UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D. C. 20555

December 1984

Recipients of the SCALE Manual, NUREG/CR-0200

Revision No. 3

This documentation material describes recent additions and modifications made to the SCALE computational system. Combining this material with previously issued material will update the SCALE Manual to be consistent with the latest publicly released version of SCALE, called SCALE-3 (see the revised FOREWORD for code acquisition information).

If out-of-date material is replaced by the revised pages/sections, the Manual as published to date can be contained in five large-ring notebook binders. Sections F13-F14 and M12-M14 are entirely new sections of documentation that should be placed in the Manual according to the revised master Table of Contents. These new sections describe geometry and output plot programs for use with HEATING6, a printer plot program for the MARS combinatorial geometry, and a library of input data for analyzing benchmark critical experiments.

The pages behind the "ERRATA AND REVISIONS" tab have been placed in order of their correct sequential appearance in the Manual and should replace the corresponding pages or sections from previous releases. Besides the revised master Table of Contents, this material includes Foreword, List of Revisions (providing publication dates and revision dates for all parts of NUREG/CR-0200), corrected and revised pages necessary to update the various sections of the Manual to the SCALE-3 version, and a completely revised documentation for KENO V and JUNEBUG which have been updated to KENO V.a and JUNEBUG-II. The existing tabs provided with earlier releases can still be used with these revised sections. KENO V.a provides significantly improved geometric modeling capabilities over KENO V while JUNEBUG-II is an improved and more efficient version of the existing three-dimensional JUNEBUG plotting package. The revised pages reflect (1) changes to CSAS4 to allow interfacing with KENO V.a and add two new analytic sequences, (2) changes to HTAS1 to update the defaults to 10CFR71 and add plotting capabilities, (3) changes to ORIGEN-S to add an improved evaluation of the neutron source and allow data output for subsequent plotting, (4) changes to MORSE-SGC/S and MARS to improve tracking efficiency and correct known errors, and (5) addition of several materials to the Standard Composition Library.

If there are any questions concerning this revision or any other portion of the SCALE Manual, please contact me at (615) 574-5280 or (FTS) 624-5280.

C. V. Parks, ORNL
SCALE Project Leader

R. H. Odegaarden, NRC
Technical Monitor

NOTE: Revised Sections H1 and F11 both contain microfiche. The microfiche (3p. for Sec. H1 and 8p. for Sec. F11) will be provided under separate cover at a later date, to be placed at the end of each section.

8501070411 841231
PDR NUREG
CR-0200 R PDR

ornl

NUREG/CR-0200
VOLUMES 1, 2, 3
ORNL/NUREG/CSD-2/R3

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

SCALE:
A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

Additions and Revisions

Published December 1984

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
Under Interagency Agreement DOE 40-549-75

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

TABLE OF CONTENTS
FOR SCALE MANUAL

	Page*
ACKNOWLEDGMENTS	vii
ABSTRACT	ix
FOREWORD	xi
LIST OF REVISIONS	xv

VOLUME 1: CONTROL MODULES

Section C1	CSAS1: A ONE-DIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C1.1.1
Section C2	CSAS2: A MULTIDIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C2.1.1
Section C3	CSAS3: AN OPTIMUM CONCENTRATION CRITICALITY SAFETY ANALYSIS MODULE (to be issued at later date)	C3.1.1
Section C4	CSAS4: AN ENHANCED CRITICALITY SAFETY ANALYSIS MODULE WITH AN OPTIMUM PITCH SEARCH OPTION	C4.1.1
Section S1	SAS1: A ONE-DIMENSIONAL SHIELDING ANALYSIS MODULE (to be issued at later date)	S1.1.1
Section S2	SAS2: A COUPLED ONE-DIMENSIONAL DEPLETION AND SHIELDING ANALYSIS MODULE	S2.1.1
Section S3	SAS3: AN AUTOMATED MONTE CARLO SHIELDING ANALYSIS MODULE	S3.1.1
Section S4	SAS4: A MONTE CARLO SHIELDING ANALYSIS MODULE WITH AUTOMATIC BIASING. (to be issued at later date)	S4.1.1
Section H1	HTAS1: A TWO-DIMENSIONAL HEAT TRANSFER ANALYSIS OF FUEL CASKS	H1.1.1

*All section page numbers refer to the page where the section introduction begins.

VOLUME 2: FUNCTIONAL MODULES

		Page
Section F1	BONAMI-S: RESONANCE SELF-SHIELDING BY THE BONDARENKO METHOD	F1.1.1
Section F2	NITAWL-S: RESONANCE SELF-SHIELDING BY THE NORDHEIM METHOD	F2.1.1
Section F3	XSDRNPM-S: A ONE-DIMENSIONAL DISCRETE-ORDINATES PROGRAM FOR TRANSPORT ANALYSIS . . .	F3.1.1
Section F4	XSDOSE: A MODULE FOR CALCULATING FLUXES AND DOSE RATES AT POINTS OUTSIDE A SHIELD	F4.1.1
Section F5	KENO-IV/S: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM	F5.1.1
Section F6	COUPLE: SCALE SYSTEM MODULE TO PROCESS PROBLEM-DEPENDENT CROSS SECTIONS AND NEUTRON SPECTRAL DATA FOR ORIGEN-S ANALYSES	F6.1.1
Section F7	ORIGEN-S: SCALE SYSTEM MODULE TO CALCULATE FUEL DEPLETION, ACTINIDE TRANSMUTATION, FISSION PRODUCT BUILDUP AND DECAY, AND ASSOCIATED RADIATION SOURCE TERMS	F7.1.1
Section F8	ICE-S: MODULE TO MIX MULTIGROUP CROSS SECTIONS	F8.1.1
Section F9	MORSE-SGC/S: A NEUTRON AND PHOTON MONTE CARLO TRANSPORT ANALYSIS WITH SUPER-GROUPING AND ADVANCED GEOMETRY FEATURES	F9.1.1
Section F10	HEATING6: A MULTIDIMENSIONAL HEAT CONDUCTION ANALYSIS WITH THE FINITE-DIFFERENCE FORMULATION	F10.1.1
Section F11	KENO-V.a: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM WITH SUPER-GROUPING	F11.1.1
Section F12	JUNEBUG-II: A THREE-DIMENSIONAL GEOMETRY PLOTTING CODE	F12.1.1
Section F13	HEATPLOT-S: A TEMPERATURE DISTRIBUTION PLOTTING PROGRAM FOR HEATING5/HEATING6.	F13.1.1
Section F14	REGPLOT5: A PLOTTING PROGRAM TO VERIFY HEATING6 INPUT DATA	F14.1.1
Section F15	PLORIGEN: A PLOTTING PROGRAM FOR ORIGEN-S OUTPUT (to be issued at later date)	F15.1.1

VOLUME 3: MISCELLANEOUS

		Page
Section M1	THE SCALE SYSTEM DRIVER	M1.1.1
Section M2	THE SCALE SYSTEM SUBROUTINE LIBRARY	M2.1.1
Section M3	SCALE FREE FORM READING ROUTINES	M3.1.1
Section M4	NEUTRON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M4.1.1
Section M5	NEUTRON-PHOTON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M5.1.1
Section M6	ORIGEN-S DATA LIBRARIES	M6.1.1
Section M7	MATERIAL PROPERTIES LIBRARIES . . . . . (to be issued at later date)	M7.1.1
Section M8	STANDARD COMPOSITION LIBRARY	M8.1.1
Section M9	MARS: A MULTIPLE ARRAY SYSTEM USING COMBINATORIAL GEOMETRY	M9.1.1
Section M10	FIDO INPUT SYSTEM	M10.1.1
Section M11	SCALE INTERACTIVE INPUT PROCESSOR	M11.1.1
Section M12	CESAR: A CRITICALITY EXPERIMENT STORAGE AND RETRIEVAL PROGRAM.	M12.1.1
Section M13	PICTURE: A PRINTER PLOT PACKAGE FOR MAKING 2-D PICTURES OF MARS GEOMETRIES.	M13.1.1
Section M14	COMPOZ DATA GUIDE.	M14.1.1

FOREWORD FOR SCALE MANUAL

This manual describes a system of computer codes for performing Standardized Computer Analyses for Licensing Evaluation (SCALE). The SCALE system was conceived and funded by the U. S. Nuclear Regulatory Commission to satisfy a strong need for a standardized method of analysis for the evaluation of nuclear fuel facility and package designs. Thus, the SCALE system will lead to a common understanding on how these designs will be evaluated by the NRC.

The overall goal of the SCALE project has been to develop easy-to-use analytical sequences which are automated to perform the necessary data processing (e.g., cross-section preparation) and manipulation of well-established computer codes (functional modules) required by the sequence. Thus, the user is able to select an analytical sequence characterized by the type of analysis (criticality, shielding, or heat transfer) to be performed and the geometric complexity of the system being analyzed. The user then prepares a single set of input for the control module corresponding to this analytical sequence. The control module input is in terms of easily visualized engineering parameters specified in a simplified, free-form format. The control modules use this information to derive additional parameters and prepare the input for each of the functional modules in the analytical sequence. Provisions have also been made to allow the user to execute the functional modules on a stand-alone basis. Stand-alone execution of the functional modules enables the SCALE system to be used for a wide range of nuclear and thermal applications.

In developing the SCALE system, an attempt has been made to adhere to certain criteria that would make SCALE an attractive computational tool for both the novice and experienced analyst. Some of the criteria are to:

a) develop a system driver which minimizes the computer core and running time,

b) develop control modules which require only a simplified set of input,

c) develop control modules which automate and couple the data processing and systems analysis into standard analytical sequences,

d) provide self-contained and thorough documentation for the entire system,

e) use well-established functional modules and data libraries that have a wide range of applicability and a proven record of reliable performance.

The overall objective of following these criteria was to develop SCALE as a system leading to analytical standardization.

The control modules available with SCALE are documented in Volume 1 of this manual. The SCALE system contains a wide assortment of control modules for performing criticality, shielding, and heat transfer analysis. Criticality Safety Analysis Sequences are referred to as CSAS1-CSASn and are described in Sections C1-Cn. Likewise, the shielding and heat transfer analysis sequences are called SAS1-SASn and HTAS1-HTASn, respectively and are described in Sections S1-Sn and H1-Hn, respectively.

Volume 2 of this manual provides documentation for each of the well-established functional modules used within the SCALE system. These sections are referred to as Sections F1-Fn. The functional modules include codes for data and/or cross section preparation (e.g. BONAMI-S and NITAWL-S), criticality analysis (e.g. XSDRNPM-S, KENO-IV/S, KENO-V), shielding analysis (e.g., XSDRNPM-S and MORSE-SGC/S), heat transfer analysis (HEATING6), and depletion analysis (ORIGEN-S). The "S" attached to the name of some functional modules indicates that the code was originally developed outside of SCALE and modified for use in the SCALE system. The exception to this is the HEATING6 code which is an upgrade of HEATING5.

Volume 3 of this manual provides documentation for the miscellaneous (but important) components of the SCALE system. These sections are referred to as Sections M1-Mn and include descriptions of the system driver, the subroutine library, the input systems, and the data, material composition, and property libraries. The data libraries include a variety of well-established cross section libraries, as well as some new libraries developed within the SCALE project.

The SCALE Manual is being released in a modular fashion; i.e., as each new component of the SCALE system is developed and released, a new section will be added to the SCALE Manual. Also, the documentation is released in looseleaf form so that as corrections and small revisions are made to the various modules, new or revised pages can be easily issued for the Manual. The List of Revisions issued with each release of new documentation provides a list of the NRC publication dates for each new or revised section and/or pages.

The SCALE system is being packaged and distributed by the Radiation Shielding Information Center (RSIC), located at Oak Ridge National Laboratory. A limited version of SCALE (SCALE-0) was made available to RSIC in July 1980. Significant revisions and additions have been made to SCALE and new versions were released as SCALE-1 in 1982 and SCALE-2 in 1983. This latest issuance of documentation should upgrade the existing documentation to match the SCALE-3 version of the system presently being issued by RSIC. Only IBM versions of SCALE are currently available from RSIC with the exception of SCALE-0 for which a CDC version is available. Requests for the latest SCALE code package can be mailed to:

Radiation Shielding Information Center
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

or telephoned to

Area Code 615; 574-6176 or to FTS 624-6176.

C. V. Parks
SCALE Project Leader
Oak Ridge National Laboratory

LIST OF REVISIONS for NUREG/CR-0200,
SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

Page/Section	New/Revised/Corrected*	Date
Table of Contents	Revised	December 1984
Foreword	Revised	December 1984
Sect. C1	Revised	January 1982
p. C1.A.5	Corrected	December 1984
Sect. C2	New	January 1982
Sect. C4	New	January 1982
title page	Revised	December 1984
p. C4.iii-vii	Revised	December 1984
Sect. C4.4	Revised	December 1984
Sect. S2	New	June 1983
title page	Revised	December 1984
p. S2.2.9	Revised	December 1984
p. S2.4.7	Revised	December 1984
p S2.5.5	Revised	December 1984
Sect. S3	New	January 1982
p. S3.A.1	Revised	June 1983
Sect. H1	New	June 1983
title page	Revised	December 1984
p. H1.iii-xi	Revised	December 1984
p. H1.xiii	Deleted	December 1984
p. H1.2.15	Revised	December 1984
p. H1.2.20	Revised	December 1984
p. H1.2.21	New	December 1984
p. H1.3.4	Revised	December 1984
p. H1.3.6	Revised	December 1984
p. H1.3.11-14	Revised	December 1984
p. H1.4.3	Revised	December 1984
p. H1.4.15	Revised	December 1984
p. H1.4.17-18	Revised	December 1984
p. H1.4.20	Revised	December 1984
p. H1.4.25-26	Revised	December 1984
p. H1.4.30	Revised	December 1984
p. H1.4.31-32	Revised	December 1984
Sect. H1.5	Revised	December 1984
Sect. H1.6	Revised	December 1984
Sect. H1.A	Revised	December 1984
Sect. H1.B	Revised	December 1984
Sect. H1.C	Deleted	December 1984
Sect. F1	New	January 1982
p. F1.2.2	Corrected	June 1983
p. F1.2.5	Corrected	June 1983
p. F1.5.1	Corrected	June 1983
p. F1.5.2	Corrected	December 1984
p. F1.6.2	Corrected	July 1982
p. F1.6.11	Corrected	July 1982

Page/Section	New/Revised/Corrected*	Date
Sect. F2	New	January 1982
p. F2.2.3	Corrected	June 1983
p. F2.2.15	Revised	June 1983
pp.F2.3.33-34	Corrected	July 1982
p. F2.3.41	Corrected	July 1982
p. F2.4.4	Corrected	December 1984
Sect. F3	New	June 1983
p. F3.12.6	Corrected	December 1984
Sect. F4	New	June 1983
Sect. F5	New	January 1982
p. F5.6.2	Corrected	June 1983
p. F5.B.3	Corrected	December 1984
Sect. F6	New	January 1982
title page	Revised	December 1984
p. F6.4.6	Revised	December 1984
p. F6.5.14	Revised	December 1984
Sect. F7	New	January 1982
title page	Revised	December 1984
p. F7.iii	Revised	December 1984
p. F7.v-vi	Revised	June 1983
p. F7.ix	Revised	December 1984
p. F7.2.15	Revised	June 1983
p. F7.2.27-28	New	December 1984
p. F7.3.3-4	Revised	December 1984
p. F7.4.2	Revised	December 1984
p. F7.5.11	Revised	December 1984
p. F7.6.2	Revised	June 1983
p. F7.6.4	Revised	December 1984
p. F7.6.6	Revised	June 1983
p. F7.6.8	Revised	December 1984
p. F7.6.10-11	Revised	December 1984
p. F7.6.15-19	Revised	December 1984
p. F7.6.21	Revised	June 1983
p. F7.6.22-23	Revised	December 1984
p. F7.6.25-26	Revised	June 1983
p. F7.6.27-28	Revised	December 1984
p. F7.6.30	Revised	December 1984
p. F7.6.33-34	Revised	June 1983
p. F7.6.37-38	Revised	December 1984
p. F7.6.39-40	Revised	June 1983
p. F7.6.43	Revised	June 1983
p. F7.6.44	Revised	December 1984
p. F7.7.1	Revised	December 1984
p. F7.8.1	Revised	December 1984
p. F7.9.2	Revised	December 1984
Sect. F8	New	June 1983

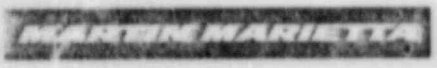
Page/Section	New/Revised/Corrected*	Date
Sect. F9	New	January 1982
title page	Revised	December 1984
p. F9.A.1	Revised	December 1984
p. F9.A.12	Revised	December 1984
p. F9.A.13	Revised	December 1984
p. F9.B.6	Corrected	December 1984
p. F9.B.7-9	Revised	December 1984
p. F9.B.10	Deleted	December 1984
Sect. F10	New	January 1982
p. F10.iv	Revised	June 1983
p. F10.iv	Corrected	December 1984
p. F10.2.23	Corrected	December 1984
p. F10.3.38	Revised	June 1983
p. F10.5.9-12	Revised	June 1983
p. F10.5.9	Corrected	December 1984
p. F10.5.14	Corrected	December 1984
p. F10.5.35	Revised	June 1983
p. F10.5.38	Corrected	December 1984
p. F10.A.1-2	Corrected	December 1984
p. F10.A.4-5	Corrected	December 1984
p. F10.C.36	Revised	June 1983
Sect. F11	Revised	December 1984
Sect. F12	Revised	December 1984
Sect. F13	New	December 1984
Sect. F14	New	December 1984
Sect. M1	New	January 1982
Sect. M2	New	January 1982
p. M2.2.58	New	June 1983
p. M2.2.73	Revised	December 1984
Sect. M3	New	January 1982
Sect. M6	New	June 1983
title page	Revised	December 1984
p. M6.v	Revised	December 1984
p. M6.vii	Revised	December 1984
p. M6.4.1-2	Revised	December 1984
p. M6.7.4	Revised	December 1984
p. M6.7.6-10	Revised	December 1984
p. M6.8.3	Revised	December 1984
Sect. M8	Revised	June 1983
title page	Revised	December 1984
p. M8.2.2-13	Revised	December 1984
p. M8.4.1	Revised	December 1984

Page/Section	New/Revised/Corrected*	Date
Sect. M9	New	January 1982
title page	Revised	December 1984
p. M9.iv	Revised	December 1984
p. M9.vii	Revised	December 1984
p. M9.6.5-6	Revised	December 1984
p. M9.E.2-3	Corrected	December 1984
p. M9.E.4-6	Revised	December 1984
p. M9.E.14-15	Revised	December 1984
p. M9.E.16-19	New	December 1984
Sect. M9.G	New	December 1984
Sect. M10	New	January 1982
Sect. M11	New	January 1982
Sect. M12	New	December 1984
Sect. M13	New	December 1984
Sect. M14	New	December 1984

* New = First issuance of document section or page
Revised = Revision of document section or pages to reflect code updates or changes
Corrected = Correction to reflect error in previous issuance of document

NUREG/CR-0200
VOLUME 2
ORNL/NUREG/CSD-2/V2/R2

OAK RIDGE
NATIONAL
LABORATORY



SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

Volume 2: Functional Modules

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
Under Interagency Agreement DOE 40-549-75

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

NUREG/CR-0200
Volume 2, Section F1
ORNL/NUREG/CSD-2/V2/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at Oak Ridge Gaseous Diffusion Plant

HEATPLOT-S: A TEMPERATURE DISTRIBUTION PLOTTING PROGRAM FOR HEATING5/HEATING6

C. K. Cobb

Manuscript Completed: September 1984

Date Published: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75

NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

CONTENTS

LIST OF FIGURES F13.v

LIST OF TABLES F13.vii

ACKNOWLEDGMENTS F13.ix

ABSTRACT F13.xi

F13.1 INTRODUCTION AND CAPABILITIES F13.1.1

F13.1.1 INTRODUCTION F13.1.1
F13.1.2 CAPABILITIES F13.1.1

F13.1.2.1 Temperature Contours F13.1.2
F13.1.2.2 Temperature-Time Profiles F13.1.2
F13.1.2.3 Temperature-Distance Profiles F13.1.3

F13.2 LOGICAL PROGRAM FLOW AND DESCRIPTION OF SUBROUTINES F13.2.1

F13.2.1 DESCRIPTION OF SUBROUTINES IN HEATPLOT-S F13.2.1
F13.2.2 OAK RIDGE SYSTEMS ROUTINES USED BY HEATPLOT-S F13.2.3

F13.3 INPUT DESCRIPTION FOR HEATPLOT-S F13.3.1

F13.3.1 INTRODUCTION F13.3.1
F13.3.2 AXIS LENGTHS DATA RECORD F13.3.1
F13.3.3 INPUT PARAMETERS DATA RECORD 1 ..... F13.3.2
F13.3.4 INPUT PARAMETERS DATA RECORD 2 ..... F13.3.3
F13.3.5 PROBLEM UNITS DATA RECORD F13.3.4
F13.3.6 CONTOUR VALUES DATA RECORD F13.3.4
F13.3.7 PLOT TIMES DATA RECORD F13.3.4
F13.3.8 X OUTPUT PLANES DATA RECORD F13.3.4
F13.3.9 Y OUTPUT PLANES DATA RECORD F13.3.4
F13.3.10 Z OUTPUT PLANES DATA RECORD F13.3.4
F13.3.11 SCALING FACTORS DATA RECORD 1 ..... F13.3.5
F13.3.12 SCALING FACTORS DATA RECORD 2 ..... F13.3.5
F13.3.13 PLOT REGION DATA RECORD F13.3.5
F13.3.14 NODES DATA RECORD F13.3.5
F13.3.15 NUMBER OF PROFILES DATA RECORD F13.3.6
F13.3.16 X PROFILES DATA RECORD F13.3.6
F13.3.17 Y PROFILES DATA RECORD F13.3.6
F13.3.18 Z PROFILES DATA RECORD F13.3.6

F13.4 SAMPLE PROBLEMS FOR HEATPLOT-S F13.4.1
F13.5 REFERENCES F13.5.1
F13.A APPENDIX A - CONTROL CARDS FOR EXECUTING HEATPLOT-S ON THE OAK RIDGE IBM SYSTEMS F13.A.1
F13.B APPENDIX B - MODIFYING THE BLOCK DATA ROUTINE F13.B.1
F13.C APPENDIX C - DESCRIPTION OF TEMPERATURE DISTRIBUTION PLOT DATA SET WRITTEN BY HEATING5/HEATING6 ..... F13.C.1

LIST OF FIGURES

F13.2.1 General flow chart of HEATPLOT-S F13.2.2

F13.4.1 Input data for sample problem 1 ..... F13.4.2

F13.4.2 Output for sample problem 1 ..... F13.4.3

F13.4.3 Plots generated for sample problem 1 ..... F13.4.5

F13.4.4 Input data for sample problem 2 ..... F13.4.9

F13.4.5 Output for sample problem 2 ..... F13.4.10

F13.4.6 Plots generated for sample problem 2 ..... F13.4.12

F13.A.1 Format of JCL for executing HEATPLOT-S through SCALE at ORNL and generating plots on a CALCOMP plotter F13.A.2

F13.A.2 Format of JCL for executing HEATPLOT-S through SCALE at ORGDP and generating plots on a CALCOMP plotter F13.A.3

F13.A.3 Format of JCL for executing HEATPLOT-S through SCALE at ORNL and generating plots on an 11-inch Versatec plotter F13.A.4

F13.A.4 Format of JCL for executing HEATPLOT-S through SCALE at ORNL and storing the plot file in the user's PDP-10 area F13.A.5

F13.B.1 Default BLOCK DATA routine in HEATPLOT-S F13.B.2

F13.B.2 Format of JCL for executing HEATPLOT-S through SCALE at ORNL while overriding BLOCK DATA and generating plots on a CALCOMP plotter F13.B.3

LIST OF TABLES

F13.C.1 Variables that appear in HEATING5/HEATING6 plot data set generation WRITE statements F13.C.2

ACKNOWLEDGMENTS

This work was funded by the Transportation Branch, Nuclear Material Safety and Safeguards, U. S. Nuclear Regulatory Commission. The author expresses appreciation to C. V. Parks of Computing and Telecommunications Division (CTD), who served as project leader for this work, for review of the document and for providing general advice; and to L. M. Petrie of CTD for providing assistance and information concerning use of the SCALE System. Appreciation is also expressed to R. H. Odegaarden of the Nuclear Regulatory Commission, for review of the document; and to G. E. Giles of CTD for review of the document. Finally, the author expresses appreciation to A. I. Rose and B. R. Coppin of CTD for typing and preparation of the document, and to Lindy Norris of CTD for the layout of the text.

ABSTRACT

HEATPLOT-S is a temperature distribution plotting program that may be used with the HEATING5 or HEATING6 generalized heat conduction codes. HEATPLOT-S has the capability of generating temperature contours, temperature-time profiles, and temperature-distance profiles from the current HEATING5/HEATING6 temperature distribution or from the temperature changes relative to the initial temperature distribution. Contour plots may be generated for two- or three-dimensional models. Temperature-time and temperature-distance profiles may be generated for one-, two-, or three-dimensional models. HEATPLOT-S uses the DISSPLA (Display Integrated Software System and Plotting Language) graphics package, and generates a compressed plot data set that can be stored on disk or routed to various plotting devices.

F13.1 INTRODUCTION AND CAPABILITIES

This document has been developed to report the changes that have been made to the HEATPLOT[1] computer program in order to make it compatible with the SCALE system. Much of the material in this document is taken from Ref. 1. However, no other document is needed to use this version of the program.

F13.1.1 INTRODUCTION

HEATPLOT-S may be used with the HEATING5[2] or HEATING6 (see Sect. F10) generalized heat conduction codes to generate plots of temperature distributions. HEATPLOT-S has the capability of generating temperature contours, temperature-time profiles, and temperature-distance profiles. The contours and temperature profiles may be generated from the current temperature distribution or from temperature changes relative to the initial temperature distribution. HEATPLOT-S requires as input the data set created by using the plotting option of HEATING5/HEATING6 and user-supplied card image input data. HEATPLOT-S determines from the card image input data the temperature, time, and distance units, the types of plots generated, how the plots are scaled, and the problem times for which the plots are generated. For temperature contour plots HEATPLOT-S determines the number of contours to be plotted, the planes and region for which contours are plotted, and whether or not the axes are reversed and/or switched. For temperature-time profile plots and temperature-distance profile plots HEATPLOT-S determines the nodes or lines for which profiles are plotted and whether each profile appears on a separate plot or whether all profiles appear on one plot.

F13.1.2 CAPABILITIES

HEATPLOT-S has the capability of generating temperature contours, temperature-time profiles, and temperature-distance profiles for two- or three-dimensional models. Temperature-time and temperature-distance profiles may also be generated for one-dimensional models. HEATPLOT-S uses the DISSPLA[3] graphics package. A compressed plot data set is generated that can be stored on disk or routed to various plotting devices.

The lengths of the axes for the plots are input to the program. If the axis lengths are not specified or are specified as zero, default values of six inches are used for the horizontal and vertical axes of all plots. The page size used in the horizontal direction is four inches greater than the length of the horizontal axis. The page size used in the vertical direction is three inches greater than the length of the vertical axis.

The temperature, time, and distance units are specified with up to four alphanumeric characters. The units are written in axis labels and in plot legends.

HEATPLOT-S uses the job description from the HEATING5/HEATING6 input as the title of all plots. Only the first 39 characters of the job description are used due to space limitations. Corresponding Greek letters are written in the plot title if English letters are surrounded by question marks in the job description. For example, including the characters ?P? in the HEATING5/HEATING6 job description causes the character π to be written in the plot title. Therefore, extraneous question marks should not be included in the HEATING5/HEATING6 job description.

F13.1.2.1 *Temperature Contours*

Temperature contours may be generated for two- or three-dimensional models. Contours may be generated of temperatures from the current temperature distribution or of temperature changes relative to the initial temperature distribution. The maximum number of contour values that can be plotted is fifty. The contour values may be specified by the user or can be determined by the program. If HEATPLOT-S determines the contour values, the minimum and maximum temperatures of the distribution to be plotted are computed, and the difference is computed and divided by the number of contours to be generated. The resulting number is the interval between the contours plotted. The value of the first contour plotted is the minimum temperature plus half the interval between the contours.

