--	Length of array in fullwords
NDS1	I*4	--	Reference number of data set from which GEOM is to be read
NDS2	I*4	--	Reference number of data set to which GEOM is to be copied
*	--	--	Error return for error in reading
*	--	--	Error return for unexpected end-of-file
*	--	--	Error return for POINTR error

2. Common blocks

/BLKSTR/ BLK(1)

3. Local variables - None

4. Functions and tasks performed by this subroutine

CPMORE reads one record, consisting of the fullword array ARRAY of length LGTH, from the data set located on NDS1, and writes the same record on the data set located on NDS2. If any error in reading or an unexpected end-of-file is encountered, or if it is discovered that space for ARRAY was not properly allocated in BLK, the appropriate error return is used and the record is not processed.

5. Subprograms called by this subroutine - None

6. Subprograms calling this subroutine

CPGEOM

7. Error messages generated by this subroutine - None

5. CPGEOM (AJC006) Common Block Descriptions

This module contains only one common block, the /BLKSTR/ common block, which contains the container array BLK used by the dynamic storage allocation module POINTR (see Volume I of this series).

The block itself is described below. The variable it contains is listed by name and type, and the subprogram where the variable is initialized by calling POINTR is given. The following section 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.

/BLKSTR/

Name	Type	Subprograms where values are set or modified	Definition
BLK(2000)	R*8	CPGEOM	Large container array for dynamic storage allocation used by ARC System module POINTR

6. CPGEOM (AJC006) 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
BSQ	R*4	6*NREG*NIL	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 8 (Finite Geometry Data) of GEOM, if present, consisting in actuality of three R*8 quantities, each dimensioned NREG*NIL
LABELS	R*4	2*(NREG+ NCOMP+ NAREA+ NZONE)	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 9 (Labels), consisting in actuality of the R*8 labels for regions, compositions, zones and areas
MC	R*4	NINTI*NINTJ	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 4 (Composition-Interval Correspondence), consisting in actuality of the I*4 index for the composition assigned to each interval
MR	R*4	NINTI*NINTJ	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 3 (Region-Interval Correspondence), consisting in actuality of the I*4 index for the region assigned to each interval
NC	R*4	NREG	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 5 (Composition-Region Correspondence) consisting in actuality of the I*4 index for the composition assigned to each region

Array Name	Type	Dimension	Subprograms issuing significant POINTR calls	Subprograms where values are set or modified	Definition
NREGA	R*4	LAREA	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 6 (Area Definitions), if present, actually I*4 quantities
NSECTZ	R*4	LZONE	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 7 (Zone Definitions), if present, actually I*4 quantities
REGVOL	R*4	2*NREG	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 10 (Region Volumes), actually R*8 quantities
XYMESH	R*4	2*(NPTI +NPTJ)	CPGEOM(PUTPNT) CPGEOM(WIPOUT)	CPMORE	The contents of record type 2 (Boundaries), consisting in actuality of the R*8 mesh coordinates in each direction

7. CPGEOM (AJC006) Data Set Usage

Only one data set is used by this module. The following information is given for that data set:

1. the name of the data set
2. the record structure of the data set
3. the names of subprograms in which the data set is read and/or written, along with a list of the record types which are read within each routine and the name of the variable in which the reference number of the data set is stored.

Complete descriptions of the record structure and contents of this data set can be found in Volume I of this series.

Name or description of data set - GEOM

Record Structure - ARC System data set

Read in subprograms	Record types	Reference variable
CPGEOM	Specifications (type 1)	NDS1
CPMORE	Boundaries (type 2); Region-Interval Correspondence (type 3); Composition-Interval Correspondence (type 4); Composition-Region Correspondence (type 5); Area Definitions (type 6) if present; Zone Definitions (type 7) if present; Finite Geometry Data (type 8) if present; Labels (type 9); and Region Volumes (type 10), one at a time	NDS1

Written in subprograms	Record types	Reference variable
CPGEOM	Specifications (type 1)	NDS2
CPMORE	Same as above	NDS2

8. CPGEOM (AJC006) Error Messages

This module generates three possible error messages; all errors are detected in subroutine CPGEOM for record type 1 or in CPMORE for record types 2-9. The message is always printed in subroutine CPGEOM.

The three possible messages follow.

Message: *****ERROR IN READING RECORD TYPE ii OF DATA SET GEOM

Significance: An error was encountered in attempting to read the record type whose number is given.

Message: *****UNEXPECTED END-OF-FILE ENCOUNTERED IN ATTEMPTING TO READ RECORD TYPE ii OF DATA SET GEOM

Significance: An end-of-file was encountered in attempting to read the record type whose number is given.

Message: *****ERROR ENCOUNTERED IN ALLOCATING POINTR ARRAY SPACE FOR READING RECORD TYPE ii OF DATA SET GEOM

Significance: There was not sufficient space in the dynamic storage array BLK to hold the array which was to contain the contents of the record type whose number is given.

These messages are each followed by the message

GEOM COULD NOT BE COPIED FROM DSRN ii TO DSRN ii

if the error occurs in reading the first record; otherwise, they are followed by the message

ONLY THE FIRST ii RECORDS OF GEOM COULD BE COPIED FROM DSRN ii TO DSRN ii

Action taken: In all cases, those records that could be copied up to the point of error are copied. Execution is terminated.

9. Timing Information

This module requires less then .02 minute for a finite two-dimensional data set with no areas or zones defined, with 100 mesh points and five regions.

ACKNOWLEDGMENT

The authors wish to thank D. E. Neal and L. T. Bryant for their editorial assistance throughout the preparation of this document.

ARGONNE NATIONAL LAB WEST
3 4444 00011422 3

B