Axis scaling factors for temperature contour plots may be input to HEATPLOT-S. A scale factor is defined as the difference of the maximum and minimum axis values divided by the axis length (or units per inch). For example, if the x-axis for a temperature contour plot is six inches long and begins at zero and an axis scaling factor of 10.0 is specified, the maximum value of the x-axis is 60.0. If the axis scaling factors for temperature contour plots are not specified or are specified as zero, the scaling factors are calculated by the program.

The times of the temperature distributions from which temperature contour plots are generated may be input to HEATPLOT-S. If no times are specified, temperature contour plots are generated for every temperature distribution on the HEATING5/HEATING6 plot data set. Contour plots are generated for a maximum of twenty times.

The region for which contours are plotted may be input to allow the user to plot contours for a selected portion of the HEATING5/HEATING6 model. The region specified for which temperature contour plots are generated is also the region for which temperature-distance profile plots are generated. The region is specified by fine grid line numbers of the HEATING5/HEATING6 model. If the region is not specified, temperature contours and temperature-distance profiles are plotted for the entire specified plane.

When plotting temperature contours for a three-dimensional model, the user must specify the number of planes and their positions normal to the x, y, and z (or r, θ, and z) axes for which temperature contours are plotted. The positions of the planes are specified by fine grid line numbers of the HEATING5/HEATING6 model.

HEATPLOT-S draws indentations or holes if they appear in the region of the model for which contours are plotted. However, an indentation is recognized only if there is a fine grid line passing through the interior of the indentation parallel to each axis.

The user may specify that any axis be reversed. The axis values will range from maximum to minimum instead of from minimum to maximum. The user may also specify that the axes be switched. If the axes are switched, the vertical axis is drawn horizontally and the horizontal axis is drawn vertically.

F13.1.2.2 *Temperature-Time Profiles*

Temperature-time profiles may be generated for one-, two-, or three-dimensional models. Profiles of up to twenty nodes may be plotted. Nodal temperatures may be plotted or nodal temperature change from the initial nodal temperature may be plotted. Profiles of each node may appear on separate plots, or the profiles of all nodes may appear on one plot.

Axis scaling factors may be input for the time and/or temperature axis. If axis scaling factors are not specified or are specified as zero, they are calculated by the program. The minimum and maximum times for which temperature-time profiles are plotted may also be input. If the minimum and maximum times are not specified or are specified as zero, temperatures at every problem time are used. Temperatures at up to 100 problem times on the plot data set can be used.

The time axis of temperature-time profiles may be linear or logarithmic. If the time axis is logarithmic and the first problem time is zero, the user should input the minimum time to be the second problem time. If the time axis is logarithmic and the minimum time is zero, HEATPLOT-S prints a warning message, and the plot may not be correct.

F13.1.2.3 *Temperature-Distance Profiles*

Temperature-distance profiles may be generated for one-, two-, or three-dimensional models. Current temperatures or temperature changes from the initial temperatures may be plotted. Profiles may appear on separate plots, or they may all appear on one plot. Temperature-distance profiles are generated for the same problem times and region used for temperature contour plots.

Axis scaling factors may be input for the temperature and/or distance axis. If axis scaling factors are not specified or are specified as zero, they are calculated by the program. The minimum temperature to appear on the temperature axis may also be input. If the minimum temperature is not specified or is specified as zero, it is also calculated by the program.

The total number of profiles to be plotted parallel to each axis must be input. A maximum of ten profiles can be plotted parallel to each axis. For two-dimensional problems, the fine grid line numbers along which the profiles are plotted are input. For three-dimensional problems, the fine grid line numbers which define the orthogonal planes along whose intersection the profiles are plotted are input.

The distance axis of temperature-distance profile plots may be linear or logarithmic. If the distance axis is logarithmic and the minimum distance for the specified profile is zero, HEATPLOT-S prints a warning message and the plots may not be correct.

F13.2 LOGICAL PROGRAM FLOW AND DESCRIPTION OF SUBROUTINES

HEATPLOT-S initially reads the user-supplied card image input data. Plotting parameters such as page size and axis lengths that are unchanged for all plots are then defined. Based on the card image input data, HEATPLOT-S then determines the borders of the region for which plots are generated, and the proper plot data set temperatures are stored in arrays for the problem times, nodes, and lines for which plots are generated. Temperature contour plots, if desired, are generated first. Temperature-distance profile plots, if desired, are then generated. Finally, temperature-time profile plots, if desired, are generated. The calling sequence of routines is depicted in Fig. F13.2.1.

F13.2.1 DESCRIPTION OF SUBROUTINES IN HEATPLOT-S

BLOCK DATA - Initializes variables in labeled common blocks NPOINT and NOPROF. Variables initialized in common block NPOINT contain the maximum number of nodes in the plot data set, the maximum number of r, θ, or z values in the plot data set, and the maximum number of values to be plotted on any horizontal and vertical axis. Variables initialized in common block NOPROF contain the maximum number of profiles to be generated, the maximum number of r, θ, or z values to be used for temperature-distance profiles, and the maximum number of times for which profiles are generated.

MAIN - Directs ALOCAT (see Sect. M2.2.1) to call subroutine ALCHTP.

ALCHTP - Serves as an interface between the HEATPLOT-S main routine and other subroutines. Determines if the region size specified to execute HEATPLOT-S is large enough based on the values of variables assigned in BLOCK DATA.

BORDER - Used in temperature contour plots to define the outer border of the region for which temperature contour lines are generated.

CONDRW - Generates all temperature contour lines.

CONTIN - Calls subroutines to store the proper temperature data for and to generate temperature-distance profile plots and temperature-time profile plots.

DFINEN - Sets flags to denote which regions are included in the plots.

ECHO - Reads card image input data from unit 5, and lists card images on unit 6. Puts 'END' at the end of the appropriate card images and writes the resulting card images to unit 4.

FIXLAX - Sets flags to determine which axes are reversed.

HEATPL - Performs initializations and calls subroutines to store the proper temperature data for and to generate temperature contour plots. Calls CONTIN to generate temperature-distance profile and temperature-time profile plots.

HOLES4 - Determines the arrays that define temperature contour lines.

HTPLMN - Performs variable dimensioning of arrays and calls HEATPL to generate plots.

INPLT - Inputs user-supplied card image input data and writes messages describing the input data.

INSERU - Places the literals to be used as plot labels in the UNITS array inside parentheses.

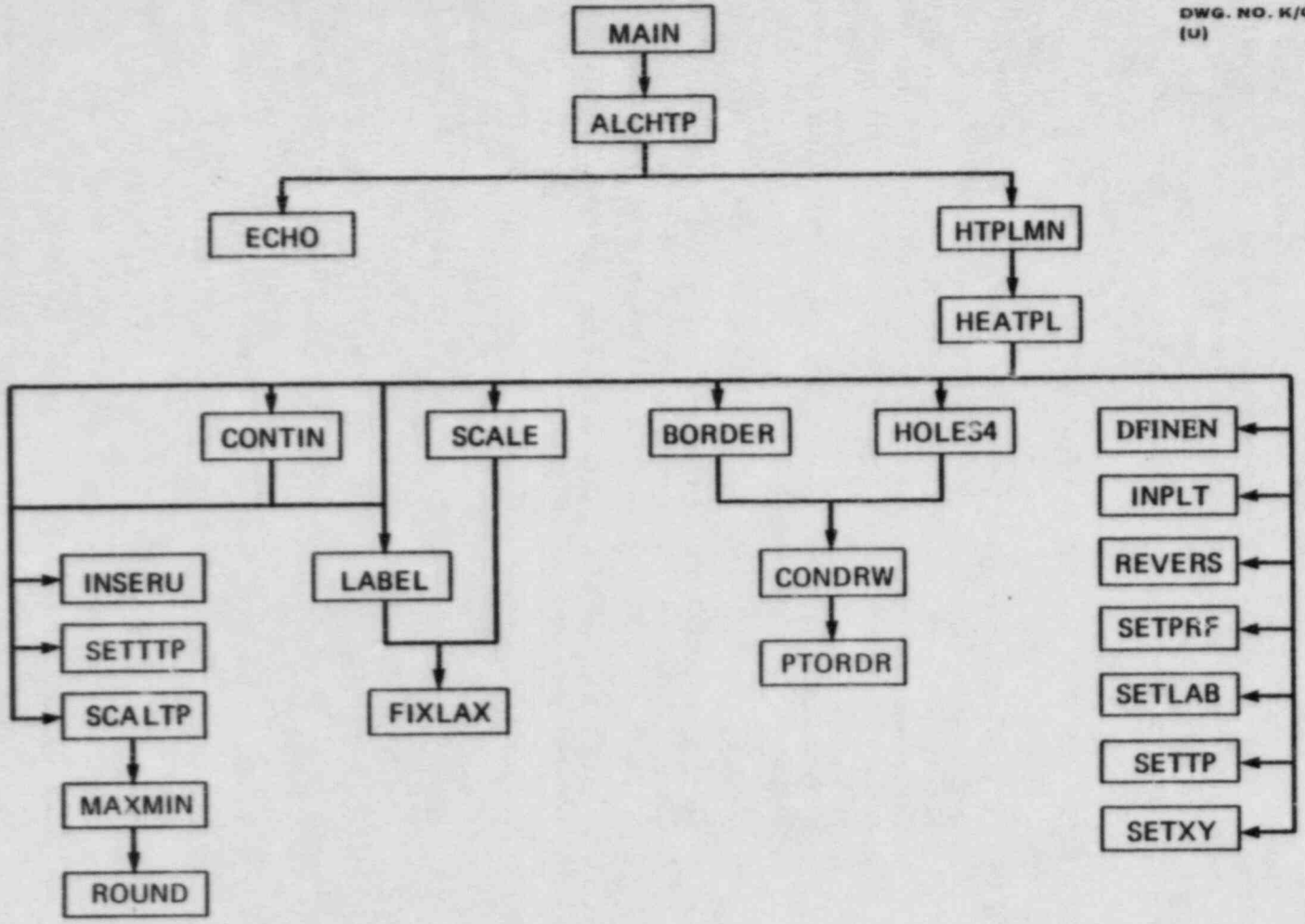


Fig. F13.2.1 General flow chart of HEATPLOT-S

LABEL - Draws and labels the axes, writes the heading, and generates the legend for each plot.

MAXMIN - Determines the maximum and minimum values of the array received in the argument list.

PTORDR - Puts x-coordinate values in ascending order for temperature contour plots.

REVERS - Reverses the values on any axis for which values are reversed.

ROUND - Performs rounding of the minimum and maximum axis values based on the minimum and maximum direction array values.

SCALE - Calculates the minimum and maximum values of the arrays to be plotted and calculates scaling factors for the horizontal and vertical axes.

SCALTP - Calls subroutines to perform axis scaling and generates curves for temperature-distance profile and temperature-time profile plots.

SETLAB - Determines the proper axis labels to be written depending on the type of plot being generated.

SETPRF - Finds the nodes which lie on the line whose profile will be plotted and stores the correct temperatures in arrays.

SETTP - Stores the correct temperatures in arrays depending on which plane is used for plotting for three-dimensional models.

SETTTP - Stores temperatures in arrays for temperature-time profile plots.

SETXY - Sets values for the horizontal and vertical axes depending on the type of plot being generated.

F13.2.2 OAK RIDGE SYSTEMS ROUTINES USED BY HEATPLOT-S

HEATPLOT-S calls Oak Ridge systems subroutines CORE, IDAY, and JOBNUM, and calls Oak Ridge systems function ICOMPA. These routines are all described in Table F10.4.2.

F13.3 INPUT DESCRIPTION FOR HEATPLOT-S

F13.3.1 INTRODUCTION

The input data for HEATPLOT-S are read using the SCALE free-form reading routines (see Sect. M3) which allow data to be entered in an unformatted manner. Data items can begin or end anywhere in columns 1 through 80. If the data for a particular data record will not fit on one card, the character '@' must appear in column 1 of each continuation card. There is no limit to the number of continuation cards that can be used. HEATPLOT-S uses the SCANON feature of the SCALE free-form reading routines, and the characters 'END' are written at the end of each card image that does not specify continuation. Therefore, trailing data items that are omitted for a card image are treated as zeroes. A comment card can be specified in the input stream by placing an asterisk in column 1 of a card image. In order to access HEATPLOT-S via SCALE, a card image containing either =HEATPLOT or #HEATPLOT, beginning in the first column, must precede the card image input data. A card image containing END, beginning in the first column, must follow the card image input data (see Sect. F13.A).

The following discussion concerning the free-form input data was taken from Sect. M3 after minor revisions. For HEATPLOT-S, any alphanumeric entry is terminated by one blank. Any numeric data entry is terminated when a blank or a comma is encountered. A comma must immediately follow the number if it is used as a terminator. The use of an E as an exponent does allow a single embedded blank after the E. An exponent is limited to two digits, and the number being read is terminated with the second digit. For this case only, therefore, another number can be started immediately following the second digit of the exponent without an intervening blank or comma. Decimal data may be entered as in FORTRAN input; e.g., 1.733-4, 1.733E-04, or 0.0001733 are the same as 1.733×10^{-4}. 1.733+4, 1.733E+4, 1.733E4, and 17330.0 are the same as 1.733×10^{4}. Integers may be entered for floating point data (i.e., 10 will be interpreted as 10.0). Similarly, floating point data may be entered for integers. Thus, 1.733E+4 would be interpreted as 17330 and 1.733E-4 would be interpreted as zero. The free-form routines have provisions for multiple entries of the same data value. This is done by entering the number of repeats, followed by an R, *, or $, followed by the data value to be repeated. For example, 5R2, 5*2, or 5$2 enters five successive 2's in the input data. Blanks are not allowed between the number of repeats and the repeat flag (R, *, $). However, a blank is allowed between the repeat flag and the data item to be repeated.

F13.3.2 AXIS LENGTHS DATA RECORD

Both entries for this data record are floating point numbers. This data record must be included.

XAX - The length of the horizontal axis in inches. If XAX is specified as zero, the default length of the horizontal axis is six inches.

YAX - The length of the vertical axis in inches. If YAX is specified as zero, the default length of the vertical axis is six inches.

F13.3.3 INPUT PARAMETERS DATA RECORD 1

All entries for this data record are integers. This data record must be included.

NL - The number of temperature contour values to be plotted is equal to the absolute value of NL, which cannot exceed 50.

< 0 Contour values are calculated by HEATPLOT-S.
= 0 No contour plots are generated.
> 0 Contour values are read from the CONTOUR VALUES data record.

NTIME - The number of problem times for which contours and temperature-distance profiles, if desired, are plotted. NTIME cannot exceed 20.

= 0 All problem times are used.
> 0 Problem times are read from the PLOT TIMES data record.

NPLANE(1) - The number of y-z or θ-z planes for which contours are plotted if the problem is three-dimensional. Enter zero if the problem is not three-dimensional. NPLANE(1) cannot exceed 10.

NPLANE(2) - The number of x-z or r-z planes for which contours are plotted if the problem is three-dimensional. Enter zero if the problem is not three-dimensional. NPLANE(2) cannot exceed 10.

NPLANE(3) - The number of x-y or r-θ planes for which contours are plotted if the problem is three-dimensional. Enter zero if the problem is not three-dimensional. NPLANE(3) cannot exceed 10.

ITMVTP - The number of nodes for which temperature-time profiles are plotted is equal to the absolute value of ITMVTP, which cannot exceed 20. The node numbers are read from the NODES data record.

< 0 Temperatures for all specified nodes are plotted on one plot.
= 0 No temperature-time profiles are generated.
> 0 Temperatures for specified nodes are plotted one node per plot.

LPLOT - Flag for temperature-distance profiles.

< 0 Profiles along a line at every problem time are plotted on one plot.
= 0 No temperature-distance profiles are generated.
> 0 A profile along a line for each problem time is plotted on a separate plot.

IRISCN - Flag to plot temperature changes for contour plots.

= 0 If NL is nonzero, temperature contours are plotted.
= 1 If NL is nonzero, contours of temperature changes relative to the initial temperature distribution are plotted.

IRISTM - Flag to plot temperature changes for temperature-time profile plots.

= 0 If ITMVTP is nonzero, temperatures are plotted.
= 1 If ITMVTP is nonzero, temperature changes are plotted.

IRISDS - Flag to plot temperature changes for temperature-distance profile plots.

= 0 If LPLOT is nonzero, temperatures are plotted.
= 1 If LPLOT is nonzero, temperature changes are plotted.

F13.3.4 INPUT PARAMETERS DATA RECORD 2

All entries for this data record are integers. This data record must be included.

NSCAL - Scaling factor flag.

= 0 Scaling factors are computed by HEATPLOT-S.
= 1 Scaling factors are read from SCALING FACTORS data records 1 and 2.

LREG - Flag to specify region for which temperature contours and temperature-distance profiles are plotted.

= 0 Plots are drawn for the entire region of each plane specified.
= 1 The region is read from the PLOT REGION data record.

LREV(1) - Flag to reverse x-axis of temperature contour plots.

= 0 X-axis ranges from minimum to maximum.
= 1 X-axis ranges from maximum to minimum.

LREV(2) - Flag to reverse y-axis of temperature contour plots.

= 0 Y-axis ranges from minimum to maximum.
= 1 Y-axis ranges from maximum to minimum.

LREV(3) - Flag to reverse z-axis of temperature contour plots.

= 0 Z-axis ranges from minimum to maximum.
= 1 Z-axis ranges from maximum to minimum.

LAX(1) - Flag to switch the y and z axes of temperature contour plots.

= 0 Axes are not switched.
= 1 Axes are switched.

LAX(2) - Flag to switch the x and z axes of temperature contour plots.

= 0 Axes are not switched.
= 1 Axes are switched.

LAX(3) - Flag to switch the x and y axes of temperature contour plots.

= 0 Axes are not switched.
= 1 Axes are switched.

LOGTIM - Flag to specify logarithmic scale for the time axis of temperature-time profile plots.

= 0 Linear scale is used.
= 1 Logarithmic scale is used. The minimum time should not be zero.

LOGDIS - Flag to specify a logarithmic scale for the distance axis of temperature-distance profile plots.

= 0 Linear scale is used.
= 1 Logarithmic scale is used. The minimum distance should not be zero.

F13.3.5 PROBLEM UNITS DATA RECORD

All entries for this data record are alphanumeric. Each entry can contain a maximum of four characters. This data record must be included.

UNITS(1) - Time units for axis labels and plot legends.

UNITS(2) - Temperature units for axis labels and plot legends.

UNITS(3) - Distance units for axis labels and plot legends.

F13.3.6 CONTOUR VALUES DATA RECORD

All entries for this data record are floating point numbers. Omit this data record if NL is not positive.

VALUE(I),I=1,NL - Contour values to be plotted.

F13.3.7 PLOT TIMES DATA RECORD

All entries for this data record are floating point numbers. Omit this data record if NTIME is zero.

TIME(I),I=1,NTIME - The problem times for which temperature contours and temperature-distance profiles are generated. If the time input is not a time on the temperature distribution plot data set, the time on the data set closest to the input time is used.

F13.3.8 X OUTPUT PLANES DATA RECORD

All entries for this data record are integers. Omit this data record if NPLANE(1) is zero.

IPLANE(I,1),I=1,NPLANE(1) - The numbers of the fine grid lines on the x-axis of the HEATING5/HEATING6 model for which y-z or θ-z contours are plotted.

F13.3.9 Y OUTPUT PLANES DATA RECORD

All entries for this data record are integers. Omit this data record if NPLANE(2) is zero.

IPLANE(I,2),I=1,NPLANE(2) - The numbers of the fine grid lines on the y-axis of the HEATING5/HEATING6 model for which x-z or r-z contours are plotted.

F13.3.10 Z OUTPUT PLANES DATA RECORD

All entries for this data record are integers. Omit this data record if NPLANE(3) is zero.

IPLANE(I,3),I=1,NPLANE(3) - The numbers of the fine grid lines on the z-axis of the HEATING5/HEATING6 model for which x-y or r-θ contours are plotted.

F13.3.11 SCALING FACTORS DATA RECORD 1

All entries for this data record are floating point numbers. HEATPLOT-S computes the values for any entries whose input value is zero. Omit this data record if NSCAL is zero.

RSCAL - The scaling factor in units per inch for the x-axis for temperature contour plots.

THSCAL - The scaling factor in units per inch for the y-axis for temperature contour plots.

ZSCAL - The scaling factor in units per inch for the z-axis for temperature contour plots.

DSCAL - The scaling factor in units per inch for the distance axis of temperature-distance profile plots.

TSCAL - The scaling factor in units per inch for the temperature axis of temperature-distance profile plots.

TVDMIN - The minimum temperature for temperature-distance profile plots.

F13.3.12 SCALING FACTORS DATA RECORD 2

All entries for this data record are floating point numbers. HEATPLOT-S computes the values for any entries whose input value is zero. Omit this data record if NSCAL is zero.

TMSCAL - The scaling factor in units per inch for the time axis of temperature-time profile plots.

TPSCAL - The scaling factor in units per inch for the temperature axis of temperature-time profile plots.

TVTMIN - The minimum temperature to appear on the temperature axis for temperature-time profile plots.

TIMMIN - The minimum time for temperature-time profile plots.

TIMMAX - The maximum time for temperature-time profile plots.

F13.3.13 PLOT REGION DATA RECORD

All entries for this data record are integers. Omit this data record if LREG is zero.

NB(1) - The number of the fine grid line of the minimum x value in the region.

NE(1) - The number of the fine grid line of the maximum x value in the region.

NB(2) - The number of the fine grid line of the minimum y value in the region.

NE(2) - The number of the fine grid line of the maximum y value in the region.

NB(3) - The number of the fine grid line of the minimum z value in the region.

NE(3) - The number of the fine grid line of the maximum z value in the region.

F13.3.14 NODES DATA RECORD

All entries for this data record are integers. Omit this data record if ITMVTP is zero.

NODE(I),I=1,ABS(ITMVTP) - The numbers of the nodes for which temperature-time profiles are plotted.

F13.3.15 NUMBER OF PROFILES DATA RECORD

All entries for this data record are integers. Omit this data record if LPLOT is zero, or if the problem is one-dimensional. The default maximum total number of profiles generated is ten as defined in BLOCK DATA. This number may be overridden in a user-supplied BLOCK DATA routine (see Appendix F13.A).

NPRFIL(1) - The number of temperature-distance profiles plotted parallel to the x-axis or normal to the y-z plane for two- or three-dimensional models.

NPRFIL(2) - The number of temperature-distance profiles plotted parallel to the y-axis or normal to the x-z plane for two- or three-dimensional models.

NPRFIL(3) - The number of temperature-distance profiles plotted parallel to the z-axis or normal to the x-y plane for two- or three-dimensional models.

F13.3.16 X PROFILES DATA RECORD

All entries for this data record are integers. Omit this data record if NPRFIL(1) is zero.

INDEX1(1,J),INDEX1(2,J),J=1,NPRFIL(1) - The numbers of the fine grid lines of the y and z axes, respectively, which define the x-z and x-y planes whose intersection is the line along which temperature-distance profiles are plotted parallel to the x-axis. For two-dimensional problems, the number of the fine grid line for the missing axis (y or z) must be one or zero.

F13.3.17 Y PROFILES DATA RECORD

All entries for this data record are integers. Omit this data record if NPRFIL(2) is zero.

INDEX2(1,J),INDEX2(2,J),J=1,NPRFIL(2) - The numbers of the fine grid lines of the x and z axes, respectively, which define the y-z and x-y planes whose intersection is the line along which temperature-distance profiles are plotted parallel to the y-axis. For two-dimensional problems, the number of the fine grid line for the missing axis (x or z) must be one or zero.

F13.3.18 Z PROFILES DATA RECORD

All entries for this data record are integers. Omit this data record if NPRFIL(3) is zero.

INDEX3(1,J),INDEX3(2,J),J=1,NPRFIL(3) - The number of the fine grid lines of the x and y axes, respectively, which define the y-z and x-z planes whose intersection is the line along which temperature-distance profiles are plotted parallel to the z-axis. For two-dimensional problems, the number of the fine grid line for the missing axis (x or y) must be one or zero.

F13.4 SAMPLE PROBLEMS FOR HEATPLOT-S

This section contains two sample problems using a three-dimensional x-y-z HEATING6 model to demonstrate the use of HEATPLOT-S. The input data used for each problem are listed, as is the output generated by HEATPLOT-S. Plots generated by HEATPLOT-S are also shown.

For the first sample problem, three temperature contours are generated for the x-y plane located at the first z fine grid line. The contour values are computed by the program. Temperature-distance profiles are generated along two lines parallel to the z-axis. The first line is located at the intersection of the x-z plane at the second x fine grid line and the y-z plane at the fourth y fine grid line. The second line is located at the intersection of the x-z plane at the fifth x fine grid line and the y-z plane at the ninth y fine grid line. Temperature contours and temperature-distance profiles are generated using the temperature distribution at the problem time of 0.19179 seconds. Temperature-time profiles are generated on one plot for three nodes for all problem times. All scaling factors are computed by the program. Input for this problem is shown in Fig. F13.4.1. Output generated for this problem is shown in Fig. F13.4.2. The four plots generated for this problem are shown in Fig. F13.4.3.

The same temperature contours are generated for the second sample problem as for the first sample problem. However, the x-axis for the temperature contour plot is reversed, and a scaling factor of 0.2 units per inch is input for the x-axis for the temperature contour plot. The contours are generated for the region between the third and sixth x fine grid lines, the first and eleventh y fine grid lines, and for all nine z fine grid lines. For the second sample problem, temperature-time profiles are generated for temperature changes relative to the initial temperature distribution. A scaling factor of 10.0 units per inch is input for the temperature axis, and the minimum temperature change is specified as -55 degrees. No temperature-distance profiles are generated for the second sample problem. Input for this problem is shown in Fig. F13.4.4. Output generated for this problem is shown in Fig. F13.4.5. The two plots generated for this problem are shown in Fig. F13.4.6.

```
* AXIS LENGTHS DATA RECORD
2*6.0
* INPUT PARAMETERS DATA RECORD 1
-3,1,0,0,1,-3,-1
* INPUT PARAMETERS DATA RECORD 2
0
* PROBLEM UNITS DATA RECORD
SEC C CM
* PROBLEM TIMES DATA RECORD
0.19179
* Z OUTPUT PLANES DATA RECORD
1
* NODES DATA RECORD
100,200,300
* NUMBER OF PROFILES DATA RECORD
2*0,2
* Z PROFILES DATA RECORD
2,4,5,9
```

Fig. F13.4.1 Input data for sample problem 1

```
THE LENGTH OF THE X-AXIS WILL BE  6.00 INCHES
THE LENGTH OF THE Y-AXIS WILL BE  6.00 INCHES

JOB DESCRIPTION - S#1 3DTA/8S/H=.04,L.A.W ROOF,KNOB,E.R,E=.34,TC=T2,8-22-83PLOT4/12/84
GEOMETRY TYPE NUMBER -   6(OR XYZ )
DATE OF RUN - 07-17-84
JOB NAME - CLIHP1SC
NUMBER OF NODES -  684
TOTAL NUMBER OF BOUNDARY CONDITIONS -    9

THE CONTOUR VALUES ARE CALCULATED BY THE PROGRAM
THE NUMBER OF CONTOURS TO BE PLOTTED IS    3

THE NUMBER OF TIME STEPS TO BE PLOTTED IS    1
THE TIME POSITIONS ARE:  1.9179E-01


THE NUMBER OF Z-PLANES TO BE PLOTTED IS   1
THE VALUES OF THE PLANE TO BE PLOTTED ARE:    1

ALL SCALE FACTORS WILL BE COMPUTED BY THE PROGRAM


UNITS -
      TIME IS IN SEC
      TEMPERATURE IS IN DEGREES C
      DISTANCE IS IN CM
```

Fig. F13.4.2 Output for sample problem 1

THE REGION OF THE GRAPH IS NOT SPECIFIED IN THE INPUT

INDEX DESCRIBING REGION FOR WHICH TO PLOT CONTOURS
BEGIN X 1 END X 7
BEGIN Y 1 END Y 15
BEGIN Z 1 END Z 9

THERE ARE 7 X (OR R) VALUES:

0.0	7.62000D-01	1.01600D+00	1.27000D+00	1.77800D+00
1.96850D+00	2.54000D+00			

THERE ARE 15 Y (OR TH) VALUES:

-1.24000D-01	-1.00000D-03	0.0	1.27000D+00	1.52400D+00
2.82575D+00	4.12750D+00	5.42925D+00	6.73100D+00	6.98500D+00
8.63600D+00	1.19380D+01	1.52400D+01	1.65100D+01	1.94000D+01

THERE ARE 9 Z VALUES:

0.0	9.50000D-01	1.82500D+00	2.70000D+00	3.57500D+00
4.45000D+00	2.38000D+01	4.31500D+01	6.25000D+01	

TEMPERATURE-TIME PROFILES WILL BE PLOTTED FOR 3 NODES ON ONE GRAPH
NODES TO BE PLOTTED FOR TEMPERATURE-TIME PROFILES 100 200 300

THERE WILL BE 2 TEMPERATURE-DISTANCE PROFILES PLOTTED FOR DISTANCES PARALLEL
TO THE Z AXIS, OR NORMAL TO THE X-Y OR R-TH PLANES
WITH ALL TIME STEPS ON ONE GRAPH
THE INDICES OF THE PLANES WHOSE INTERSECTION DEFINES THE LINE ALONG
WHICH THE PROFILES ARE PLOTTED FOLLOW:

NUMBER	X-PLANE	Y-PLANE
1	2	4
2	5	9

PROFIL NO.= 1

NODE NO.	RADIUS
23	0.0
120	9.50000D-01
217	1.82500D+00
314	2.70000D+00
411	3.57500D+00
508	4.45000D+00

TOTAL NUMBER OF NODES 6

PROFIL NO.= 2

NODE NO.	RADIUS
61	0.0
158	9.50000D-01
255	1.82500D+00
352	2.70000D+00
449	3.57500D+00
546	4.45000D+00

TOTAL NUMBER OF NODES 6

Fig. F13.4.2 (continued)

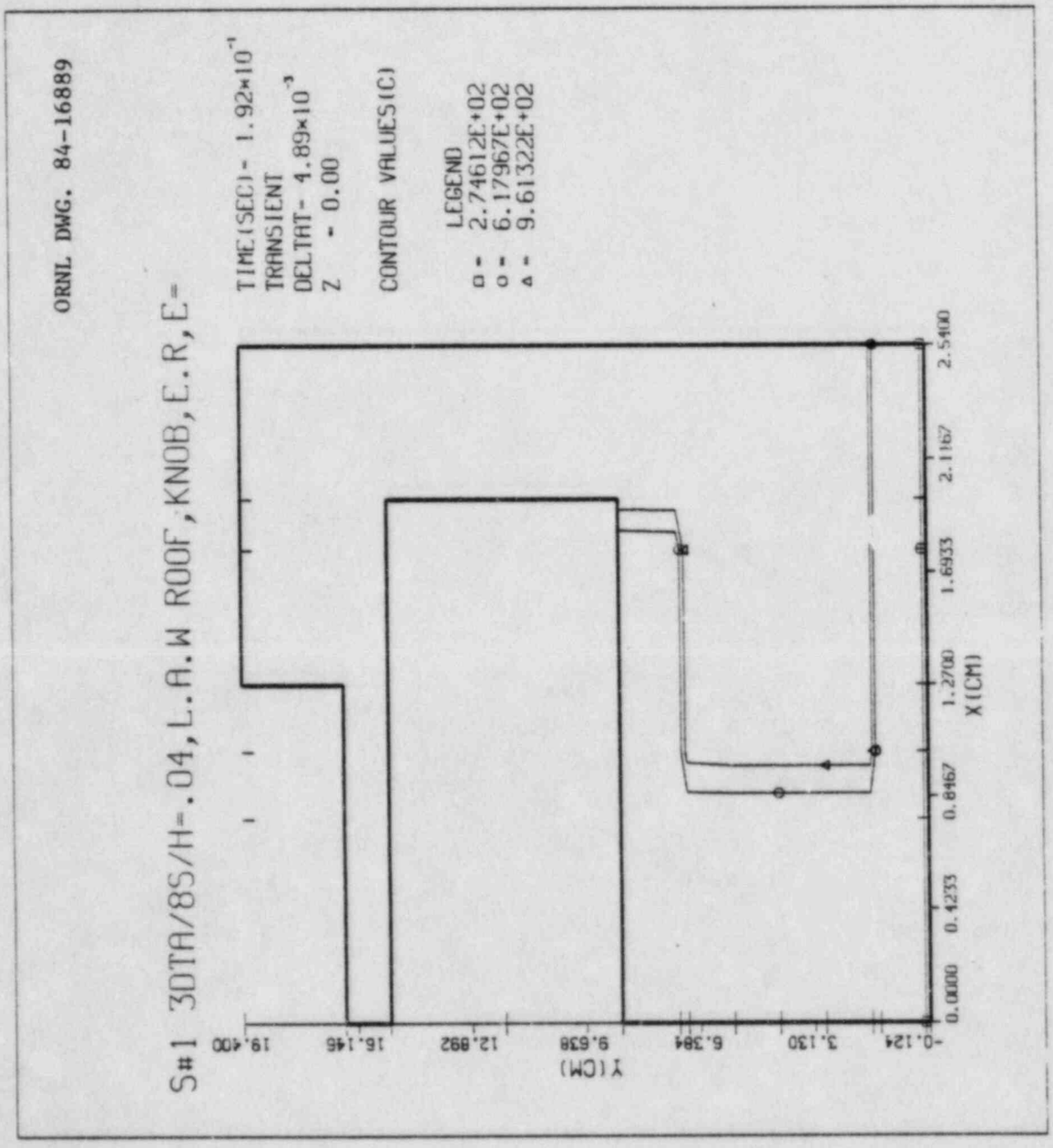


Fig. F13.4.3 Plots generated for sample problem 1

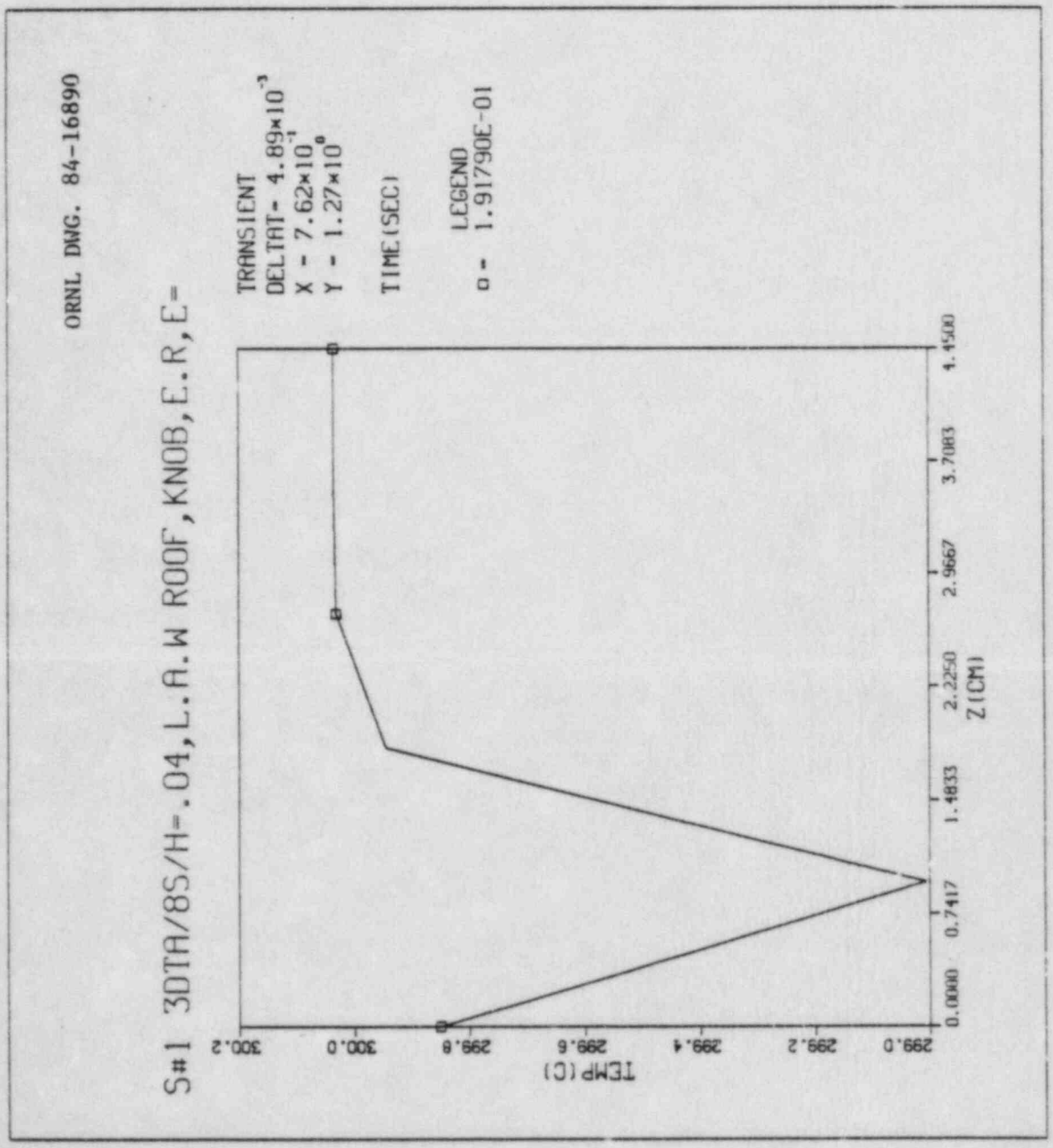


Fig. F13.4.3 (continued)

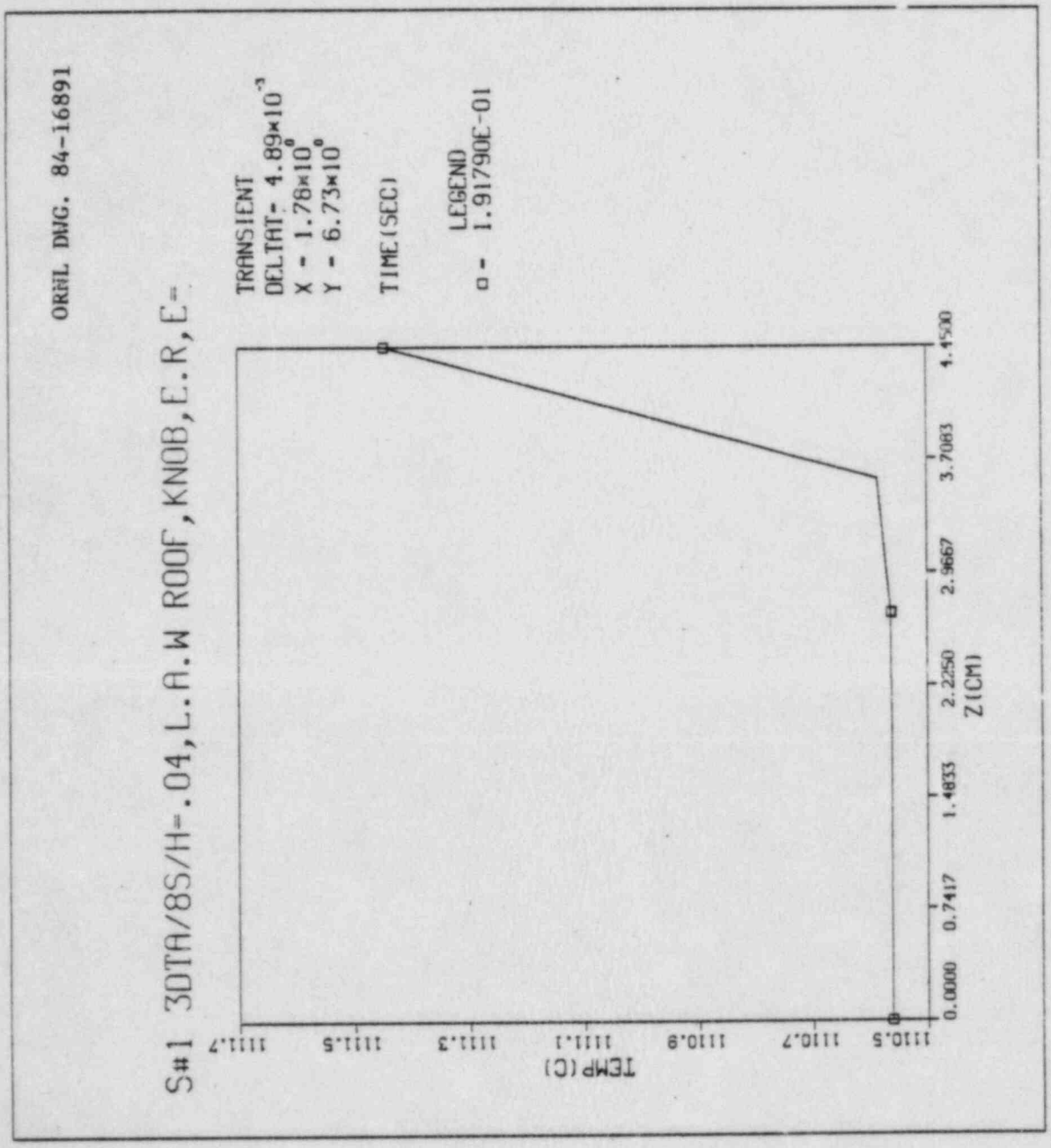


Fig. F13.4.3 (continued)

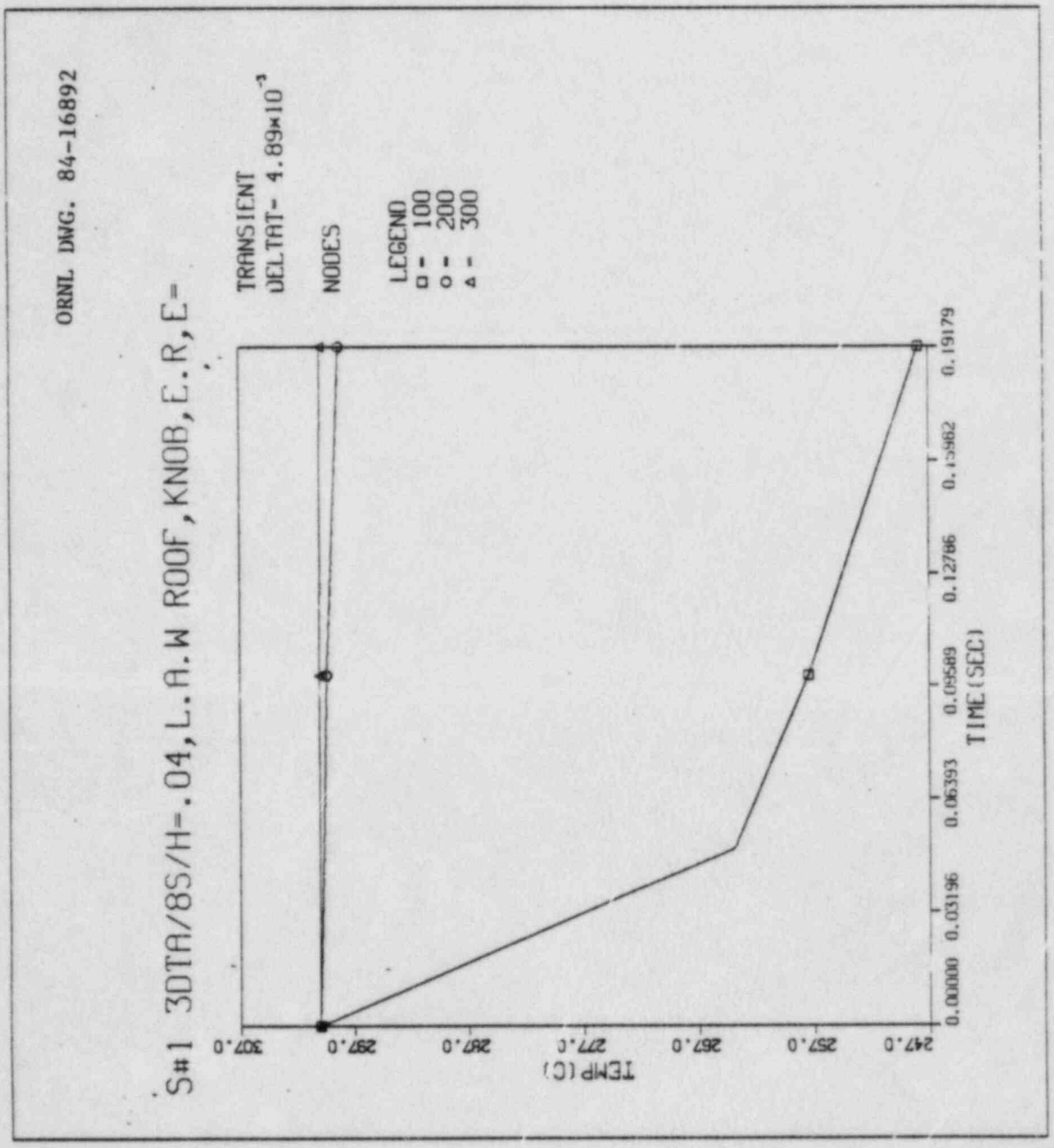


Fig. F13.4.3 (continued)

```
* AXIS LENGTHS DATA RECORD
2*6.0
* INPUT PARAMETERS DATA RECORD 1
-3,1,0,0,1,-3,0,0,1
* INPUT PARAMETERS DATA RECORD 2
3*1
* PROBLEM UNITS DATA RECORD
SEC C CM
* PROBLEM TIMES DATA RECORD
0.19179
* Z OUTPUT PLANES DATA RECORD
1
* SCALING FACTORS DATA RECORD 1
0.2
* SCALING FACTORS DATA RECORD 2
0.0,10.0,-55.0
* PLOT REGION DATA RECORD
3,6,1,11,1,9
* NODES DATA RECORD (NOTE CONTINUATION)
100,
@200,
@300
```

Fig. F13.4.4 Input data for sample problem 2

```
THE LENGTH OF THE X-AXIS WILL BE  6.00 INCHES
THE LENGTH OF THE Y-AXIS WILL BE  6.00 INCHES

JOB DESCRIPTION - S#1 3DTA/8S/H=.04,L.A.W ROOF,KNOB,E.R,E=.34,TC=T2,8-22-83PLOT4/12/84
GEOMETRY TYPE NUMBER -  6(OR XYZ )
DATE OF RUN - 07-17-84
JOB NAME - CLIHP2SC
NUMBER OF NODES -  684
TOTAL NUMBER OF BOUNDARY CONDITIONS -  9

THE CONTOUR VALUES ARE CALCULATED BY THE PROGRAM
THE NUMBER OF CONTOURS TO BE PLOTTED IS  3

THE NUMBER OF TIME STEPS TO BE PLOTTED IS  1
THE TIME POSITIONS ARE  1.9179E-01


THE NUMBER OF Z-PLANES TO BE PLOTTED IS  1
THE VALUES OF THE PLANE TO BE PLOTTED ARE:  1

THE SCALE FACTORS ARE READ BY THE PROGRAM
THOSE EQUAL TO ZERO ARE COMPUTED

THE INPUT SCALE FACTORS FOR THE CONTOUR PLOTS ARE:
      X OR R      2.0000E-01
      Y OR TH     0.0
      Z           0.0

THE INPUT SCALE FACTORS FOR THE TEMPERATURE-TIME PROFILES ARE:
      TIME                      0.0
      TEMPERATURE               1.0000E+01
      MINIMUM TEMPERATURE       -5.5000E+01
      MINIMUM TIME              0.0          MAXIMUM TIME   0.0

THE INPUT SCALE FACTORS FOR THE TEMPERATURE-DISTANCE PROFILES ARE:
      DISTANCE                  0.0
      TEMPERATURE               0.0
      MINIMUM TEMPERATURE       0.0


THE X (OR R) AXIS WILL BE FROM MAXIMUM TO MINIMUM


UNITS -
      TIME IS IN SEC
      TEMPERATURE IS IN DEGREES C
      DISTANCE IS IN CM
```

Fig. F13.4.5 Output for sample problem 2

```
THE REGION OF THE GRAPH IS SPECIFIED IN THE INPUT

INDEX DESCRIBING REGION FOR WHICH TO PLOT CONTOURS
      BEGIN X  3    END X  6
      BEGIN Y  1    END Y  11
      BEGIN Z  1    END Z  9

THERE ARE  7 X (OR R) VALUES:
      0.0             7.62000D-01    1.01600D+00    1.27000D+00    1.77800D+00
      1.96850D+00     2.54000D+00

THERE ARE  15 Y (OR TH) VALUES:
     -1.24000D-01    -1.00000D-03    0.0            1.27000D+00    1.52400D+00
      2.82575D+00     4.12750D+00    5.42925D+00    6.73100D+00    6.98500D+00
      8.63600D+00     1.19380D+01    1.52400D+01    1.65100D+01    1.94000D+01

THERE ARE  9 Z VALUES:
      0.0             9.50000D-01    1.82500D+00    2.70000D+00    3.57500D+00
      4.45000D+00     2.38000D+01    4.31500D+01    6.25000D+01

TEMPERATURE-TIME PROFILES WILL BE MADE OF TEMPERATURE CHANGES
      FOR  3 NODES ON ONE GRAPH
NODES TO BE PLOTTED FOR TEMPERATURE-TIME PROFILES  100  200  300
```

Fig. F13.4.5 (continued)

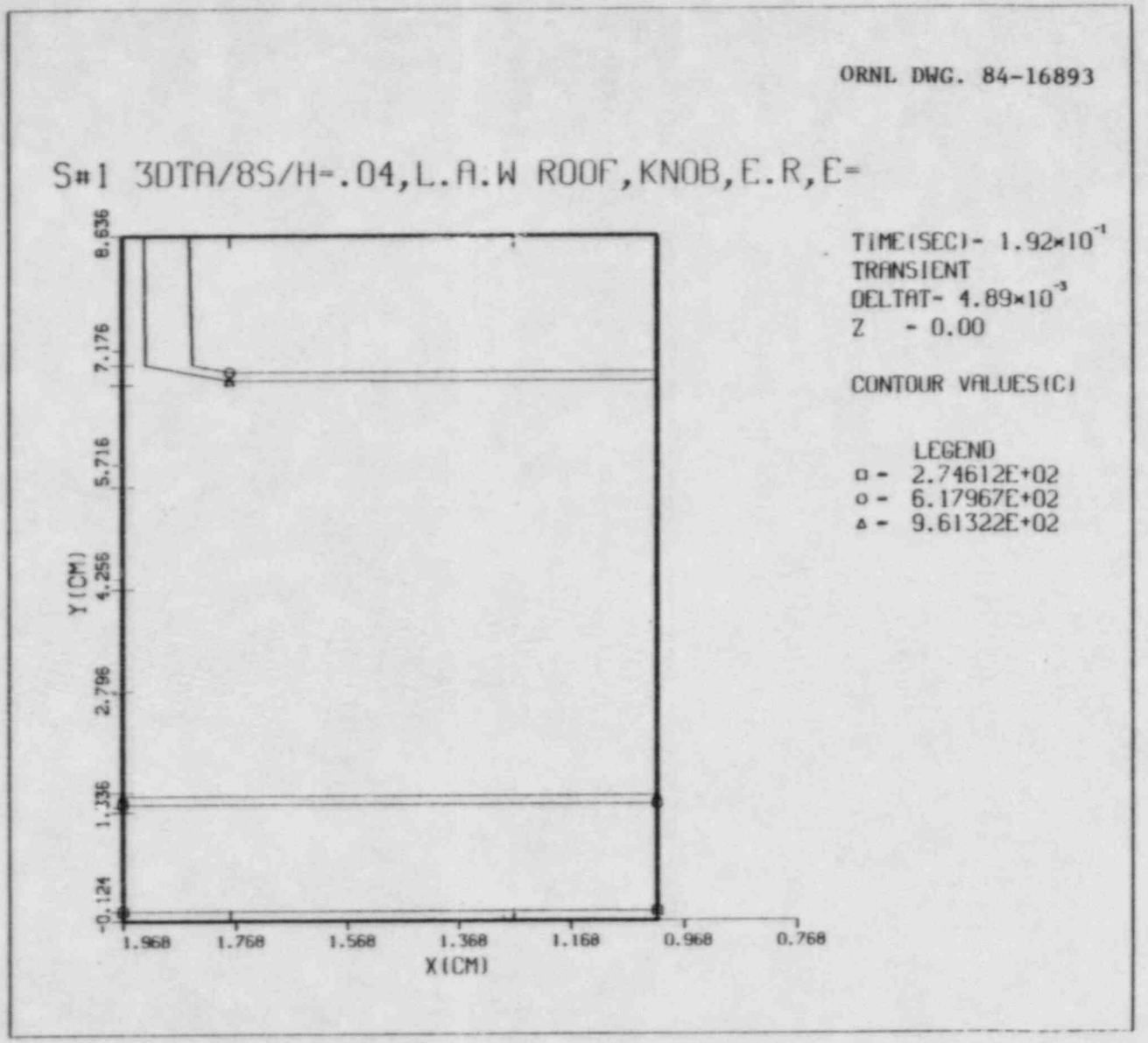


Fig. F13.4.6 Plots generated for sample problem 2

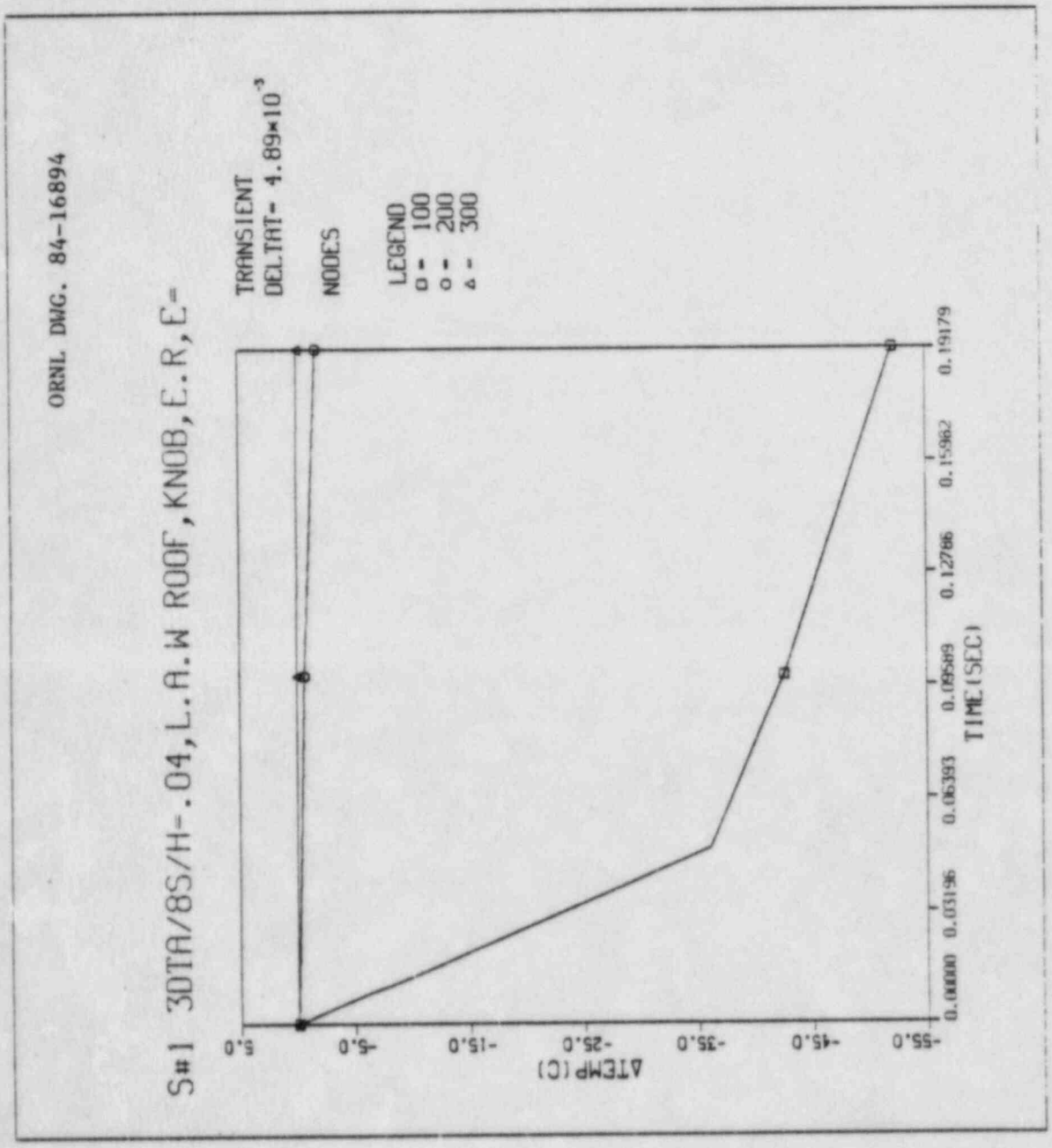


Fig. F13.4.6 (continued)

F13.5 REFERENCES

1. D. C. Elrod and W. D. Turner, *HEATPLOT - A Temperature Distribution Plotting Program for HEATING5*, K/CSD/TM-11, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, July 1977.
2. W. D. Turner, D. C. Elrod, and I. I. Simon-Tov, *HEATING5 - An IBM 360 Heat Conduction Program*, ORNL/CSD/TM-15, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, March 1977.
3. *DISSPLA User's Manual*, Integrated Software Systems Corporation, San Diego, California, 1981.
4. *CSD Programmer's Notebook*, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, March 10, 1981.
5. *CS Graphics Manual*, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, June 1, 1982.

F13.A APPENDIX A - CONTROL CARDS FOR EXECUTING HEATPLOT-S ON THE OAK RIDGE IBM SYSTEMS

The HEATPLOT-S module is part of the SCALE system at the computing centers at both Oak Ridge National Laboratory (ORNL) and Oak Ridge Gaseous Diffusion Plant (ORGDP). This appendix discusses the job control language (JCL) which is required to execute this version of HEATPLOT-S. Examples of JCL are presented which show how to execute HEATPLOT-S and generate plots on CALCOMP plotters at ORNL and ORGDP and on Versatec plotters at ORNL. An example of JCL which shows how to generate a compressed plot file and route the file to the user's PDP-10 area is also presented. For additional information concerning JCL and plotting devices, the user is referred to the CSD Programmer's Notebook[4] and the CS Graphics Manual.[5]

The control cards necessary to generate plots on a CALCOMP plotter using this version of HEATPLOT-S at ORNL and ORGDP are shown in Fig. F13.A.1 and Fig. F13.A.2, respectively. The jobname field on the JOB statement has a format of uidxxxxx, where uid is the three-letter user identification assigned to the user by CTD, and the characters xxxxx may be up to five characters which make the job name unique. The charge field, ccccc, contains a four- or five-digit CTD charge number. The programmer name field contains information such as the user's name and address which is necessary to route the job output back to the user. This field may contain up to twenty characters. The data set name assigned to the plot data set generated by HEATING5/HEATING6 replaces the characters 'HEATING plot data set.' All examples assume that the plot data set was cataloged when it was written by HEATING5/HEATING6.

The control cards necessary to generate plots on an 11-inch Versatec plotter at ORNL using this version of HEATPLOT-S are shown in Fig. F13.A.3. These control cards are identical to those shown in Fig. F13.A.1 with the exception of the plot spooler request (REQ=) parameters. Omitting the parameter PAPER=600 will cause plots to be generated on a 36-inch Versatec plotter at ORNL.

The control cards necessary to generate a compressed plot file and route the file to the user's PDP-10 area are shown in Fig. F13.A.4. The field ppppp,ppp should be replaced with the programmer number assigned to the user by CTD. The file is named uidxxx.PCH, where the characters uidxxx are the first six characters of the job name. The plot file can be sent from the user's PDP-10 area to various plotting devices on which the plots are generated.

```
//jobname JOB (ccccc,IO5),'programmer name',TIME=(1,0)
/*JOBPARM LINES=5
//STEP1 EXEC SCALE,REGION=510K
//GO.FT01F001 DD DSN=HEATING plot data set,DISP=SHR
//GO.SYSIN DD *
#HEATPLOT
 (Enter HEATPLOT-S Input Data Here)
END
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
//  SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=XC92.POP
/*
//
```

Fig. F13.A.1 Format of JCL for executing HEATPLOT-S through SCALE at ORNL and generating plots on a CALCOMP plotter

```
//jobname JOB(ccccc),'programmer name'
//*CLASS CPU95=1M,IO=5,L=5,REGION=510K
//STEP1 EXEC SCALE,REGION=510K
//GO.FT01F001 DD DSN=HEATING plot data set,DISP=SHR
//GO.SYSIN DD *
#HEATPLOT
 (Enter HEATPLOT-S Input Data Here)
END
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
// SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=KC92.POP
/*
//
```

Fig. F13.A.2 Format of JCL for executing HEATPLOT-S through SCALE at ORGDP and generating plots on a CALCOMP plotter

```
//jobname JOB (ccccc,IO5),'programmer name',TIME=(1,0)
/*JOBPARM LINES=5
//STEP1 EXEC SCALE,REGION=510K
//GO.FT01F001 DD DSN=HEATING plot data set,DISP=SHR
//GO.SYSIN DD *
#HEATPLOT
 (Enter HEATPLOT-S Input Data Here)
END
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
// SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=XV36.POP,PAPER=600
/*
//
```

Fig. F13.A.3 Format of JCL for executing HEATPLOT-S through SCALE at ORNL and generating plots on an 11-inch Versatec plotter

```
//jobname JOB (ccccc,IO5),'SAVEppppp,pppH',TIME=(1,0)
/*JOBPARM LINES=5,CARDS=10000
/*ROUTE PUNCH RMT45
//STEP1 EXEC SCALE,REGION=510K
//GO.FT01F001 DD DSN=HEATING plot data set,DISP=SHR
//GO.SYSIN DD *
#HEATPLOT
 (Enter HEATPLOT-S Input Data Here)
END
/*
//GO.COMPOUT DD DISP=(NEW,PASS),DSN=&&HTPLOT,UNIT=SPDA,
// SPACE=(TRK,(5,1),RLSE)
//STEP2 EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1 DD DSN=&&HTPLOT,DISP=(OLD,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//SYSUT2 DD SYSOUT=B,DCB=(RECFM=FB,BLKSIZE=80,LRECL=80)
//
```

Fig. F13.A.4 Format of JCL for executing HEATPLOT-S through SCALE at ORNL and storing the plot file in the user's PDP-10 area

F13.B APPENDIX B - MODIFYING THE BLOCK DATA ROUTINE

The default BLOCK DATA routine in HEATPLOT-S is shown in Fig. F13.B.1. This version of BLOCK DATA allows the user to use a temperature distribution plot data set that has up to 1000 nodes, with no more than 50 nodes in any one direction. There may be a total of 10 temperature-distance profiles generated at up to 20 problem times.

If any values in BLOCK DATA need to be increased to use a particular temperature distribution plot data set, the user can override the default BLOCK DATA routine by compiling his own BLOCK DATA routine and linking his BLOCK DATA routine with HEATPLOT-S. The control cards necessary to compile and link a user-supplied BLOCK DATA routine, execute HEATPLOT-S, and generate plots on a CALCOMP plotter at ORNL are shown in Fig. F13.B.2.

All variables listed in the default BLOCK DATA routine should appear in both a COMMON statement and a DATA statement in the user-supplied BLOCK DATA routine. MAXNDS must be greater than or equal to the number of nodes in the model. MAXVAL must be greater than or equal to the maximum number of nodes in any one direction. MAXHOR must be greater than or equal to the maximum number of values on the horizontal axis of any plot plus two. MAXVER must be greater than or equal to the maximum number of values on the vertical axis of any plot plus two. MAXPRF must be greater than or equal to the total number of temperature-distance profiles generated. NPRVAL must be greater than or equal to the maximum number of nodes in any one direction for which temperature-distance profiles are generated. NPRTIM must be greater than or equal to the number of problem times for which temperature-distance profiles are generated.

The region size required to execute HEATPLOT-S can be determined from the following two formulas:

$$N = 3*MAXNDS + 3*MAXVAL + 10*MAXHOR + 2*MAXHOR*MAXVER + 3*MAXPRF + NPRVAL*MAXPRF + NPRVAL*NPRTIM*MAXPRF$$

$$REGION = (N/128 + 356)K \text{ bytes.}$$

```
      BLOCK DATA
C
C ***************************************************************
C *
C *  MAXNDS = MAXIMUM NUMBER OF NODES IN THE TEMPERATURE DISTRIBUTION
C *           PLOT DATA SET
C *  MAXVAL = MAXIMUM NUMBER OF R, TH, OR Z VALUES TO BE READ FROM THE
C *           TEMPERATURE DISTRIBUTION PLOT DATA SET
C *  MAXHOR = MAXIMUM NUMBER OF VALUES ON THE HORIZONTAL AXIS OF ANY
C *           PLOT + 2
C *  MAXVER = MAXIMUM NUMBER OF VALUES ON THE VERTICAL AXIS OF ANY PLOT
C *           + 2
C *  MAXPRF = MAXIMUM TOTAL NUMBER OF TEMPERATURE-DISTANCE PROFILES TO
C *           BE PLOTTED
C *  NPRVAL = MAXIMUM NUMBER OF R, TH, OR Z VALUES TO BE READ FROM THE
C *           DATA SET IF TEMPERATURE-DISTANCE PROFILES ARE GENERATED
C *  NPRTIM = MAXIMUM NUMBER OF TIMES FOR WHICH TEMPERATURE-DISTANCE
C *           PROFILES ARE GENERATED
C *
C ***************************************************************
C
      COMMON /NPOINT/ MAXNDS,MAXHOR,MAXVER,MAXVAL
      COMMON /NOPROF/ MAXPRF,NPRVAL,NPRTIM
      DATA MAXNDS /1000/
      DATA MAXHOR /52/
      DATA MAXVER /52/
      DATA MAXVAL /50/
      DATA MAXPRF /10/
      DATA NPRVAL /50/
      DATA NPRTIM /20/
      END
```

Fig. F13.B.1 Default BLOCK DATA routine in HEATPLOT-S

```
//jobname JOB (ccccc,IO5),'programmer name',TIME=(1,0)
/*JOBPARM LINES=5
//COMPILE EXEC FORTQC,REGION=nnnnK
//FORT.SYSIN DD *
(Insert FORTRAN Here)
/*
//STEP1 EXEC SCALE,REGION=nnnnK
//LKED.SYSIN DD *
 INCLUDE MODULES(HEATPLOT)
 INCLUDE OVLY(HEATPLOT)
 NAME HEATPLOT
/*
//GO.FT01F001 DD DSN=HEATING plot data set,DISP=SHR
//GO.SYSIN DD *
#HEATPLOT
 (Enter HEATPLOT-S Input Data Here)
END
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
// SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=XC92.POP
/*
//
```

Fig. F13.B.2 Format of JCL for executing HEATPLOT-S through SCALE at ORNL while overriding BLOCK DATA and generating plots on a CALCOMP plotter

F13.C APPENDIX C - DESCRIPTION OF TEMPERATURE DISTRIBUTION PLOT DATA SET WRITTEN BY HEATING5/HEATING6

The following seven statements are executed in HEATING5/HEATING6 in order to write data records that comprise the temperature distribution plot data set.

```
WRITE(IPLOT) JOBDES,NGEOM,NT,NBDTPT
WRITE(IPLOT) IT,(R(I),I=1,IT)
WRITE(IPLOT) JT,(TH(I),I=1,JT)
WRITE(IPLOT) KT,(Z(I),I=1,KT)
WRITE(IPLOT) (I,NTPI(I),NTPJ(I),NTPK(I),I=1,NT)
WRITE(IPLOT) NSET,NOIT,DELTAT,TIM
WRITE(IPLOT) (T1(I),I=1,NT),(TDUM(I),I=1,NBDTPT)
```

The first five records are written only once. Records 6 and 7 appear in pairs and are written each time the temperature distribution is output, including the initial temperature distribution for transient problems.

The following table contains a list of variables that appear in the statements, the type of each variable, the dimension of variables that are arrays, and a description of each variable. Variables that are not arrays are denoted by a dash (-) in the dimension column of the table.

Table F13.C.1. Variables that Appear in HEATING5/HEATING6 Plot Data Set Generation WRITE Statements

Variable	Type	Dimension	Description
IPLOT	INTEGER*4	-	Logical unit number of the plot data set
JOBDES	INTEGER*4	18	Descriptive title of problem
NGEDM	INTEGER*4	-	Type of geometry
NT	INTEGER*4	-	Number of nodes
NBDTPT	INTEGER*4	-	Number of boundary condition nodes
IT	INTEGER*4	-	Number of x or r fine grid lines
JT	INTEGER*4	-	Number of y or θ fine grid lines
KT	INTEGER*4	-	Number of z fine grid lines
R	REAL*8	IT	Coordinates of x or r fine grid lines
TH	REAL*8	JT	Coordinates of y or θ fine grid lines
Z	REAL*8	KT	Coordinates of z fine grid lines
NTPI	INTEGER*2	NT	X or r fine grid line indices for each node
NTPJ	INTEGER*2	NT	Y or θ fine grid line indices for each node
NTPK	INTEGER*2	NT	Z fine grid line indices for each node
NSET	INTEGER*4	-	Type of problem
NOIT	INTEGER*4	-	Number of iterations
DELTAT	REAL*8	-	change in problem time from the previous time
TIM	REAL*8	-	Current problem time
T1	REAL*8	NT	Temperatures at each node
TDUM	REAL*8	NBDTPT	Temperatures at each boundary condition node

NUREG/CR-0200
Volume 2, Section F14
ORNL/NUREG/CSD-2/V2/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at Oak Ridge Gaseous Diffusion Plant

REGPLOT6: A PLOTTING PROGRAM TO VERIFY HEATING6 INPUT DATA

C. K. Cobb

Manuscript Completed: September 1984
Date Published: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75

NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

CONTENTS

LIST OF FIGURES F14.v

ACKNOWLEDGMENTS F14.vii

ABSTRACT F14.ix

F14.1 INTRODUCTION AND CAPABILITIES F14.1.1

F14.2 LOGICAL PROGRAM FLOW AND DESCRIPTION OF SUBROUTINES F14.2.1

F14.2.1 DESCRIPTION OF SUBROUTINES IN REGPLOT6 ..... F14.2.1
F14.2.2 OAK RIDGE SYSTEMS ROUTINES USED BY REGPLOT6 ..... F14.2.2

F14.3 INPUT DESCRIPTION FOR REGPLOT6 ..... F14.3.1

F14.3.1 REGP DATA BLOCK KEYWORD DATA RECORD F14.3.1
F14.3.2 PLANE SPECIFICATION DATA RECORD F14.3.1
F14.3.3 PLOT DESCRIPTION DATA RECORD F14.3.1

F14.4 SAMPLE PROBLEM FOR REGPLOT6 ..... F14.4.1

F14.5 REFERENCES F14.5.1

F14.A APPENDIX A - CONTROL CARDS FOR EXECUTING REGPLOT6 ON THE OAK RIDGE IBM SYSTEMS F14.A.1

F14.B APPENDIX B - DESCRIPTION OF AN INTERACTIVE VERSION OF REGPLOT6 ..... F14.B.1

LIST OF FIGURES

F14.2.1 General flow chart of REGPLOT6 ..... F14.2.2

F14.4.1 Input data for the sample problem F14.4.2

F14.4.2 Output for the sample problem F14.4.3

F14.4.3 Plots generated for the sample problem F14.4.6

F14.A.1 Format of JCL for executing REGPLOT6 through SCALE at ORNL and generating plots on a CALCOMP plotter F14.A.2

F14.A.2 Format of JCL for executing REGPLOT6 through SCALE at ORGDP and generating plots on a CALCOMP plotter F14.A.2

F14.A.3 Format of JCL for executing REGPLOT6 through SCALE at ORNL and generating plots on an 11-inch Versatec plotter F14.A.3

F14.A.4 Format of JCL for executing REGPLOT6 through SCALE at ORNL and storing the plot file in the user's PDP-10 area F14.A.3

ACKNOWLEDGMENTS

This work was funded by the Transportation Branch, Nuclear Material Safety and Safeguards, U. S. Nuclear Regulatory Commission. The author expresses appreciation to C. V. Parks of Computing and Telecommunications Division (CTD), who served as project leader for this work, for review of the document and for providing general advice; and to L. M. Petrie of CTD for providing assistance and information concerning use of the SCALE System. Appreciation is also expressed to R. H. Odegaarden of the Nuclear Regulatory Commission, for review of the document; and to G. E. Giles of CTD for review of the document. Finally, the author expresses appreciation to A. I. Rose and B. R. Coppin of CTD for typing and preparation of the document, and to Lindy Norris of CTD for the layout of the text.

ABSTRACT

REGPLOT6 is a plotting program that enables the user of the HEATING6 generalized heat conduction code to graphically verify the accuracy of his HEATING6 input data. Maps of the regions, the materials, the heat generation function numbers, the initial temperature function numbers, and the boundary condition function numbers are generated by plotting the region boundaries and labeling each region appropriately. REGPLOT6 uses the DISSPLA (Display Integrated Software System and Plotting Language) graphics package, and generates a compressed plot data set that can be stored on disk or routed to various plotting devices.

F14.1 INTRODUCTION AND CAPABILITIES

This document has been developed to report the changes that have been made to REGPLOT[1] to make it compatible with HEATING6 (see Sect. F10) and the SCALE system. Much of the material in this document is taken from Ref. 1 and the memorandum "REGPLOT6 Program" to R. H. Odegaarden.[2] However, no other document is needed to use REGPLOT6.

REGPLOT6 is a plotting program that enables the user of the HEATING6 generalized heat conduction code to graphically verify the accuracy of his HEATING6 input data. REGPLOT6 may be used to plot the region boundaries of a one-, two-, or three-dimensional model. The region numbers, the material names, the heat generation function numbers, the initial temperature function numbers, and the boundary condition function numbers may be plotted for each region in any plane of the model. This version of REGPLOT6 may be used with HEATING6 input data sets that specify a maximum of 300 regions and 300 materials. REGPLOT6 uses the DISSPLA[3] graphics package. A compressed plot data set is generated that can be stored on disk or routed to various plotting devices.

REGPLOT6 uses the job description from the HEATING6 input data as the title of the plots. Only the first 59 characters of the job description are used. Corresponding Greek letters are written in the plot title if English letters are surrounded by question marks in the job description. For example, including the characters ?P? in the HEATING6 job description causes the character π to be written in the plot title. Therefore, extraneous question marks should not be included in the HEATING6 job description.

REGPLOT6 input data is specified by adding another data block to the HEATING6 input data (see Sect. F10.5.1). The data block must appear after all HEATING6 data blocks and before the percent (%) end of data indicator card. The first card in the data block must be a block indicator card containing the keyword REGP in the first four columns. Plots may be generated for more than one case in one job by concatenating HEATING6 input data sets that each contain a REGP data block.

The only data blocks read by REGPLOT6 are the REGIONS data block, the MATERIALS data block, and the REGP data block. The heat generation, initial temperature, and boundary condition function numbers for each region are read from the REGIONS data block. All other data blocks may be omitted when executing REGPLOT6. The title card, the two parameter cards, and the end of data card are also read by REGPLOT6 and must be included.

If the HEATING6 model is three-dimensional, the plane and the value of the axis normal to the plane for which plots are generated must be specified. The types of plots that are generated must be specified for each plane. Axis scaling factors and the region of the plane for which plots are generated may be input to REGPLOT6. A scaling factor is defined as the difference of the maximum and minimum axis value divided by the axis length (or units per inch). For example, if the x-axis is six inches long and begins at zero and the x-axis scaling factor is specified as 10.0, the maximum value of the x-axis is 60.0. If axis scaling factors are not specified or are specified as zero, the scaling factors are calculated by REGPLOT6. The largest scaling factor is used for all axes, which allows REGPLOT6 to draw the region boundaries in the proper proportions. If the region of the plane for which plots are generated is not specified, plots are drawn for the entire specified plane.

REGPLOT6 draws the region boundaries for each plot using six-inch axes. Plots are generated using a page size of 11 by 8.5 inches. For plots of the regions, the region numbers are written inside the proper region boundaries. For plots of the materials, the material names are written inside the proper region boundaries. For plots of the regions with heat generation and initial temperatures, the function numbers are written inside the proper region boundaries. Boundary conditions are depicted by symbols drawn along the borders of the proper regions. The boundary condition function numbers are written beside the symbols.

F14.2 LOGICAL PROGRAM FLOW AND DESCRIPTION OF SUBROUTINES

REGPLOT6 initially reads the input data for the first case. All plots for the first case are then generated. An attempt is made to read input data for another case. If another case does not exist, execution is halted. If another case does exist, all plots for that case are generated, and an attempt is made to read input data for yet another case. Execution is halted after the input data are read and plots are generated for all cases. The calling sequence of routines is depicted in Fig. F14.2.1.

F14.2.1 DESCRIPTION OF SUBROUTINES IN REGPLOT6

BLOCK DATA - Initializes variables that contain the logical unit numbers and the maximum number of regions and materials allowed.

MAIN - Directs ALOCAT (see Sect. M2.2.1) to call subroutine ALCRGP.

ALCRGP - Serves as an interface between the REGPLOT6 main routine and other subroutines. Determines the maximum dimension of the array that holds all variably dimensioned arrays.

BCDRAW - Draws the boundary condition symbols on the plots with boundary condition function numbers.

ECHO - Reads input data from unit 5, and lists card images on unit 6. Puts 'END' at the end of the appropriate card images and writes the resulting card images to unit 4.

ERRCK - Ensures that the material for each region has been specified in the MATERIALS data block. Writes an error message if any material has not been specified.

ERRMSG - Writes an error message if the number of regions or the number of materials specified in the input exceeds 300.

INITIL - Initializes variables and arrays to zero when a new case is encountered.

INPUT - Reads the title card and the two parameter cards and calls subroutines to read the remaining input data. Job information is also output.

MATERL - Reads input for one material from the MATERIALS data block.

NEXTCD - Reads the first word of the next card image. A message is written if end of file or end of data is encountered. Writes entire card image if it is a comment card, and reads the first word of the next card. If a data card is found, flags are set so that the next read is from the first word of the current card.

OUTPUT - Writes the parameters read from the two parameter cards read in subroutine INPUT.

READER - Reads for keywords and calls subroutines to read the corresponding data.

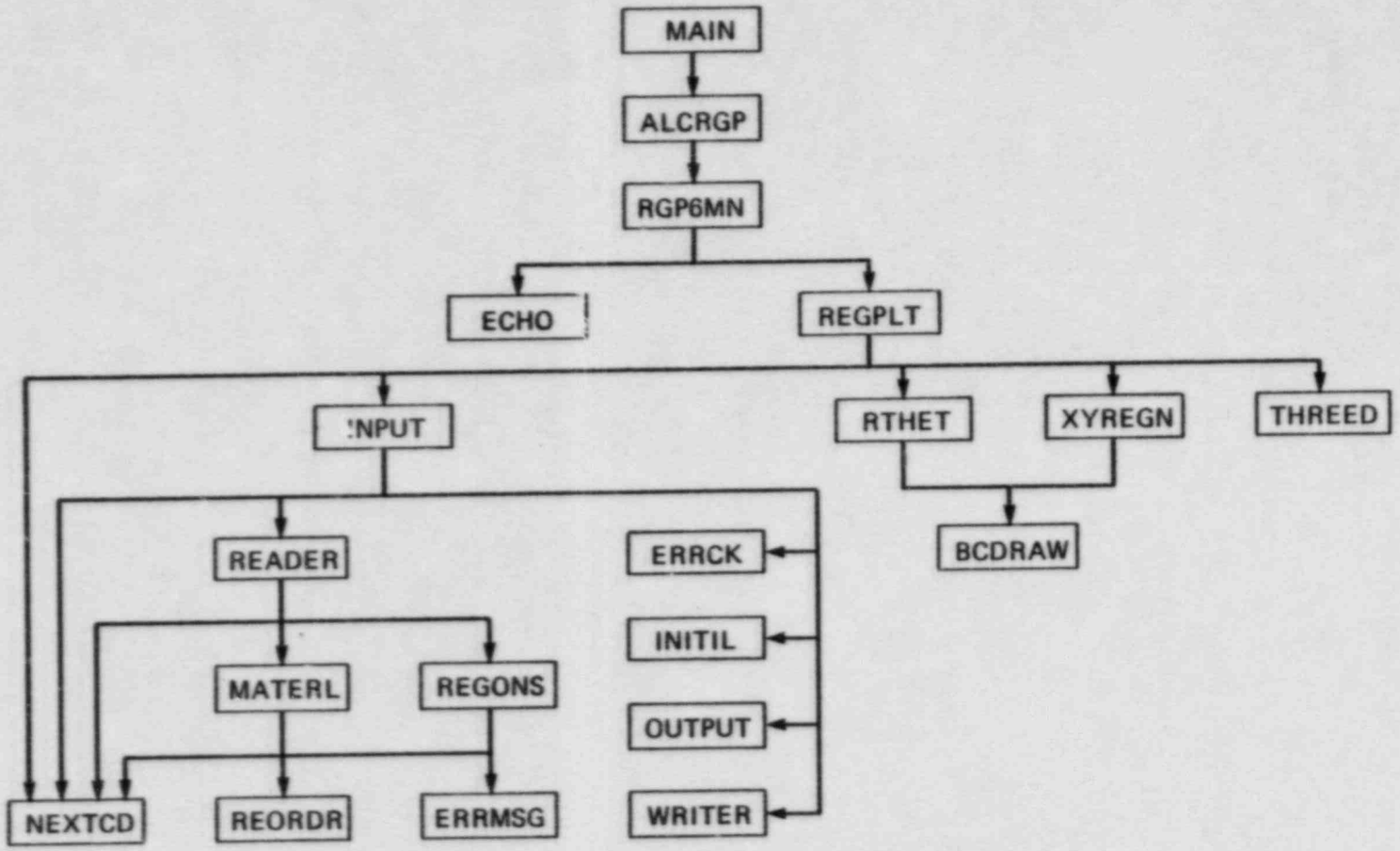
REGONS - Reads input for one region from the REGIONS data block.

REGPLT - Reads input from the REGP data block, draws the axes and writes the heading for each plot, and calls subroutines to complete each plot.

REORDR - Reorders the regions and materials if a region or a material is deleted.

RGP6MN - Initializes variables, performs variable dimensioning of arrays, and calls REGPLT to generate plots.

DWG. NO. K/G-84-3005
(U)



F14.2.2

Fig. F14.2.1 General flow chart of REGPLOT6

RTHET - Plots the region boundaries of an r-θ model in polar coordinates and writes the correct function number or material name inside each region boundary.

THREED - Stores the correct values in arrays depending on which plane is used for plotting for three-dimensional models.

WRITER - Writes the region data.

XYREGN - Plots the region boundaries of a Cartesian model in Cartesian coordinates and writes the correct function number or material name inside each region boundary.

F14.2.2 OAK RIDGE SYSTEMS ROUTINES USED BY REGPLOT6

REGPLOT6 calls Oak Ridge systems subroutines CORE, IDAY, JOBNUM, and TIME, and calls Oak Ridge systems functions ICOMPA and MODEL. These routines are all described in Table F10.4.2.

F14.3 INPUT DESCRIPTION FOR REGPLOT6

The input data for REGPLOT6 are read using the SCALE free-form reading routines (see Sect. M3) which allow data to be entered in an unformatted manner. REGPLOT6 input data are specified by adding another data block, called the REGP data block, to the HEATING6 input data. A description of the HEATING6 input data is found in Sect. F10.5. The REGP data block must appear after all other HEATING6 data blocks and before the percent (%) end of data indicator card. Plots may be generated for more than one case in one job by concatenating HEATING6 input data sets that each contain a REGP data block.

The only data blocks read by REGPLOT6 are the REGIONS data block, the MATERIALS data block, and the REGP data block. The heat generation, initial temperature, and boundary condition function numbers for each region are read from the REGIONS data block. Other data blocks may be either included or omitted when executing REGPLOT6. The title card, the two parameter cards, and the end of data card are also read by REGPLOT6 and must be included. In order to access REGPLOT6 via SCALE, a card image containing either =REGPLOT or #REGPLOT, beginning in the first column, must precede the card image input data. A card image containing END, beginning in the first column, must follow the card image input data (see Sect. F14.A). Examples of REGP data blocks are found in the REGPLOT6 input data in Fig. F14.4.1.

F14.3.1 REGP DATA BLOCK KEYWORD DATA RECORD

The characters 'REGP' must be entered in columns 1 through 4. This data record must be included and must be the first data record in the REGP data block.

F14.3.2 PLANE SPECIFICATION DATA RECORD

Omit this data record for one- and two-dimensional models. The first entry for this data record is alphanumeric and must be entered in columns 1 and 2. The second entry for this data record is a floating point number and must begin in column 4. This data record and corresponding PLOT DESCRIPTION data records may be repeated for any plane for which plots are desired for a three-dimensional model.

PLANE - The type of plane for which plots are generated. Valid entries are XY, RT, XZ, RZ, YZ, and TZ.

AXVAL - The value of the axis normal to the plane specified by PLANE.

F14.3.3 PLOT DESCRIPTION DATA RECORD

The first entry for this data record is an integer. All other entries for this data record are floating point numbers. This data record is repeated for every plot that is generated.

ITYPE - The number corresponding to the type of plot generated.

- = 1 A plot is generated with region numbers inside the region boundaries.
- = 2 A plot is generated with material names inside the region boundaries.
- = 3 A plot is generated with heat generation function numbers inside the region boundaries.
- = 4 A plot is generated with initial temperature function numbers inside the region boundaries.
- = 5 A plot is generated with boundary condition function numbers inside the region boundaries.

XSCALE - The scaling factor in units per inch for the horizontal axis of the plot. If XSCALE is not specified or is specified as zero, the scaling factor for the horizontal axis is computed by REGPLOT6.

YSCALE - The scaling factor in units per inch for the vertical axis of the plot. If YSCALE is not specified or is specified as zero, the scaling factor for the vertical axis is computed by REGPLOT6.

XREG1 - The minimum horizontal value of the area for which the plot is generated. If XREG1 is not specified or is specified as zero, the minimum horizontal value of the plane for which the plot is generated is used.

XREG2 - The maximum horizontal value of the area for which the plot is generated. If XREG2 is not specified or is specified as zero, the maximum horizontal value of the plane for which the plot is generated is used.

YREG1 - The minimum vertical value of the area for which the plot is generated. If YREG1 is not specified or is specified as zero, the minimum vertical value of the plane for which the plot is generated is used.

YREG2 - The maximum vertical value of the area for which the plot is generated. If YREG2 is not specified or is specified as zero, the maximum vertical value of the plane for which the plot is generated is used.

F14.4 SAMPLE PROBLEM FOR REGPLOT6

This section contains a sample problem with two cases to demonstrate the use of REGPLOT6. Input data for this problem are shown in Fig. F14.4.1. Output generated by REGPLOT6 is shown in Fig. F14.4.2. The plots generated for this problem are shown in Fig. F14.4.3.

The first case is a two-dimensional r-θ model. The only data blocks included in the input data for this case are the REGIONS, MATERIALS, and REGP data blocks. Plots are generated with region numbers, material names, heat generation function numbers, and boundary condition function numbers labeled in each region. Axis scaling factors, the horizontal minimum and maximum, and the vertical minimum and maximum are computed by REGPLOT6 for all plots. The first four plots shown in Fig. F14.4.3 are the plots generated for this case.

The second case is a three-dimensional x-y-z model. The only data blocks included in the input data for this case are the REGIONS, MATERIALS, and REGP data blocks. Plots are generated for two planes in the model. For the x-z plane at y=0.0, a plot is generated with region numbers labeled in each region. Axis scaling factors, the horizontal minimum and maximum, and the vertical minimum and maximum are computed by REGPLOT6. For the y-z plane at x=5.0, plots are generated with region numbers, material names, and initial temperature function numbers labeled in each region. For the plot of initial temperature function numbers, the axis scaling factors, the horizontal minimum and maximum, and the vertical minimum and maximum are specified. The final four plots shown in Fig. F14.4.3 are the plots generated for this case.

```
TWO DIMENSIONAL ?P? R-TH PROBLEM - STEADY STATE
10,2,1
8,3*0,9
REGIONS
     1     1    1.0    2.0     0.0  0.26180
     1           7      1
     2     2    1.0    2.0 0.26180  0.78540
     1           1      2
     3     1    1.0    2.0 0.78540  1.30900
     1           2      1
     4     3    1.0    2.0 1.30900  1.83260
     1           3      3
     5     1    1.0    2.0 1.83260  2.35620
     1           4      1
     6     2    1.0    2.0 2.35620  2.87980
     1           5      2
     7     1    1.0    2.0 2.87980   3.1416
     1           6      1
MATERIALS
     1 COPPER      4.0     1.4     0.06
     2 SILVER
     3 GOLD
REGPLOT6 DATA
1
2
3
5
%
3-D X-Y-Z PROBLEM
60,6,-1,0.0,5.0,0.0
5*0,2
REGIONS
1,1,0.0,2.0,0.0,3.0,0.0,4.0
1,5,1,0,6,0,8,0
2,2,2.0,3.0,0.0,3.0,0.0,4.0
1,6,0,0,6,0,8,0
3,3,3.0,5.0,2.0,3.0,0.0,4.0
1,7,9,0,0,0,2,0
4,4,3.0,5.0,0.0,2.0,0.0,2.0
2,8,0,0,5,0,8,0
5,5,3.0,5.0,1.0,2.0,2.0,4.0
2,9
6,1,3.0,5.0,0.0,1.0,2.0,4.0
1,3,0,0,1
MATERIALS
1,COPPER 0.93,8.94,0.091
2,ALUM 0.48,2.7,0.206
3,TIN 0.15,7.3,0.053
4,STEEL 0.11,7.85,0.118
5,ZINC 0.27,7.14,0.092
REGPLOT6 DATA
XZ
1
YZ 5.0
1
2
4,2*0.5,0.0,3.0,2.0,4.0
%
```

Fig. F14.4.1 Input data for the sample problem

```
CODE: REGPLOT6 07/10/84
DATE: 07-17-84
TIME: 12.19.41
JOBNAM: CLIRG6EX
COMPUTER: IBM3033/3033

JOB DESCRIPTION -- TWO DIMENSIONAL ?P? R-TH PROBLEM - STEADY STATE
 GEOMETRY TYPE NUMBER                          2 (OR RT )
 THIS IS A NEW PROBLEM.

DEFINITION OF INPUT/OUTPUT UNITS FOR CASE NUMBER   1

NAME    NO    UNIT NUMBER DEFINITION
IECHO    5    INITIAL INPUT DATA SET
IN       4    STANDARD INPUT DATA SET
IO       6    STANDARD OUTPUT DATA SET
          NUMBER OF REGIONS                    7
          NUMBER OF MATERIALS                  3
```

SUMMARY OF REGION DATA

********** DIMENSIONS **********

REGION NUMBER	LEFT-X OR INNER-R	RIGHT-X OR OUTER-R	LOWER-Y OR LEFT-THETA	UPPER-Y OR RIGHT-THETA	REAR-Z	FRONT-Z
1	1.0000	2.0000	0.0	0.2618	0.0	0.0
2	1.0000	2.0000	0.2618	0.7854	0.0	0.0
3	1.0000	2.0000	0.7854	1.3090	0.0	0.0
4	1.0000	2.0000	1.3090	1.8326	0.0	0.0
5	1.0000	2.0000	1.8326	2.3562	0.0	0.0
6	1.0000	2.0000	2.3562	2.8798	0.0	0.0
7	1.0000	2.0000	2.8798	3.1416	0.0	0.0

REGION NUMBER	MATERIAL NAME	INITIAL TEMP	HEAT GENERATION	BOUNDARY NUMBERS LF-X IN-R	RT-X OT-R	LO-Y LF-T	UP-Y RT-T	RR-Z	FT-Z
1	COPPER	1	7	1	0	0	0	0	0
2	SILVER	1	1	2	0	0	0	0	0
3	COPPER	1	2	1	0	0	0	0	0
4	GOLD	1	3	3	0	0	0	0	0
5	COPPER	1	4	1	0	0	0	0	0
6	SILVER	1	5	2	0	0	0	0	0
7	COPPER	1	6	1	0	0	0	0	0

```
***** CASE NUMBER   1
***** JOB DESCRIPTION *****TWO DIMENSIONAL ?P? R-TH PROBLEM - STEADY STATE
***** THE INPUT DATA FOR THIS CASE HAVE BEEN READ AND PROCESSED
***** NUMBER OF ERRORS ENCOUNTERED IN THE INPUT DATA FOR THIS CASE WAS -  0
```

Fig. F14.4.2 Output for the sample problem

```
PLOT  1 WILL BE DRAWN OF THE REGION .
  THE HORIZONTAL SCALE VALUE IS 0.0
  THE VERTICAL SCALE VALUE IS 0.0
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
          THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.


PLOT  2 WILL BE DRAWN OF THE MATERIAL.
  THE HORIZONTAL SCALE VALUE IS 0.0
  THE VERTICAL SCALE VALUE IS 0.0
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
          THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.


PLOT  3 WILL BE DRAWN OF THE HEAT GEN.
  THE HORIZONTAL SCALE VALUE IS 0.0
  THE VERTICAL SCALE VALUE IS 0.0
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
          THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.


PLOT  4 WILL BE DRAWN OF THE B COND .
  THE HORIZONTAL SCALE VALUE IS 0.0
  THE VERTICAL SCALE VALUE IS 0.0
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  0.0       0.0     , RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
          THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.
```

Fig. F14.4.2 (continued)

CODE: REGPLOT6 07/10/84
DATE: 07-17-84
TIME: 12.19.41
JOBNAM: CLIRG6EX
COMPUTER: IBM3033/3033

JOB DESCRIPTION -- 3-D X-Y-Z PROBLEM
GEOMETRY TYPE NUMBER 6 (OR XYZ)
THIS IS A NEW PROBLEM.

DEFINITION OF INPUT/OUTPUT UNITS FOR CASE NUMBER 2

NAME	NO	UNIT NUMBER DEFINITION
IECHO	5	INITIAL INPUT DATA SET
IN	4	STANDARD INPUT DATA SET
IO	6	STANDARD OUTPUT DATA SET

NUMBER OF REGIONS 6
NUMBER OF MATERIALS 5

SUMMARY OF REGION DATA

********** DIMENSIONS **********

REGION NUMBER	LEFT-X OR INNER-R	RIGHT-X OR OUTER-R	LOWER-Y OR LEFT-THETA	UPPER-Y OR RIGHT-THETA	REAR-Z	FRONT-Z
1	0.0	2.0000	0.0	3.0000	0.0	4.0000
2	2.0000	3.0000	0.0	3.0000	0.0	4.0000
3	3.0000	5.0000	2.0000	3.0000	0.0	4.0000
4	3.0000	5.0000	0.0	2.0000	0.0	2.0000
5	3.0000	5.0000	1.0000	2.0000	2.0000	4.0000
6	3.0000	5.0000	0.0	1.0000	2.0000	4.0000

REGION NUMBER	MATERIAL NAME	INITIAL TEMP	HEAT GENERATION	BOUNDARY NUMBERS LF-X IN-R	RT-X OT-R	LO-Y LF-T	UP-Y RT-T	RR-Z	FT-Z
1	COPPER	1	5	1	0	6	0	8	0
2	ALUM	1	6	0	0	6	0	8	0
3	TIN	1	7	9	0	0	0	2	0
4	STEEL	2	8	0	0	5	0	8	0
5	ZINC	2	9	0	0	0	0	0	0
6	COPPER	1	3	0	0	1	0	0	0

***** CASE NUMBER 2
***** JOB DESCRIPTION *****3-D X-Y-Z PROBLEM
***** THE INPUT DATA FOR THIS CASE HAVE BEEN READ AND PROCESSED
***** NUMBER OF ERRORS ENCOUNTERED IN THE INPUT DATA FOR THIS CASE WAS -- 0

Fig. F14.4.2 (continued)

```
THE XZ PLANE WILL BE PLOTTED AT  Y =  0.0


PLOT  1 WILL BE DRAWN OF THE REGION .
  THE HORIZONTAL SCALE VALUE IS 0.0
  THE VERTICAL SCALE VALUE IS 0.0
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0        0.0       , RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  0.0        0.0       , RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
           THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.



THE YZ PLANE WILL BE PLOTTED AT  X =  5.00000E+00


PLOT  2 WILL BE DRAWN OF THE REGION .
  THE HORIZONTAL SCALE VALUE IS 0.0
  THE VERTICAL SCALE VALUE IS 0.0
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0        0.0       , RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  0.0        0.0       , RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
           THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.



PLOT  3 WILL BE DRAWN OF THE MATERIAL.
  THE HORIZONTAL SCALE VALUE IS 0.0
  THE VERTICAL SCALE VALUE IS 0.0
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0        0.0       , RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  0.0        0.0       , RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
           THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.



PLOT  4 WILL BE DRAWN OF THE I TEMP .
  THE HORIZONTAL SCALE VALUE IS 5.00000E-01
  THE VERTICAL SCALE VALUE IS 5.00000E-01
    *** NOTE - SCALE VALUES WHICH ARE ZERO WILL BE COMPUTED BY THE PROGRAM.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE HORIZONTAL AXIS ARE  0.0        3.00000E+00, RESPECTIVELY.
  THE MINIMUM AND THE MAXIMUM VALUES ON THE VERTICAL AXIS ARE  2.00000E+00 4.00000E+00, RESPECTIVELY.
    *** NOTE - IF THE AXIAL MINIMUM OR MAXIMUM IS ZERO,
           THE MINIMUM AND/OR THE MAXIMUM OF THE MODEL WILL BE USED.
```

Fig. F14.4.2 (continued)

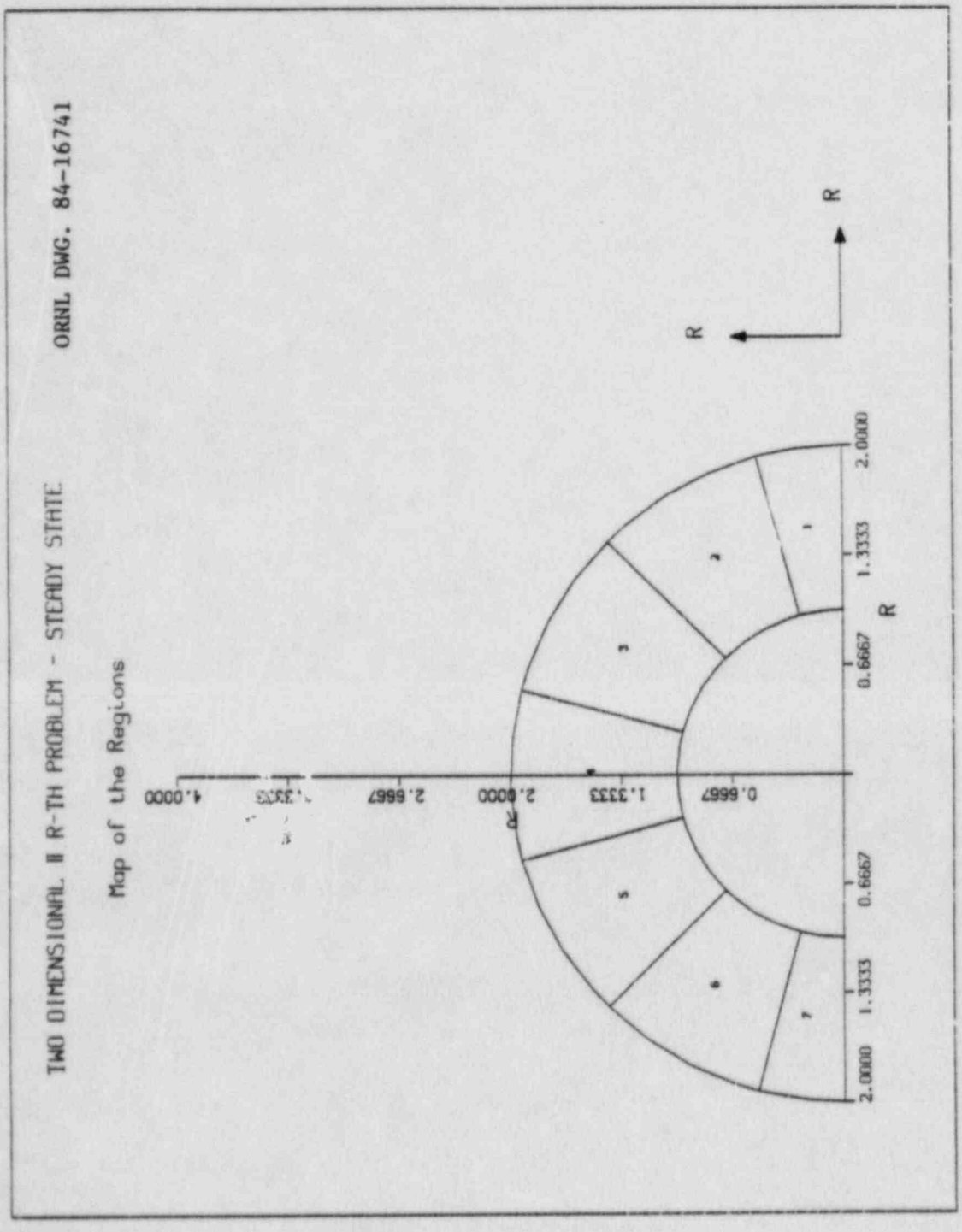


Fig. F14.4.3 Plots generated for the sample problem

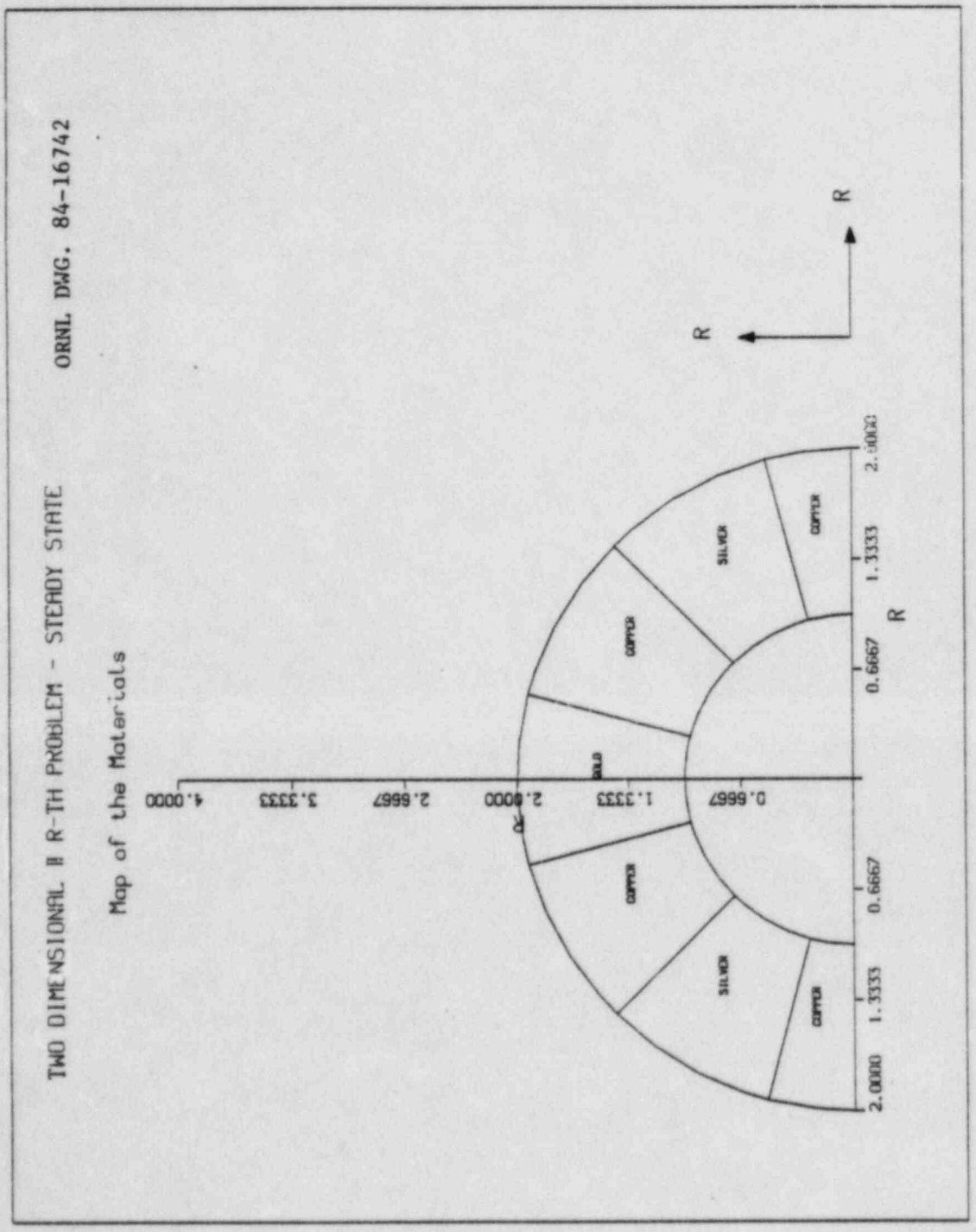


Fig. F14.4.3 (continued)

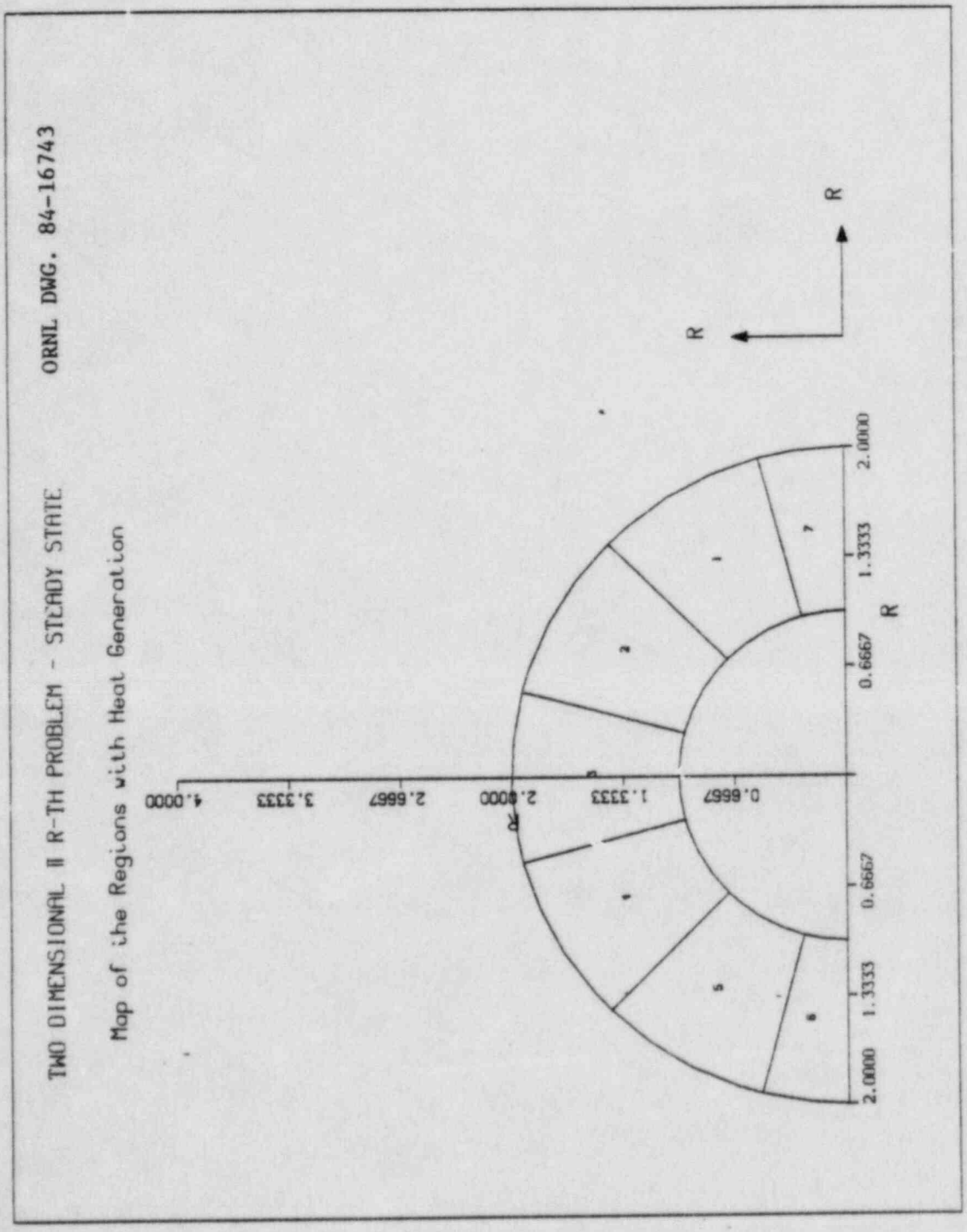


Fig. F14.4.3 (continued)

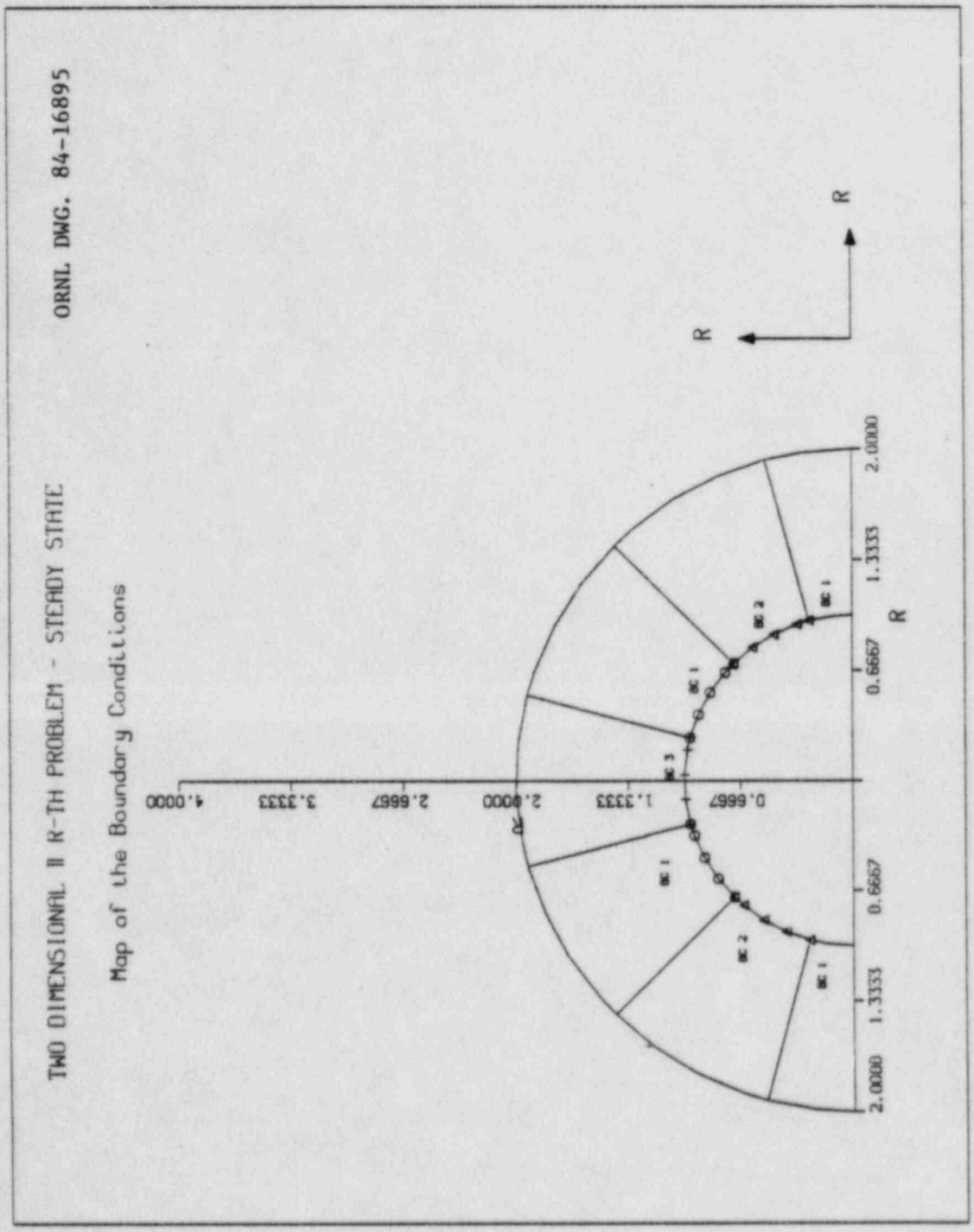


Fig. F14.4.3 (continued)

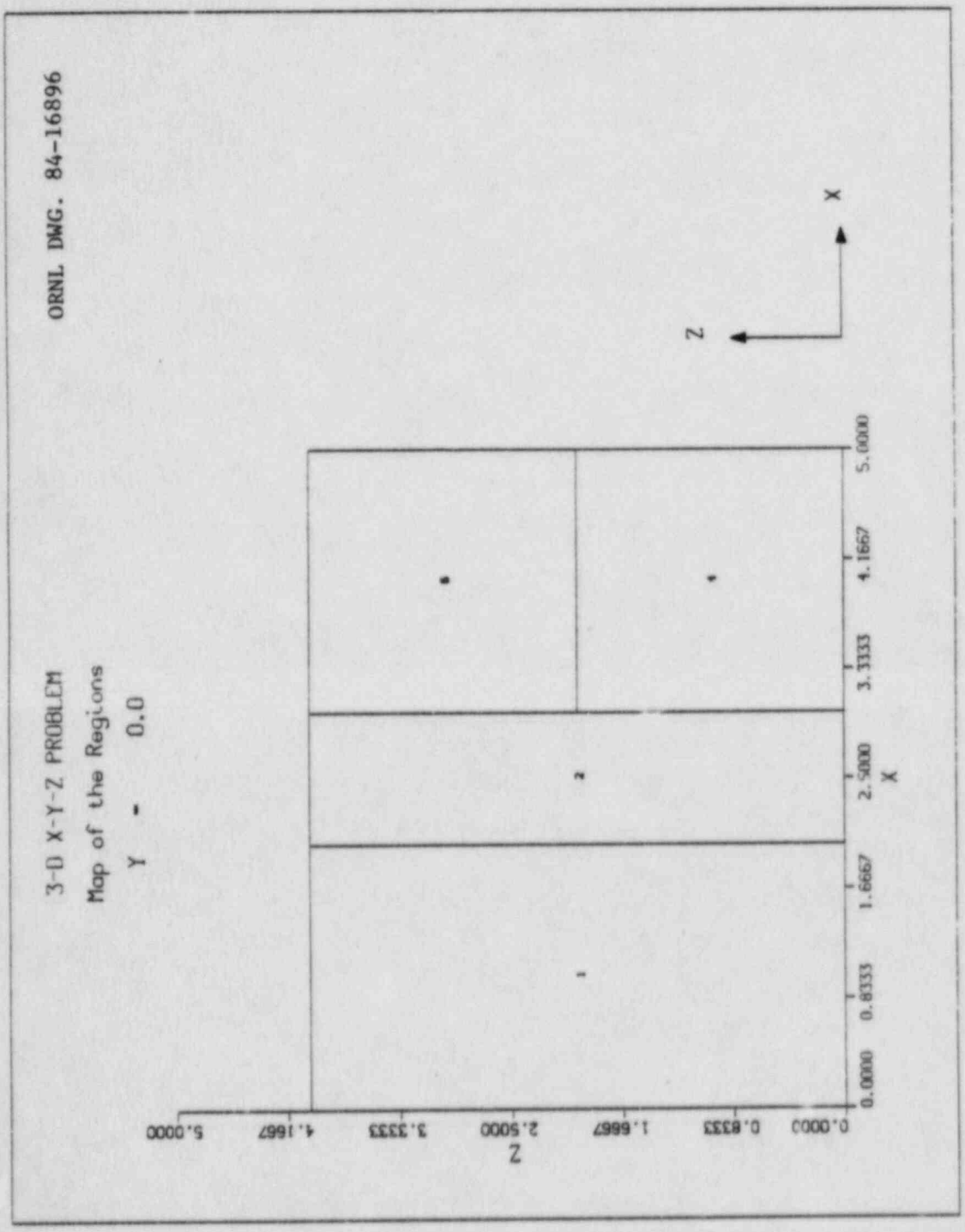


Fig. F14.4.3 (continued)

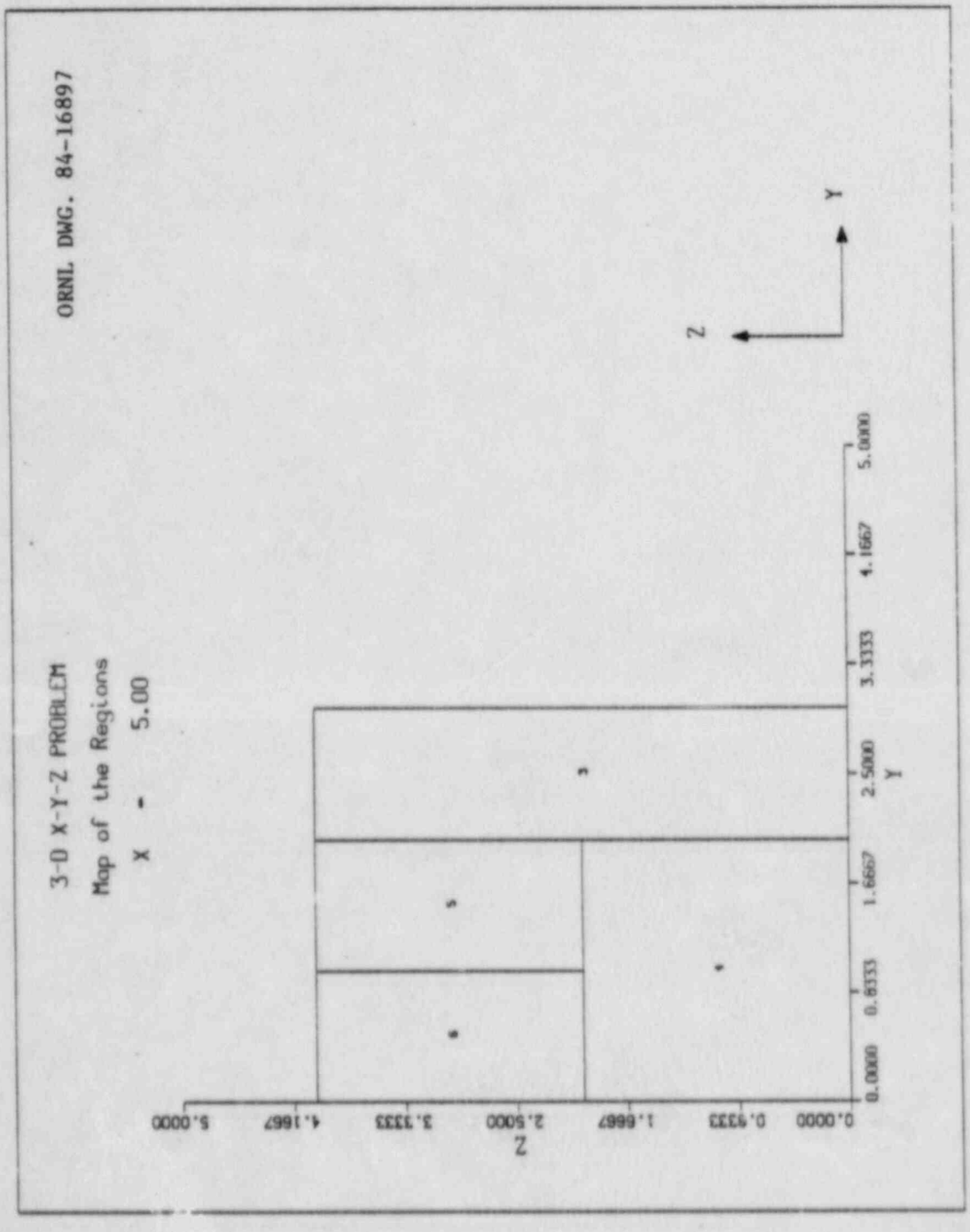


Fig. F14.4.3 (continued)

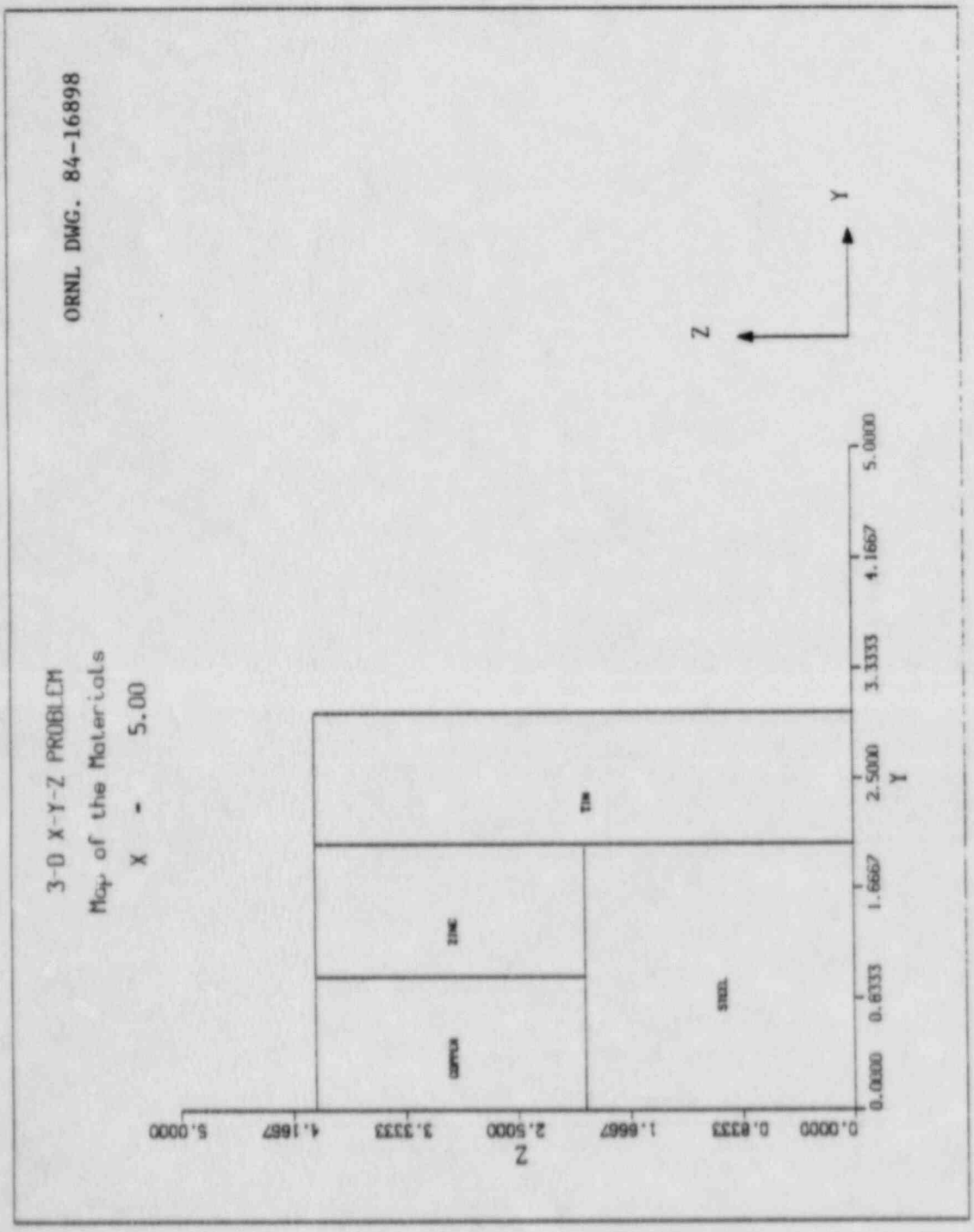


Fig. F14.4.3 (continued)

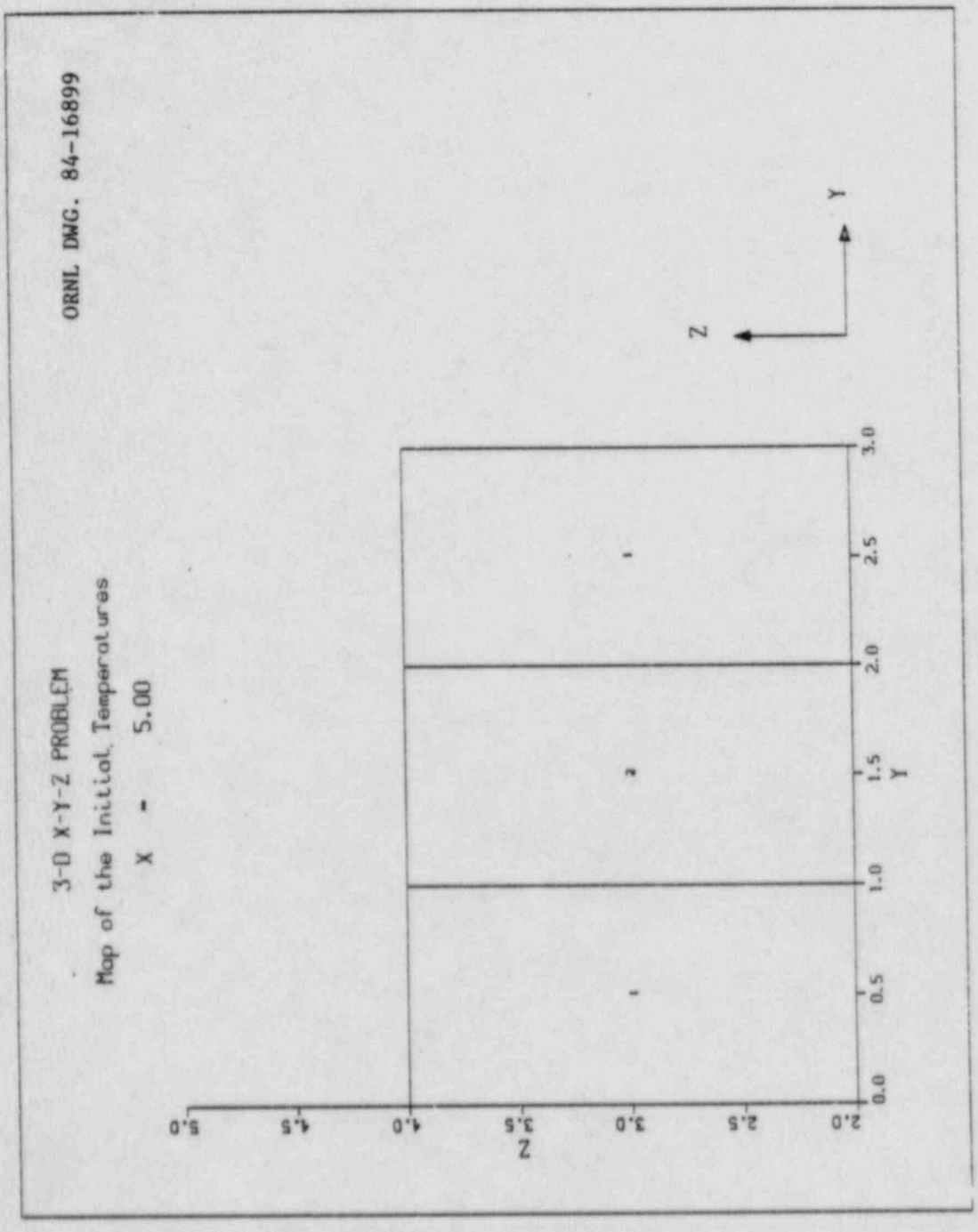


Fig. F14.4.3 (continued)

F14.5 REFERENCES

1. D. C. Elrod, "REGPLOT - A Plotting Program to Graphically Check HEATING5 Input Data," K/CSD/TM-12, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, August 1977.

2. C. A. Sady, External Memorandum to R. H. Odegaarden, May 22, 1980.

3. "DISSPLA User's Manual," Integrated Software Systems Corporation, San Diego, California, 1981.

4. "CSD Programmer's Notebook," Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, March 10, 1981.

5. "CS Graphics Manual," Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, June 1, 1982.

F14.A APPENDIX A - CONTROL CARDS FOR EXECUTING REGPLOT6 ON THE OAK RIDGE IBM SYSTEMS

The REGPLOT6 module is part of the SCALE system at the computing centers at both Oak Ridge National Laboratory (ORNL) and Oak Ridge Gaseous Diffusion Plant (ORGDP). This appendix discusses the job control language (JCL) which is required to execute this version of REGPLOT6. Examples of JCL are presented which show how to execute REGPLOT6 and generate plots on CALCOMP plotters at ORNL and ORGDP and on Versatec plotters at ORNL. An example of JCL which shows how to generate a compressed plot file and route the file to the user's PDP-10 area is also presented. For additional information concerning JCL and plotting devices, the user is referred to the CSD Programmer's Notebook[4] and the CS Graphics Manual.[5]

The control cards necessary to generate plots on a CALCOMP plotter using this version of REGPLOT6 at ORNL and ORGDP are shown in Fig. F14.A.1 and Fig. F14.A.2, respectively. The jobname field on the JOB statement has a format of uidxxxxx, where uid is the three-letter user identification assigned to the user by CTD, and the characters xxxxx may be up to five characters which make the job name unique. The charge field, ccccc, contains a four- or five-digit CTD charge number. The programmer name field contains information such as the user's name and address which is necessary to route the job output back to the user. This field may contain up to twenty characters.

The control cards necessary to generate plots on an 11-inch Versatec plotter at ORNL using this version of REGPLOT6 are shown in Fig. F14.A.3. These control cards are identical to those shown in Fig. F14.A.1 with the exception of the plot spooler request (REQ=) parameters. Omitting the parameter PAPER=600 will cause plots to be generated on a 36-inch Versatec plotter at ORNL.

The control cards necessary to generate a compressed plot file and route the file to the user's PDP-10 area are shown in Fig. F14.A.4. The field ppppp,ppp should be replaced with the programmer number assigned to the user by CTD. The file is named uidxxx.PCH, where the characters uidxxx are the first six characters of the job name. The plot file can be sent from the user's PDP-10 area to various plotting devices on which the plots are generated.

```
//jobname JOB (ccccc,IO5),'programmer name',TIME=(1,0)
/*JOBPARM LINES=5
//STEP1 EXEC SCALE,REGION=500K
//GO.FT04F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(1,1)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//GO.SYSIN DD *
#REGPLOT
 (Enter REGPLOT6 Input Data Here)
END
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
//  SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=XC92.POP
/*
//
```

Fig. F14.A.1 Format of JCL for executing REGPLOT6 through SCALE at ORNL and generating plots on a CALCOMP plotter

```
//jobname JOB (ccccc),'programmer name'
//*CLASS CPU95=1M,IO=5,L=5,REGION=500K
//STEP1 EXEC SCALE,REGION=500K
//GO.FT04F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(1,1)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//GO.SYSIN DD *
#REGPLOT
 (Enter REGPLOT6 Input Data Here)
END
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
//  SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=KC92.POP
/*
//
```

Fig. F14.A.2 Format of JCL for executing REGPLOT6 through SCALE at ORGDP and generating plots on a CALCOMP plotter

```
//jobname JOB (ccccc,IO5),'programmer name',TIME=(1,0)
/*JOBPARM LINES=5
//STEP1 EXEC SCALE,REGION=500K
//GO.FT04F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(1,1)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//GO.SYSIN DD *
#REGPLOT
 (Enter REGPLOT6 Input Data Here)
END
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
//  SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=XV36.POP,PAPER=600
/*
//
```

Fig. F14.A.3 Format of JCL for executing REGPLOT6 through SCALE at ORNL and generating plots on an 11-inch Versatec plotter

```
//jobname JOB (ccccc,IO5),'SAVEppppp,pppH ',TIME=(1,0)
/*JOBPARM LINES=5,CARDS=10000
/*ROUTE PUNCH RMT45
//STEP1 EXEC SCALE,REGION=500K
//GO.FT04F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(1,1)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//GO.SYSIN DD *
#REGPLOT
 (Enter REGPLOT6 Input Data Here)
END
/*
//GO.COMPOUT DD DISP=(NEW,PASS),DSN=&&REGPL6,UNIT=SPDA,
//  SPACE=(TRK,(5,1),RLSE)
//STEP2 EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1 DD DSN=&&REGPL6,DISP=(OLD,DELETE),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=4
//SYSUT2 DD SYSOUT=B,DCB=(RECFM=FB,L      =80,LRECL=80)
//
```

Fig. F14.A.4 Format of JCL for executing REGPLOT6 through SCALE at ORNL and storing the plot file in the user's PDP-10 area

F14.B APPENDIX B - DESCRIPTION OF AN INTERACTIVE VERSION OF REGPLOT6

A version of REGPLOT6 may be executed in an interactive manner on the PDP-10 computer at ORNL. Plots are generated on a Tektronix graphics terminal.

The user must recall the file RG6SCL.EXE from DECtape D3361 (use the command FILE R,D3361,RG6SCL.EXE). When the file RG6SCL.EXE is in the user's PDP-10 area, the interactive version of REGPLOT6 is executed by the command RUN RG6SCL. A file named FOR04.DAT is created by the interactive version of REGPLOT6. This file will replace a previously existing file named FOR04.DAT. Therefore, the user should not have a file named FOR04.DAT in his PDP-10 area when interactively executing REGPLOT6.

The first prompt asks the user to enter the name of the file that contains the HEATING6 input data for which plots are generated. The REGP data block should not be included with the HEATING6 input data for the interactive version of REGPLOT6. The region input is summarized on the terminal screen, and the user is then prompted for the input given in the REGP data block in the version of REGPLOT6 in SCALE. After each plot is generated, the user is given a choice of generating another plot or stopping executing of the program.

NUREG/CR-0200
VOLUME 3
ORNL/NUREG/CSD-2/V3/R2

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

Volume 3: Miscellaneous

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
Under Interagency Agreement DOE 40-549-75

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

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

NUREG/CR-0200
Volume 3, Section M12
ORNL/NUREG/CSD-2/V3/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

CESAR: A Criticality Experiment Storage and Retrieval Program

J. R. Knight
R. M. Westfall

Manuscript Completed: February 1984

Date Published: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

CONTENTS

LIST OF TABLES M12.v

ACKNOWLEDGEMENTS M12.vii

ABSTRACT M12.ix

M12.1 INTRODUCTION M12.1.1

M12.2 PROGRAM AND LIBRARY DESCRIPTION M12.2.1

M12.3 PROGRAM FLOW M12.3.1

M12.4 INPUT DESCRIPTION M12.4.1

M12.5 SAMPLE PROBLEM M12.5.1

M12.6 REFERENCES M12.6.1

LIST OF TABLES

Table M12.2.1 CESAR problem ID's listed in ascending order with the associated citation ID M12.2.2

Table M12.2.2 CESAR citation ID's listed in ascending order with the associated problem ID M12.2.4

ACKNOWLEDGEMENTS

The authors would like to express appreciation for the work of L. M. Petrie who put the CESAR program and library into LIBRARIAN files and made the program operable on the SCALE system. We also would like to thank C. V. Parks for his many suggestions on revisions in the CESAR program and library.

ABSTRACT

CESAR is a program for storing and retrieving input data for criticality calculations using the SCALE system. Each problem consists of a problem description and an input dataset to run the problem. Editing and updating features are described. As of January 1984, 447 experiments were included in the CESAR library.

M12.1 INTRODUCTION

The critical experiment storage and retrieval program (CESAR) was developed to provide a means of storing input data for various cross-section preparation and criticality modules in the SCALE system. These input datasets are based on documented critical experiments performed at national laboratories and other installations. An effort has been made to assure that each experiment modeled has input data compatible with the programs and cross sections available in the SCALE system.

The CESAR library contains problem sets that use cross-section processing routines, transport calculations, and Monte Carlo calculations. In many instances, SCALE procedures such as CSAS1 (see Sect. C1) or CSAS25 (see Sect. C4) are called for. Each problem is based on a documented critical experiment. The problem sets consist of two parts: a brief description of the experiment, and the data needed to execute the problem. Each problem is assigned a problem ID and a citation ID. The problem ID is assigned based on the fundamental characteristics of the experiment while the citation ID is taken from the bibliography[1] developed for the Nuclear Criticality Information System (NCIS) at Lawrence Livermore National Laboratory and identifies the reference that describes the experiment. Thus, a user can search and retrieve datasets for various experiment types or for a particular reference.

M12.2 PROGRAM AND LIBRARY DESCRIPTION

The number of problems that can be stored by CESAR is limited only by the space available on the storage device used. Procedures for adding to or correcting the data are included (see Sect. M12.4). Data may be retrieved in the form of printed output, punched cards, or both. Edits are available for the entire library, for an individual problem ID or problem ID list, or for a given problem type. Problem-type categories are by maximum and minimum enrichment, by homogeneous or heterogeneous system geometry, and by fissile nuclide in the fuel. Edits may be made on one or more of these types of specifications. Edits are also available by citation ID. The output from an edit may consist of the experiment description, the input data list, or both.

The problem ID is a 7-digit integer. The first three digits specify the fuel type: 000-500 for ^{235}U, 600-899 for ^{239}Pu or $^{239}Pu/^{235}U$ mixtures, and 900-999 for ^{233}U. The fourth digit is 0 for homogeneous systems and 1 for heterogeneous systems. The last three digits are the fuel enrichment to the nearest tenth of a percent.

Example: 1071043 ^{235}U fuel, heterogeneous, 4.3% enrichment

The citation ID refers to the reference from which the experiment was taken and is identical to the reference ID used in the NCIS bibliography.[1] This means that all problems taken from a single reference have the same citation ID. In some cases, individual problems within a given reference are assigned arbitrary experiment numbers, which are for internal use at Oak Ridge National Laboratory. A citation ID cross reference is provided so that if the input ID sought is not found, any available alternate citation of the same experiment will be found.

As of January 1984, the CESAR library contains 432 experiments. Additional experiments may be added at a later date. The experiments are located in the library in order of increasing problem ID's. Table M12.2.1 is a list of all the problem ID's and the citation ID's associated with each experiment. Table M12.2.2 gives a list of citation ID's in ascending order, along with the associated problem ID's.

It should be emphasized that the problem input data in CESAR were formulated to run with the SCALE driver (Sect. M1) which requires a module activation record (an = or # in the first column followed by the module name) as the first data entry and a final data record with an **END** in columns 1-3. To run these datasets under AMPX[2] or another system, one or both of these cards may have to be removed (for AMPX only the **END** record needs to be removed). Also, it should be noted that some problem sets contain two sets of KENO input data for the same experimental model. The difference is in the cross-section sets used. At least one of the cross-section sets will be available with SCALE. However, several problems also contain KENO input data which uses the 16-group cross sections provided with KENO IV and described in ORNL-4938.[3]

Table M12.2.1 CESAR problem ID's listed in ascending order with the associated citation ID

Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID
11024	2125	591932	206	1081043	2112	1321024	2094	6020752	2110
21024	2125	601932	206	1081048	2129	1321043	2112	6020758	2110
31024	2125	611932	1358	1091043	2112	1331024	2094	6020770	1548
41024	2125	621932	1413	1091048	2129	1361024	2094	6020859	2110
51024	2125	641926	206	1101043	2112	1371024	2094	6020885	1453
61024	2125	651926	206	1101048	2129	1381024	2094	6020920	1677
91024	2125	661926	206	1111043	2112	1391024	2094	6020937	1588
111024	2125	671930	2121	1111048	2129	1411024	2094	6020942	1588
121024	2125	681932	2095	1121024	2094	1431024	2094	6020944	1696
141043	2125	691932	2095	1121043	2112	1441024	2094	6020951	2110
161043	2125	700020	250	1121048	2129	1461024	2094	6020953	1502
171043	2125	710030	250	1131043	2112	1471024	2094	6021920	1547
191043	2125	720020	250	1131048	2129	1481024	2094	6021944	1960
211043	2125	730020	250	1141043	2112	1501024	2094	6030752	2110
221043	2125	740030	250	1141048	2129	1521024	2094	6030758	2110
231043	2125	750045	2124	1151024	2094	1541024	2094	6030770	1548
240977	2086	781025	2126	1151043	2112	6000414	2110	6030859	2110
250932	164	791025	2126	1151048	2129	6000752	2110	6030885	1453
260932	1092	801025	2126	1161024	2094	6000758	2110	6030920	1677
270932	1092	811025	2126	1161043	2112	6000770	1548	6030937	1588
280050	2131	821025	2126	1161048	2129	6000859	2110	6030942	1588
290932	1199	831025	2126	1171043	2112	6000862	2110	6030944	1696
300932	2119	841025	2126	1171048	2129	6000885	1453	6030951	2110
310932	2119	851025	2126	1181043	2112	6000907	2110	6031920	1547
321024	2122	861025	2126	1181048	2129	6000920	1677	6031944	1960
331024	2122	871025	2126	1191043	2112	6000937	1588	6040752	2110
341024	2122	881025	2126	1191048	2129	6000942	1588	6040758	2110
351024	2122	891025	2126	1201024	2094	6000944	1696	6040770	1548
361024	2122	901025	2126	1201043	2112	6000951	2110	6040859	2110
371024	2122	911025	2126	1201048	2129	6000953	1502	6040885	1453
381027	2120	921025	2126	1211043	2112	6000977	2110	6040920	1677
390932	2121	931025	2126	1211048	2129	6000995	2110	6040937	1588
400932	2121	941025	2126	1221043	2112	6000999	2110	6040942	1588
410938	324	951025	2126	1221048	2129	6001885	2116	6040944	1696
420932	2004	961025	2126	1231043	2112	6001920	1547	6040951	2110
430932	164	971025	2126	1241024	2094	6001944	1960	6041920	1547
440932	2119	981025	2126	1241043	2112	6010752	2110	6041944	1960
450935	1	991048	2129	1251011	2130	6010758	2110	6050752	2110
460932	838	1001048	2129	1251043	2112	6010770	1548	6050758	2110
460940	1	1011048	2129	1261024	2094	6010859	2110	6050770	1548
470931	653	1021048	2129	1261043	2112	6010885	1453	6050859	2110
480931	653	1031048	2129	1271024	2094	6010920	1677	6050885	1453
490977	2086	1041048	2129	1271043	2112	6010937	1588	6050920	1677
500932	2095	1051024	2094	1281024	2094	6010942	1588	6050937	1588
511932	801	1051043	2112	1281043	2112	6010944	1696	6050942	1588
521932	801	1051048	2129	1291024	2094	6010951	2110	6050944	1696
551932	110	1061043	2112	1291043	2112	6010953	1502	6050951	2110
561932	110	1061048	2129	1301043	2112	6010977	2110	6051835	1960
571932	110	1071043	2112	1311024	2094	6011920	1547	6051920	1547
581932	110	1071048	2129	1311043	2112	6011944	1960	6060752	2110

Table M12.2.1 (continued)

Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID
6060770	1548	6100770	1548	6160944	1502	6230944	1502	6341885	2117
6060839	2110	6100885	1453	6160951	2110	6230954	525	6351885	2117
6060885	1453	6100920	1677	6161885	1547	6231885	2118	6361885	2117
6060920	1677	6100944	1696	6161920	1547	6231920	1547	6371885	2117
6060937	1588	6101885	1547	6170770	1548	6240920	1677	6381885	2117
6060942	1588	6101920	2122	6170920	1677	6240944	1502	6391885	2117
6060944	1696	6110770	1548	6170944	1502	6240954	525	6401885	2117
6060951	2110	6110885	1453	6170951	2110	6241885	2118	6411885	2117
6061835	1960	6110920	1677	6171885	1547	6241920	1547	6421885	2117
6061920	1547	6110944	1696	6171920	1547	6250920	1677	6431885	2117
6070752	2110	6111885	1547	6180770	1548	6250944	1696	6441885	2117
6070770	1548	6111920	2122	6180862	2110	6250954	525	6451885	2117
6070885	1453	6120770	1548	6180920	1677	6251885	2118	6461885	2117
6070920	1677	6120885	1453	6180944	1502	6251920	1547	6471885	2117
6070937	1588	6120920	1677	6181885	1547	6260920	1677	6481885	2117
6070942	1588	6120944	1696	6181920	1547	6260944	1696	6491885	2117
6070944	1696	6121885	1547	6190907	2110	6260954	525	6501885	2117
6070951	2110	6121920	2122	6190920	1677	6261885	2118	6511885	2117
6071835	1960	6130770	1548	6190944	1502	6261920	1547	6521885	2117
6071920	1547	6130885	1453	6191885	2116	6270944	1696	6531885	2117
6080752	2110	6130920	1677	6191920	1547	6270951	1538	6541885	2117
6080770	1548	6130944	1696	6200414	2110	6271885	2117	6551885	2117
6080885	1453	6131885	1547	6200920	1677	6271920	1547	6561885	2117
6080920	1677	6131920	2122	6200944	1502	6280944	1502	6571885	2117
6080937	1588	6140770	1548	6201885	2116	6280951	1538	6581885	2117
6080942	1588	6140885	1453	6201920	1547	6281885	2117	6591885	2117
6080944	1696	6140920	1677	6210920	1677	6290944	1502	6601885	2117
6081835	1960	6140944	1502	6210944	1502	6291885	2117	6611885	2117
6081920	1547	6141885	1547	6211885	2116	6300944	1696	6621885	2117
6090752	2110	6141920	2122	6211920	1547	6301885	2117	6631885	2117
6090770	1548	6150770	1548	6220920	1677	6310944	1696	6641885	2117
6090885	1453	6150920	1677	6220944	1502	6311885	2117	6651885	2117
6090920	1677	6150944	1502	6220951	2110	6320944	1696	6661885	2117
6090942	1588	6151885	1547	6220974	2123	6321885	2117	9000977	2119
6090944	1696	6151920	2122	6221885	2118	6330944	1696		
6091835	1960	6160770	1548	6221920	1547	6331885	2117		
6091920	1547	6160920	1677	6230920	1677	6340944	1696		

Table M12.2.2 CESAR citation ID's listed in ascending order with the associated problem ID

Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID
1	450935	1502	6020953	1548	6030770	1677	6150920	2094	1161024
1	460940	1502	6140944	1548	6040770	1677	6160920	2094	1201024
110	551932	1502	6150944	1548	6050770	1677	6170920	2094	1241024
110	561932	1502	6160944	1548	6060770	1677	6180920	2094	1261024
110	571932	1502	6170944	1548	6070770	1677	6190920	2094	1271024
110	581932	1502	6180944	1548	6080770	1677	6200920	2094	1281024
164	250932	1502	6190944	1548	6090770	1677	6210920	2094	1291024
164	430932	1502	6200944	1548	6100770	1677	6220920	2094	1311024
206	591932	1502	6210944	1548	6110770	1677	6230920	2094	1321024
206	601932	1502	6220944	1548	6120770	1677	6240920	2094	1331024
206	641926	1502	6230944	1548	6130770	1677	6250920	2094	1361024
206	651926	1502	6240944	1548	6140770	1677	6260920	2094	1371024
206	661926	1502	6280944	1548	6150770	1696	6000944	2094	1381024
250	700020	1502	6290944	1548	6160770	1696	6010944	2094	1391024
250	710030	1538	6270951	1548	6170770	1696	6020944	2094	1411024
250	720020	1538	6280951	1548	6180770	1696	6030944	2094	1431024
250	730020	1547	6001920	1588	6000937	1696	6040944	2094	1441024
250	740030	1547	6011920	1588	6000942	1696	6050944	2094	1461024
324	410938	1547	6021920	1588	6010937	1696	6060944	2094	1471024
525	6230954	1547	6031920	1588	6010942	1696	6070944	2094	1481024
525	6240954	1547	6041920	1588	6020937	1696	6080944	2094	1501024
525	6250954	1547	6051920	1588	6020942	1696	6090944	2094	1521024
525	6260954	1547	6061920	1588	6030937	1696	6100944	2094	1541024
653	470931	1547	6071920	1588	6030942	1696	6110944	2095	500932
653	480931	1547	6081920	1588	6040937	1696	6120944	2095	681932
801	511932	1547	6091920	1588	6040942	1696	6130944	2095	691932
801	521932	1547	6101885	1588	6050937	1696	6250944	2110	6000414
838	460932	1547	6111885	1588	6050942	1696	6260944	2110	6000752
1092	260932	1547	6121885	1588	6060937	1696	6270944	2110	6000758
1092	270932	1547	6131885	1588	6060942	1696	6300944	2110	6000859
1199	290932	1547	6141885	1588	6070937	1696	6310944	2110	6000862
1358	611932	1547	6151885	1588	6070942	1696	6320944	2110	6000907
1413	621932	1547	6161885	1588	6080937	1696	6330944	2110	6000951
1453	6000885	1547	6161920	1588	6080942	1696	6340944	2110	6000977
1453	6010885	1547	6171885	1588	6090942	1960	6001944	2110	6000995
1453	6020885	1547	6171920	1677	6000920	1960	6011944	2110	6000999
1453	6030885	1547	6181885	1677	6010920	1960	6021944	2110	6010752
1453	6040885	1547	6181920	1677	6020920	1960	6031944	2110	6010758
1453	6050885	1547	6191920	1677	6030920	1960	6041944	2110	6010859
1453	6060885	1547	6201920	1677	6040920	1960	6051835	2110	6010951
1453	6070885	1547	6211920	1677	6050920	1960	6061835	2110	6010977
1453	6080885	1547	6221920	1677	6060920	1960	6071835	2110	6020752
1453	6090885	1547	6231920	1677	6070920	1960	6081835	2110	6020758
1453	6100885	1547	6241920	1677	6080920	1960	6091835	2110	6020859
1453	6110885	1547	6251920	1677	6090920	2004	420932	2110	6020951
1453	6120885	1547	6261920	1677	6100920	2086	240977	2110	6030752
1453	6130885	1547	6271920	1677	6110920	2086	490977	2110	6030758
1453	6140885	1548	6000770	1677	6120920	2094	1051024	2110	6030859
1502	6000953	1548	6010770	1677	6130920	2094	1121024	2110	6030951
1502	6010953	1548	6020770	1677	6140920	2094	1151024	2110	6040752

Table M12.2.2 (continued)

Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID	Citation ID	Problem ID
2110	6040758	2112	1221043	2117	6491885	2122	6101920	2126	911025
2110	6040859	2112	1231043	2117	6501885	2122	6111920	2126	921025
2110	6040951	2112	1241043	2117	6511885	2122	6121920	2126	931025
2110	6050752	2112	1251043	2117	6521885	2122	6131920	2126	941025
2110	6050758	2112	1261043	2117	6531885	2122	6141920	2126	951025
2110	6050859	2112	1271043	2117	6541885	2122	6151920	2126	961025
2110	6050951	2112	1281043	2117	6551885	2123	6220974	2126	971025
2110	6060752	2112	1291043	2117	6561885	2124	750045	2126	981025
2110	6060859	2112	1301043	2117	6571885	2125	11024	2129	991048
2110	6060951	2112	1311043	2117	6581885	2125	21024	2129	1001048
2110	6070752	2112	1321043	2117	6591885	2125	31024	2129	1011048
2110	6070951	2116	6001885	2117	6601885	2125	41024	2129	1021048
2110	6080752	2116	6191885	2117	6611885	2125	51024	2129	1031048
2110	6090752	2116	6201885	2117	6621885	2125	61024	2129	1041048
2110	6160951	2116	6211885	2117	6631885	2125	91024	2129	1051048
2110	6170951	2117	6271885	2117	6641885	2125	111024	2129	1061048
2110	6180862	2117	6281885	2117	6651885	2125	121024	2129	1071048
2110	6190907	2117	6291885	2117	6661885	2125	141043	2129	1081048
2110	6200414	2117	6301885	2118	6221885	2125	161043	2129	1091048
2110	6220951	2117	6311885	2118	6231885	2125	171043	2129	1101048
2112	1051043	2117	6321885	2118	6241885	2125	191043	2129	1111048
2112	1061043	2117	6331885	2118	6251885	2125	211043	2129	1121048
2112	1071043	2117	6341885	2118	6261885	2125	221043	2129	1131048
2112	1081043	2117	6351885	2119	300932	2125	231043	2129	1141048
2112	1091043	2117	6361885	2119	310932	2126	781025	2129	1151048
2112	1101043	2117	6371885	2119	440932	2126	791025	2129	1161048
2112	1111043	2117	6381885	2119	9000977	2126	801025	2129	1171048
2112	1121043	2117	6391885	2120	381027	2126	811025	2129	1181048
2112	1131043	2117	6401885	2121	390932	2126	821025	2129	1191048
2112	1141043	2117	6411885	2121	400932	2126	831025	2129	1201048
2112	1151043	2117	6421885	2121	671930	2126	841025	2129	1211048
2112	1161043	2117	6431885	2122	321024	2126	851025	2129	1221048
2112	1171043	2117	6441885	2122	331024	2126	861025	2130	1251011
2112	1181043	2117	6451885	2122	341024	2126	871025	2131	280050
2112	1191043	2117	6461885	2122	351024	2126	881025		
2112	1201043	2117	6471885	2122	361024	2126	891025		
2112	1211043	2117	6481885	2122	371024	2126	901025		

M12.3 PROGRAM FLOW

The main program reads the CESAR parameters that define the problem to be run. Subroutine MESAGE is called to print the header page. Subroutines SCANON, SCANOF, and IREAD are called as needed by the free-form data input system (see Sect. M3). If the library is being updated, subroutines COPY, ADD, and RIN are called. If an edit is requested, subroutines RLIB and OUTPT are called, and if the edit is for citation ID's, CROSS is called.

Free-Form Data Input Routines

Subroutines SCANON, SCANOF, IREAD, FREAD, and AREAD are called at various places throughout the program to control the reading of free-form input data. SCANON and SCANOF turn on and turn off the scan-ahead feature which checks to see if the next item in the input stream is an END. IREAD, FREAD, and AREAD read input data as integers, floating point decimal, or alphanumeric data.

Subroutine CROSS

This subroutine is called when the citation ID sought is not found; a cross-reference check is made to see if another citation ID refers to the same set of experiments. If found, an edit of the alternate citation ID problem set is given. RLIB and OUTPT are called.

Subroutine PTYPE

This subroutine searches for any experiments that fit the specifications of a problem-type edit. All experiments that match the input enrichment, fuel type, and heterogeneous or homogeneous requirements are edited. SCANON, SCANOF, FREAD, IREAD, and OUTPT are called.

Subroutine OUTPT

This subroutine prints and punches the results of the edits requested by the input data.

Subroutine ADD

This subroutine writes the new CESAR library when the data is updated.

Subroutine COPY

This subroutine reads the old CESAR library for transfer to the new library during updating.

Subroutine RLIB

This subroutine reads the CESAR library for purposes of finding the data to be edited. All edit types use this routine.

M12.4 INPUT DESCRIPTION

CESAR uses free-form input data (see Sect. M3). Enter one or more blanks between each entry. All 80 columns of a card may be used. Bold face words or numbers represent actual data to be input.

Module Activation Record: Module name beginning in column 1, **=CESAR**

Data Record 1: Control parameters

IDA **problem ID** to be edited or corrected. Enter 0 if not used.

LIB
- **0** not used
- **1** make new library
- **2** add to old library
- **3** replace entry IDA
- **4** delete entry IDA
- **5** update citation ID cross-reference data

LEDIT
- **0** not used
- **1** edit entire library
- **2** edit entry IDA
- **3** problem-type edit
- **4** edit IDD array of ID's
- **5** edit citation ID
- **6** edit all citation ID's

JTC
- **0** no effect
- **1** print problem description
- **2** punch problem description
- **3** print and punch problem description

JPD
- **0** no effect
- **1** print problem data
- **2** punch problem data
- **3** print and punch problem data

KIDA **citation ID** to be edited. Enter 0 if not used.

NEXA **citation experiment no.** Enter 0 if not used.

Data Record 2: Problem-type data (enter only if LEDIT = 3)

ENRL lower enrichment, **wt. percent**

ENRH upper enrichment, **wt. percent** (enter 0 for no upper limit)

IHOMA
- **0** not used
- **1** homogeneous
- **2** heterogeneous

IFUELA
- 0 not used
- 1 ^{235}U
- 2 ^{239}Pu or $^{239}Pu/^{235}U$
- 3 ^{233}U

Data Record 3: Enter only if LEDIT = 4.

IDD **Array of problem ID's** to be edited. End the array with a zero entry. Maximum of 50 entries.

Data Record 4: Library additions or replacements (enter only if LIB = 1, 2, 3, or 4).

problem ID citation ID experiment number
description of problem (100 card maximum)
$$$$
input data (1500 card maximum)
$$$$

The above data can be repeated for as many problems as are added or replaced. Terminate the data by entering a **0** for the problem ID.

Data Record 5: Cross-reference data update (enter only if LIB = 5).

MKID Array of 40 citation ID's not found in the library but which are referenced to other citation ID's found in the library. Enter 0 for values not used.

KIDX 5 citation ID's for **each** MKID entry. Each KIDX entry is a citation ID in the library referenced by the respective MKID entry ID. Enter a 0 for values not used.

Module Termination Record: END (starting in column 1)

The input/output assignments for the edits are as follows:

LOGICAL NO.	PURPOSE
5	card input
6	printed output
7	punched card output
30	current library
40	updated library
50	citation ID cross-reference data

M12.5 SAMPLE PROBLEM

Sample problem 1. A complete library edit of the problem description is desired. Thus LEDIT=1, JTC=1, and all other parameters are 0. The input data would be:

```
=CESAR
0 0 1 1 0 0 0
END
```

Sample problem 2. A search for all ^{235}U systems with enrichment equal to or less than 3% is desired. The problem description should be printed, and the problem data should be provided on punched cards. Then LEDIT=3, JTC=1, and JPD=2. All other parameters are 0. For the problem-type edit data, ENRH=3 and IFUELA=1. All other items are 0. The input data would be:

```
=CESAR
0 0 3 1 2 0 0
0 3 0 1
END
```

Sample problem 3. A list of all of the problem datasets having a citation ID of 2112 is desired. Thus LEDIT=5, JPD=1, KIDA=2112, and all other parameters are 0. The input data would be:

```
=CESAR
0 0 5 0 1 2112 0
END
```

Sample problem 4. A list of the problem descriptions and problem data for problem ID 511932 is desired. Thus IDA=511932, LEDIT=2, JTC=1, JPD=1, and the remaining parameters are 0. The input data would be:

```
=CESAR
511932 0 2 1 1 0 0
END
```

The output listing for Sample Problem 4 is given on the following pages. Note that some edits can create a large amount of printed or punched card output.

```
IDA     511932  ID OF PROBLEM TO BE EDITED OR CORRECTED
LIB          0  0/1/2/3/4/5 NO EFFECT/MAKE NEW LIBRARY/ADD TO OLD LIBRARY/REPLACE ENTRY IDA/DELETE ENTRY IDA
                /UPDATE KOPONEN CROSS REF
LEDIT        2  0/1/2/3/4/5/6 NO EFFECT/EDIT ENTIRE LIBRARY/EDIT ENTRY IDA/PROBLEM TYPE EDIT/EDIT IDD ARRAY
                /EDIT KOPONEN ID/EDIT ALL KOPONEN ID'S
JTC          1  0/1/2/3   NO EFFECT/PRINT/PUNCH/PRINT & PUNCH PROBLEM DESCRIPTION
JPD          1  0/1/2/3   NO EFFECT/PRINT/PUNCH/PRINT & PUNCH PROBLEM DATA

PROBLEM ID  511932    KOPONEN ID  801  EXP. NO.  0

U METAL, 93.2% U-235, CYLINDER ANNULUS 13.97 CM INSIDE RADIUS, 19.05 CM OUTSIDE
RADIUS, 10.11 CM HEIGHT. 8.89 CM RADIUS, 10.11 CM HIGH CYLINDER IN CENTRAL HOLE
TOUCHING WALL. UNREFLECTED.
REFERENCE: TRANS. AM. NUC. SOC., 7,287(1964)
SCALE MODULE: CSAS25

=CSAS25
KENO-5 VALIDATION CASE B-1
HANSEN-ROACH  MULTIREGION
U-235  1 0 4.4796-2 END
U-238  1 0 3.2271-3 END
END COMP
CYLINDRICAL  END
1 8.89 NOEXTERMOD END ZONE
KENO-5 VALIDATION CASE B-1
READ PARAM TKE=4 GEN=203 END PARAM
READ GEOM
CYLINDER  1 1 8.89 10.1092 0. ORIGIN -5.08 0.
CYLINDER  0 1 13.97 10.1092 0.
CYLINDER  1 1 19.05 10.1092 0.
END GEOM
END DATA
END


ET TU, BRUTE! THEN FALL, CESAR.
```

```
      PRIMARY MODULE ACCESS AND INPUT RECORD ( SCALE DRIVER - 11/24/82 - 16.59)

MODULE CESAR    WILL BE CALLED  TIME OF DAY 14.49.06  DATE 84.074

  511932 0 2 1 1 0 0
  END

MODULE CESAR    IS FINISHED.  COMPLETION CODE - SYSTEM 000 USER  0000. CPU TIME USED      .24 (SECONDS). I/O'S USED      54.
```

M12.5.3

M12.6 REFERENCES

1. B. L. Koponen, T. P. Wilcox, and V. E. Hampel, "Nuclear Criticality Experiments from 1943 to 1978, an Annotated Bibliography," UCRL-52769 (April 1979).

2. N. M. Greene, J. L. Lucius, L. M. Petrie, W. E. Ford, III, J. E. White, and R. Q. Wright, "AMPX - A Modular Code System to Generate Coupled Multigroup Neutron-Gamma Cross Sections from ENDF/B," ORNL/TM-3706 (AMPX-1) (March 1976).

3. L. M. Petrie and N. F. Cross, "KENO IV - An Improved Monte Carlo Criticality Program," ORNL-4938 (November 1975).

NUREG/CR-0200
Volume 3, Section M13
ORNL/NUREG/CSD-2/V3/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

PICTURE: A Printer Plot Package for Making 2-D Pictures of MARS Geometries

Margaret B. Emmett

Manuscript Completed: August 1984

Date Published: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

CONTENTS

LIST OF FIGURES M13.v

ACKNOWLEDGEMENTS M13.vii

ABSTRACT M13.ix

M13.1 INTRODUCTION M13.1.1

M13.2 ROUTINES M13.2.1

M13.2.1 Main Program M13.2.1
M13.2.2 Subroutines M13.2.1

M13.3 INPUT DATA M13.3.1

M13.4 OPTIONS M13.4.1

M13.5 PICTURE SAMPLE PROBLEM M13.5.1

LIST OF FIGURES

Fig. M13.5.1 Input Data for PICTURE Sample Problem M13.5.2

Fig. M13.5.2 Output of Combinatorial Geometry Tank Sample Problem M13.5.3

ACKNOWLEDGEMENT

The funding for this update to the PICTURE code was provided by NRC's Office of Nuclear Material Safety and Safeguards. Special thanks to Lindy Norris for typing this manual.

ABSTRACT

The PICTURE program was written to provide aid in preparing correct input data for the MARS geometry package which is a multiple array system using combinatorial geometry. It provides a printed view of arbitrary two-dimensional slices through the geometry. By inspecting these pictures one may determine if the geometry specified by the input cards is indeed the desired geometry. This report describes PICTURE, its options, and input.

M13.1 INTRODUCTION

The PICTURE program was written to help the user determine whether his MARS input data correctly describes his geometry. PICTURE displays, as printed output, two-dimensional slices through the specified geometry. A regularly spaced array of points is generated and each point is plotted as a symbol related to either media, region, or zone, depending on the option selected by the user. By printing out this array, a rough picture of one view of the geometry is produced. The user may then look at the picture and determine if the geometry is as intended. A sample problem is described in Section M13.5.

Running PICTURE requires the routines described in this report plus those making up the MARS geometry package described in Sect. M9. MARS also uses routines from the SCALE subroutine library.

PICTURE is used by some of the control modules in SCALE.

M13.2 ROUTINES

M13.2.1 Main Program

The executive routine for the PICTURE program sets the dimension for blank common, the I/O unit numbers, and the ratio of the number of characters per inch down the page to the number across the page, and calls subroutine PICTUR.

Subroutines called: PICTUR

Commons required: Blank

Variables changed: INT, IOT, RAYX, NDIM

Significant internal variables:

INT -	input logical unit (=5),
IOT -	output logical unit (=6),
RAYX -	ratio of number characters/inch down to number characters/inch across page (default = 0.8 for 14 x 8 1/2 paper; user should use 0.6 for 14 x 11 paper).

M13.2.2 Subroutines

Subroutine PRINT (KXX, KYY, ATABLE)

This routine controls the printing of the picture. First it is determined if the picture is to be more than one "page" (130 columns) wide, and then for each line of the picture on the first page the information to be printed is determined by calling MESH and printed with a 130A1 format. The next lines are then calculated and printed until that "page" is finished. Note that a page refers to width, not length. Thus, as much detail may be obtained as necessary in both directions by piecing together the output.

Called from: PICTUR

Subroutine called: MESH

Commons required: PICT

Variables required:

KXX -	number of intervals in the U direction (direction of paper movement through the printer),
KYY -	number of intervals in the V direction (line),
ATABLE -	table of characters to be printed.

Significant interval variable:

NPAGES -	number of subpictures required to cover the width of the total picture,
IOT -	output logical unit,
NV -	number of characters per line (characters/page width).

Subroutine MESH (XS, YS, ZS, NV)

Subroutine MESH is used by the PICTURE package to set up one line of print in the array NSTOR. Both for efficiency and to debug the MARS array geometry data, this version has been modified to work exactly like particle tracking. CALI is first called to determine the zone of the first grid point. A trajectory to the last grid point is then initialized, and successive calls to PILOT, which calls 'G1,' "track" a particle to the last point, setting the region of each grid point in NSTOR. By setting IRG negative, zero, or positive, either NREG, IR, or NMED will be stored in the print array NSTOR.

Called from: PRINT

Subroutines called: CALI, RESET, PILOT

Commons required: PICT, GOMLOC, PAREM, ORGI, REPEAT, ARAR, MG0MV, SS, JOMN2, blank.

Variables required:

XS, YS, ZS -	coordinates of first grid point,
NV	number of grid points,
DELV -	distance between grid points,
IRG -	flag to print NREG, IR, or NMED if IRG is negative, zero, or positive,
XV, YV, ZV -	coordinates of width (in direction V) of picture in geometry units.

Variables changed: NSTOR - print array.

Significant internal variables:

J -	grid point index,
ISTOR -	value to be stored in NSTOR between successive boundary crossings,
IROLD -	present input zone number,
SUMS1 -	distance traveled on current line (DIST),
DLINE -	total distance to end of current line (DIST0).

Subroutine PICTUR (INT, IOT, NDIM, RAYX)

Subroutine PICTUR reads in the input, calculates the coordinates of the picture to be plotted, and controls the calls to other routines. There are several different ways in which a two-dimensional slice through the geometry may be obtained. These different options are discussed in Sect. M13.4 with discussions of the input. The characters to be printed for corresponding media or regions may be changed by altering the values in ATABLE as given in the data statement.

Called from: Main Program

Subroutines called: JOMIN, PRINT

Commons required: PICT, MGOMV

Variables required: Several input cards are read.

INT - input logical unit,
IOT - output logical unit.

Variables changed: All variables in common PICT (see Table M13.2.2.1)

IPRET in MGOMV†

Significant internal variables:

NADD(1) - first location in blank common for storage of geometry data,

NCK - flag to indicate which set of input options was used to define the two-dimensional slice.

---

†(see Table M9.E.8)

Table M13.2.2.1 Definition of variables in common PICT

Variable	Definition
DELU	The increment in geometry units between lines in the picture in the U direction.
DELV	The increment in geometry units between lines in the picture in the V direction.
X0, Y0, Z0	The coordinates in geometry units defining the first point (upper left hand corner) of the picture.
XU, YU, ZU	The length (in direction U) of the picture in geometry units.
XV, YV, ZV	The width (in direction V) of the picture in geometry units.
NSTOR(130)	An array used to store the medium, region, or zone number for one line of the picture.
IRG	A flag indicating that region, zone, or medium geometry parameter should be printed if IRG is negative, zero, or positive, respectively.

M13.3 INPUT DATA

1. Card PA: Format (I5)

   NUSE: The number of characters to be read on Card B to replace the standard values of ATABLE. Leave Card A blank and omit Card B if the standard ATABLE is desired. NUSE $\leq$ 50.

2. Card PB: Format (50A1) (omit if NUSE = 0)

   ATABLE(I), I=1, NUSE: The list of characters that are to be printed for each medium. For medium N, ATABLE (N+1) is printed. If N $\geq$ 47, ATABLE (48) is printed. The standard values of ATABLE are:

Medium Number	Character Printed
0 (external void)	
1 through 9	1 through 9
10 through 35	A through Z
36 through 46	various special characters
$\geq$ 47 (including internal voids)	(blank)

3. GEOM input: MARS combinatorial geometry input.†

4. Card PC: Format (2I2, 18A4)

   INCT:
   - = 0 After this picture, return to Card PC for another picture with the same geometry.
   - = 1 After this picture, read in a new GEOM input. (step 3 above)

   IRG:
   - = -1 Display the region geometry.
   - = 0 Display the zone geometry.
   - = 1 Display the material geometry.

   TITLE(I), I=1, 18 72 characters to be printed as a title.

5. Card PD: Format (6E10.5)

   X_{UL}, Y_{UL}, Z_{UL} X, Y, and Z coordinates in the combinatorial geometry of the upper left corner of the picture.

   X_{LR}, Y_{LR}, Z_{LR} X, Y, and Z coordinates in the combinatorial geometry of the lower right corner of the picture.

   Note: Card PD partially describes the plane of the slice by defining two points in the plane and designates the top, bottom, left, and right sides of the picture.

†See description of geometry input in Section M9.A.

6. Card PL: Format (6E10.5)

U_X
U_Y Direction numbers proportional to the direction cosines for the U axis of the picture. The U axis points down the printed page in the direction the page moves through the printer.
U_Z

V_X
V_Y Direction numbers for the V axis of the picture. The V axis points to the right across the page.
V_Y

NOTE: Card PL completes the description of the plane of the slice by giving a line in the plane, also specifies the orientation of the picture on the output.

7. Card PF: Format (2I5, 2E10.5)

NU: Number of intervals to print along the U axis (overrides DELU).
NV: Number of intervals to print along the V axis (overrides DELV).
DELU: Spacing (in GEOM units) of intervals along the U axis.
DELV: Spacing (in GEOM units) of intervals along the V axis.

NOTE: All four entries are not required as input on Card PF; see Section M13.4 for explanation.

M13.4 OPTIONS

1. $X_{LR} = Y_{LR} = Z_{LR} = 0.$

   For this case NU and NV must be specified. In addition, either DELU or DELV must be specified. If the other is left blank, the code will produce an undistorted picture. If both DELU and DELV are specified, the picture is likely to be distorted. The standard printers give 10 characters to the inch across a line but only 6 lines per inch down the page. Because of this, DELV = 0.6*DELU is necessary to produce an undistorted picture.

2. X_{LR} or Y_{LR} or Z_{LR} 0.

   If any one variable on Card PD is specified, the code will calculate the others to produce an undistorted picture.

   If both NU and DELU (or both NV and DELV) are specified, DELU (or DELV) will be ignored.

   The U and V axes may have arbitrary orientation. (If they are not orthogonal, the resulting picture will be distorted.) In Option 1, the first point will be at $(X, Y, Z)_{UL}$, and the remaining points in the directions and at the distances specified. In Option 2, the range from X_{UL} to X_{LR} is divided into intervals and the calculated points will be at the midpoints of the intervals. The first point will be 1/2 interval past $(X, Y, Z)_{UL}$ and the final point will be within 1/2 interval of $(X, Y, Z)_{LR}$. If $(X, Y, Z)_{LR}$ does not lie on the U-V plane, or if the U and V axes are not orthogonal, the location of the final point is not readily predictable.

   The simplest method to obtain the correct results is to specify two diagonal corners of the plane of the slice on Card PD, with the top having the short dimension. Then, on Card PE, specify the U axis to be parallel to the edge of the slice with the large dimension (left or right side), and the V axis to be parallel to the edge of the slice with the small dimension (top or bottom). Finally, let the only entry on Card PF be NV equal to maximum number of characters per line on your printer; this will provide the largest undistorted picture.

M13.5 PICTURE SAMPLE PROBLEM

This problem illustrates the use of the PICTURE program for looking at the geometry of a given problem. The geometry being illustrated is a tank. This tank model was constructed purely as an illustration of the combinatorial geometry and is in no way accurate or detailed. Both the use of the OR operator and the ARB body is demonstrated. The input data are listed in Fig. M13.5.1. The output and picture produced by the problem are shown in Fig. M13.5.2.

```
   7
.1ETA0
COMBINATORIAL GEOMETRY TANK SAMPLE PROBLEM
 0 0 0 10
 ARB       1.25       -1.25       .5         1.25       -2.0       0.0
           1.25       2.0         0.0        1.25       1.25       .5
           -1.25      -1.25       .5         -1.25      -2.0        0.0
           -1.25      2.0         0.0        -1.25       1.25       .5
           1234.      4158.       6587.      2673.      5621.      4378.
 ELL       0.0        -.5         .5         0.0        .5         .5
           1.5
 TRC       0.0        0.0         .8         0.0        2.5        0.0
           .1         .05
 ARB       1.25       -2.0        0.0        1.25       -1.8       -.8
           1.25        2.0        0.0        1.25        1.8       -.8
          -1.25        2.0        0.0       -1.25        1.8       -.8
          -1.25       -2.0        0.0       -1.25       -1.8       -.8
           1243.      7135.       8756.      2864.      7821.      3465.
 BOX       1.25       -1.979      -.6895     0.0        .358       .179
           0.0        -.2         .4         -2.5       0.0        0.0
 BOX       1.25       1.979       -.6895     0.0        -.358      .179
           0.0        .2          .4         -2.5       0.0        0.0
 RCC       1.25       -2.0        -.2        -2.5       0.0        0.0
           .2
 RCC       1.25       -1.8        -.6        -2.5       0.0        0.0
           .2
 RCC       1.25        2.0        -.2        -2.5       0.0        0.0
           .2
 RCC       1.25       1.80        -.6        -2.5       0.0        0.0
           .2
 RCC       1.25       -.9         -.6        -2.5       0.0        0.0
           .2
 RCC       1.25       -.45        -.2        -2.5       0.0        0.0
           .2
 RCC       1.25       0.0         -.6        -2.5       0.0        0.0
           .2
 RCC       1.25       .45         -.2        -2.5       0.0        0.0
           .2
 RCC       1.25       .9          -.6        -2.5       0.0        0.0
           .2
 RPP       -10.       10.         -10.       10.        -10.       10.
 END
  Z1          +1
  Z2          +2      -1
  Z3          +3      -2
  Z4    OR    +4      -7      -8      -9      -10     -11     -12     -13     14
            -15OR     +5      -7      -8OR     +6      -9      -10
  Z5    OR    +7OR    +8OR    +9OR   +10OR   +11OR   +12OR   +13OR   +14OR   15
  Z6         +16      -1      -2      -3       -4      -5      -6      -7      -8
              -9      -10
 END
   1    1     1     1     1     1
 6*0
   1    2     3     4     5 1000
0
0 0  THIS IS A COMBINATORIAL GEOMETRY TANK.
0.0        -3.0        -2.0        0.0        3.0        2.0
0.0        1.0         0.0         0.0        0.0        1.0
      130
```

Fig. M13.5.1. Input Data for PICTURE Sample Problem

COMBINATORIAL GEOMETRY TANK SAMPLE PROBLEM

IVOPT = 0 IDBG = 0

BODY DATA

ARB	1	0.1250000D+01	-0.1250000D+01	0.5000000D+00	0.1250000D+01	-0.2000000D+01	0.0
		0.1250000D+01	0.2000000D+01	0.0	0.1250000D+01	0.1250000D+01	0.5000000D+00
		-0.1250000D+01	-0.1250000D+01	0.5000000D+00	-0.1250000D+01	-0.2000000D+01	0.0
		-0.1250000D+01	0.2000000D+01	0.0	-0.1250000D+01	0.1250000D+01	0.5000000D+00
		0.1234000D+04	0.4158000D+04	0.6587000D+04	0.2673000D+04	0.5621000D+04	0.4378000D+04
			-0.1000000D+01	0.0	0.0	0.1250000D+01	
			0.0	0.0	-0.1000000D+01	0.5000000D+00	
			0.1000000D+01	0.0	0.0	0.1250000D+01	
			0.0	0.0	0.1000000D+01	0.0	
			0.0	0.5547002D+00	-0.8320503D+00	0.1109400D+01	
			0.0	-0.5547002D+00	-0.8320503D+00	0.1109400D+01	
			0.9013878D+00	0.6000000D+01			
ELL	2	0.0	-0.5000000D+00	0.5000000D+00	0.0	0.5000000D+00	0.5000000D+00
		0.1500000D+01					
TRC	3	0.0	0.0	0.8000000D+00	0.0	0.2500000D+01	0.0
		0.1000000D+00	0.5000000D-01				
ARB	4	0.1250000D+01	-0.2000000D+01	0.0	0.1250000D+01	-0.1800000D+01	-0.8000000D+00
		0.1250000D+01	0.2000000D+01	0.0	0.1250000D+01	0.1800000D+01	-0.8000000D+00
		-0.1250000D+01	0.2000000D+01	0.0	-0.1250000D+01	0.1800000D+01	-0.8000000D+00
		-0.1250000D+01	-0.2000000D+01	0.0	-0.1250000D+01	-0.1800000D+01	-0.8000000D+00
		0.1243000D+04	0.7135000D+04	0.8756000D+04	0.2864000D+04	0.7821000D+04	0.3465000D+04
			-0.1000000D+01	0.0	0.0	0.1250000D+01	
			0.0	0.0	-0.1000000D+01	0.0	
			0.1000000D+01	0.0	0.0	0.1250000D+01	
			0.0	0.0	0.1000000D+01	0.8000000D+00	
			0.0	0.9701425D+00	0.2425356D+00	0.1940285D+01	
			0.0	-0.9701425D+00	0.2425356D+00	0.1940285D+01	
			0.8246211D+00	0.6000000D+01			
BOX	5	0.1250000D+01	-0.1979000D+01	-0.6895000D+00	0.0	0.3580000D+00	0.1790000D+00
		0.0	-0.2000000D+00	0.4000000D+00	-0.2500000D+01	0.0	0.0
BOX	6	0.1250000D+01	0.1979000D+01	-0.6895000D+00	0.0	-0.3580000D+00	0.1790000D+00
		0.0	0.2000000D+00	0.4000000D+00	-0.2500000D+01	0.0	0.0
RCC	7	0.1250000D+01	-0.2000000D+01	-0.2000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	8	0.1250000D+01	-0.1800000D+01	-0.6000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	9	0.1250000D+01	0.2000000D+01	-0.2000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	10	0.1250000D+01	0.1800000D+01	-0.6000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	11	0.1250000D+01	-0.9000000D+00	-0.6000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	12	0.1250000D+01	-0.4500000D+00	-0.2000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	13	0.1250000D+01	0.0	-0.6000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	14	0.1250000D+01	0.4500000D+00	-0.2000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RCC	15	0.1250000D+01	0.9000000D+00	-0.6000000D+00	-0.2500000D+01	0.0	0.0
		0.2000000D+00					
RPP	16	-0.1000000D+02	0.1000000D+02	-0.1000000D+02	0.1000000D+02	-0.1000000D+02	0.1000000D+02

NUMBER OF BODIES 16
LENGTH OF FPD-ARRAY 252

Fig. M13.5.2. Output of Combinatorial Geometry Tank Sample Problem

	INPUT ZONE	CODE ZONE		BODY NUMBERS	INPUT ZONE DATA						
Z1	1	1		1							
Z2	2	2		2	-1						
Z3	3	3		3	-2						
Z4	4	4		4	-7	-8	-9	-10	-11	-12	-13	-14
				-15								
		5	OR	5	-7	-8						
		6	OR	6	-9	-10						
Z5	5	7		7								
		8	OR	8								
		9	OR	9								
		10	OR	10								
		11	OR	11								
		12	OR	12								
		13	OR	13								
		14	OR	14								
		15	OR	15								
Z6	6	16		16	-1	-2	-3	-4	-5	-6	-7	-8
				-9	-10							

NUMBER OF INPUT ZONES 6
NUMBER OF CODE ZONES 16
LENGTH OF INTEGER ARRAY 453

CODE ZONE	INPUT ZONE	ZONE DATA LOC.	NO. OF BODIES	REGION NO.	MEDIA NO.	BOX INPUT ZONE	BOX CODE ZONE
1	1	113	1	1	1	0	0
2	2	119	2	1	2	0	0
3	3	130	2	1	3	0	0
4	4	141	10	1	4	0	0
5	4	192	3	1	4	0	0
6	4	208	3	1	4	0	0
7	5	224	1	1	5	0	0
8	5	230	1	1	5	0	0
9	5	236	1	1	5	0	0
10	5	242	1	1	5	0	0
11	5	248	1	1	5	0	0
12	5	254	1	1	5	0	0
13	5	260	1	1	5	0	0
14	5	266	1	1	5	0	0
15	5	272	1	1	5	0	0
16	6	278	11	1	1000	0	0

I	KR1(I)	KR2(I)
1	1	1
2	2	2
3	3	3
4	4	6
5	7	15
6	16	16

MORSE REGION IN INPUT ZONE(I) ARRAY (MR1Z(I),I=1, 6)

1 1 1 1 1 1

MORSE MEDIA IN INPUT ZONE(I) ARRAY (MM1Z(I),I=1, 6)

1 2 3 4 5 1000

MORSE UNIVERSE IN INPUT ZONE(I) ARRAY (NB1Z(I),I=1, 6)

0 0 0 0 0 0

OPTION 0 WAS USED IN CALCULATING VOLUMES, FOR 1 REGIONS
0-SET VOLUMES = 1, 1-CONCENTRIC SPHERES, 2-SLABS, 3-INPUTVOLUMES.

VOLUMES (CC) USED IN COLLISIONS DENSITY AND TRACK LENGTH ESTIMATORS.

REG	1
VOLUME	1.000D+00

ARRAY DATA REQUIRES 8 LOCATIONS, LEAVING 12787 LOCATIONS

Fig. M13.5.2 (continued)

```
 THIS IS A COMBINATORIAL GEOMETRY TANK.
ZONE GEOMETRY
THE SELECTED ATABLE VALUES ARE
      I E T A 0
     UPPER LEFT    LOWER RIGHT
     COORDINATES   COORDINATES
X      0.0            0.0
Y     -0.3000E+01     0.3000E+01
Z     -0.2000E+01     0.2000E+01
        U AXIS         V AXIS
       (DOWN)         (ACROSS)
X      0.0            0.0
Y      1.00000        0.0
Z      0.0            1.00000
NU= 157   NV= 130   DELU= 0.3846E-01   DELV= 0.3077E-01
```

Fig. M13.5.2 (continued)

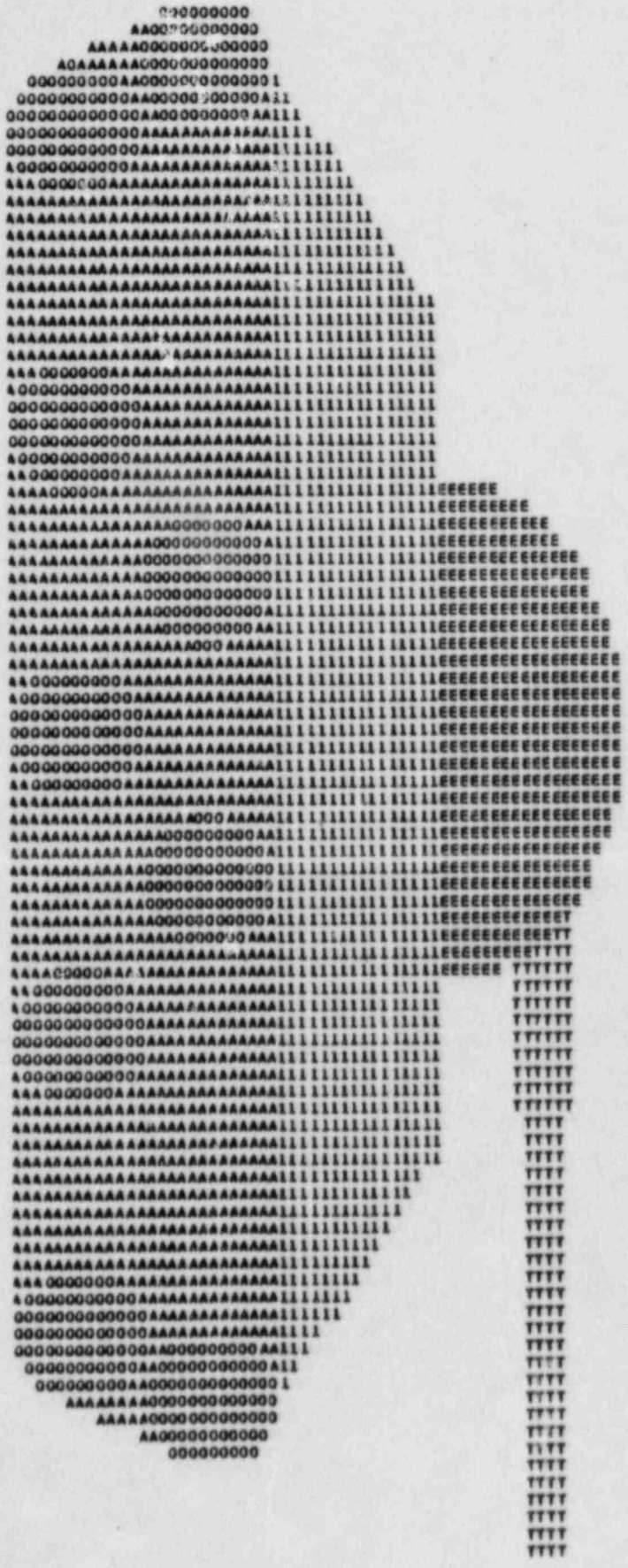


Fig. M13.5.2 (continued)

NUREG/CR-0200
Volume 3, Section M14
ORNL/NUREG/CSD-2/V3/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

COMPOZ Data Guide

J. R. Knight

Manuscript Completed: July 1984
Date Published: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

CONTENTS

ACKNOWLEDGEMENT M14.v

ABSTRACT M14.vii

M14.1 INTRODUCTION M14.1.1

M14.2 INPUT DATA DESCRIPTION M14.2.1

M14.2.1 HEAD RECORD CONTENTS M14.2.1
M14.2.2 STANDARD COMPOSITION DIRECTORY M14.2.2
M14.2.3 NUCLIDE INFORMATION TABLE M14.2.2
M14.2.4 ISOTOPIC DISTRIBUTION CATEGORY M14.2.2
M14.2.5 ISOTOPIC DISTRIBUTION TABLE M14.2.2
M14.2.6 STANDARD COMPOSITION TABLE M14.2.3

M14.3 UPDATE PROCEDURE AT OAK RIDGE FACILITIES M14.3.1

ACKNOWLEDGEMENT

This work was funded by the Transportation and Certification Branch of the Nuclear Regulatory Commission through the Office of Nuclear Material Safety and Safeguards.

ABSTRACT

The COMPOZ Data Guide used to create the Standard Composition Library is described. Of particular importance is documentation of the COMPOZ input data file structure. Knowledge of the file structure allows users to edit the data file and subsequently create their own site-specific composition library.

M14.1 INTRODUCTION

COMPOZ is the program which creates (writes) the SCALE Standard Composition Library (see Sect. M8). Data are input in free form using the AREAD, FREAD, and IREAD functions in SCALE (see Sects. M2-M3). A card-image data file containing the input to COMPOZ (and consequently the standard composition data) is available with the SCALE system. Execution of COMPOZ using this data file creates the Standard Composition Library currently available with the SCALE package. This section provides documentation of the data file structure. Knowledge of the data file structure allows users to edit the data file and subsequently create their own site-specific or user-specific composition library.

COMPOZ is intended to create or make permanent changes to the composition library and should not be used for any other purpose. To avoid confusion with the Standard Composition Library provided with SCALE, it is strongly recommended that only new keywords and compositions be used in any site-specific or user-specific library.

M14.2 INPUT DATA DESCRIPTION

COMPOZ input data are entered in free form (see Sect. M3). Alphanumeric data, such as nuclide and composition names, must be followed by two or more blanks before the next data item is entered. Numeric data must be followed by at least one blank. The COMPOZ input data file contains six data records or blocks:

1. The head record contains the library identification, date, a set of parameters describing the size of the library, and a three-line library title with 80 characters per line.
2. The standard composition directory contains the names and the number of words required to describe each standard composition. Individual nuclides, mixtures, and compounds are all included in the directory and in the standard composition table, which will be described later.
3. The nuclide information table contains the nuclide identification number, atomic mass, resonance energy cross sections, and a resonance nuclide identifier for each nuclide.
4. The isotopic distribution directory lists the identification number of each element that allows a variable enrichment of the individual isotopes that make up the element. It also contains the number of words required to describe the default enrichment, given in the isotopic distribution table.
5. The isotopic distribution table contains the nuclide identification number and the percent of atoms of each nuclide used to give the default enrichment of each element in the isotopic distribution directory.
6. The standard composition table contains the theoretical density, number of elements, and other information about each standard composition. It includes all individual nuclides as well as mixtures and compounds.

Note that for executing COMPOZ via SCALE an **=COMPOZ** is required in the first eight columns of a record preceding the head record and an **END** is required in the first three columns of a record inserted after the standard composition table data.

It is necessary for the word count in the standard composition directory and the isotopic distribution directory to be correct since it is used to determine the space allocation and address of items in the direct access dataset written by COMPOZ.

M14.2.1 HEAD RECORD CONTENTS

1.	IDT	Library update number.
2.	MDY	Date - six digits, month, day, year.
3.	NSC	Number of standard compositions.
4.	NNUC	Number of nuclides.
5.	NELV	Number of elements with variable isotopic distribution.
6.	MXELSC	Maximum number of elements per standard composition.
7.	MXISEL	Maximum number of isotopes per element with variable isotopic distribution.
8.	TITLE	Three-line title with 80 characters per line.

M14.2.2 STANDARD COMPOSITION DIRECTORY

1. SCID Composition name, maximum of 12 characters.

2. NWD Number of words of storage in Standard Composition Table for above composition (NWD = 5 + 2 * Number of elements)

Repeat items 1 and 2 for all compositions. Nuclides are entered first, in order of increasing nuclide ID, followed by mixtures and compounds in arbitrary order. There should be a total of NSC pairs of entries.

M14.2.3 NUCLIDE INFORMATION TABLE

1. NZA Nuclide ID. This should be the mass number + 1000 * the atomic number.

2. AM Atomic mass, C-12 scale.

3. SIGS Resonance energy scattering cross section, barns.

4. SIGT Resonance energy total cross section, barns.

5. JRS Resonance nuclide identification.
   0, no resonance data;
   1, resonance data.

6. MTS Multiple thermal scatter identifier. Enter 1.

The resonance energy cross sections are averaged over the appropriate energy range for the nuclide. Items 1-6 are repeated for all nuclides. There should be a total of NNUC sets of entries. Entries should be made in order of increasing nuclide ID.

M14.2.4 ISOTOPIC DISTRIBUTION DIRECTORY

1. NZN 1000 * atomic number of variable isotope elements.
2. NWRD Number of words of storage in the isotopic distribution table for this element. (NWRD = 2 * number of isotopes in the default abundance).

There should be a total of NELV pairs of entries.

M14.2.5 ISOTOPIC DISTRIBUTION TABLE

1. ISZA Isotope ID

2. ABWP Default abundance, atom percent.

The default abundance is generally the naturally occurring abundance. For each element, items 1 and 2 are repeated until 100% total abundance is described, making a set for this element. The next element is described in the same fashion in the next set, etc. There should be a total of NELV sets. These sets must be in the same order as the NZN element entries in the isotopic distribution directory. The number of isotopes per element cannot exceed MXISEL.

M14.2.6 STANDARD COMPOSITION TABLE

1. ROTH — Theoretical density, gm/cm^3.

2. NEL — Number of elements.

3. IVIS — Does first element have variable isotope distribution?
   0, no;
   1, yes.

4. ICP — 0 for a mixture,
   1 for a compound.

5. IRS — 0 for no resonance data,
   1 for resonance data in any constituent.

6. NCZA — Element or nuclide ID

7. ATPM — Weight percent if ICP=0. Number of atoms per molecule if ICP=1.

For each composition, items 6 and 7 are repeated until all components of the composition are described. There should be NEL pairs of entries. Items 1 through 7 are entered in a similar fashion for all compositions. There should be a total of NSC composition descriptions. IF IVIS=1, the first NCZA for that composition must be one of the NZN values from the isotopic distributions directory. Only one isotopic distribution element per composition is allowed. The composition descriptions must be in the same order as the composition names in the standard composition directory.

M14.3 UPDATE PROCEDURE AT OAK RIDGE FACILITIES

As stated previously, updates obtained by the procedure described here are considered permanent. For this reason, no changes should be attempted without the approval of the individual in charge of maintaining SCALE datasets. At the date of this report, this is L. M. Petrie of the Nuclear Engineering Applications Department, Computer Services Division. The results of any update should be checked carefully to assure that it was executed properly, and test problems should be run to verify that the Standard Composition Library is functioning properly.

The job control statements required for an update are outlined below.

```
//A EXEC LIBR
//MASTER DD DSN=XXXXX,DISP=SHR
//SYSIN DD *
-OPT
-SEL YYYYY,EXEC
LIBRARIAN commands required to modify data entered here.
-END
//B EXEC SCALE
//GO.SYSIN DD DSN=*.A.EDIT.OSJOB,DISP=(OLD,DELETE)
```

In the above, XXXXX is the dataset name of the LIBRARIAN master containing the COMPOZ input data module, and YYYYY is the module name.

ornl

NUREG/CR-0200
VOLUMES 1, 2, 3
ORNL/NUREG/CSD-2/R3

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

REVISIONS AND ERRATA SHEETS FOR VOLUMES 1, 2, 3

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
under Interagency Agreement DOE 40-549-75

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

TABLE OF CONTENTS
FOR SCALE MANUAL

		Page*
ACKNOWLEDGMENTS		vii
ABSTRACT		ix
FOREWORD		xi
LIST OF REVISIONS		xv
	VOLUME 1: CONTROL MODULES	
Section C1	CSAS1: A ONE-DIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C1.1.1
Section C2	CSAS2: A MULTIDIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C2.1.1
Section C3	CSAS3: AN OPTIMUM CONCENTRATION CRITICALITY SAFETY ANALYSIS MODULE (to be issued at later date)	C3.1.1
Section C4	CSAS4: AN ENHANCED CRITICALITY SAFETY ANALYSIS MODULE WITH AN OPTIMUM PITCH SEARCH OPTION	C4.1.1
Section S1	SAS1: A ONE-DIMENSIONAL SHIELDING ANALYSIS MODULE (to be issued at later date)	S1.1.1
Section S2	SAS2: A COUPLED ONE-DIMENSIONAL DEPLETION AND SHIELDING ANALYSIS MODULE	S2.1.1
Section S3	SAS3: AN AUTOMATED MONTE CARLO SHIELDING ANALYSIS MODULE	S3.1.1
Section S4	SAS4: A MONTE CARLO SHIELDING ANALYSIS MODULE WITH AUTOMATIC BIASING. (to be issued at later date)	S4.1.1
Section H1	HTAS1: A TWO-DIMENSIONAL HEAT TRANSFER ANALYSIS OF FUEL CASKS	H1.1.1

*All section page numbers refer to the page where the section introduction begins.

VOLUME 2: FUNCTIONAL MODULES

		Page
Section F1	BONAMI-S: RESONANCE SELF-SHIELDING BY THE BONDARENKO METHOD	F1.1.1
Section F2	NITAWL-S: RESONANCE SELF-SHIELDING BY THE NORDHEIM METHOD	F2.1.1
Section F3	XSDRNPM-S: A ONE-DIMENSIONAL DISCRETE-ORDINATES PROGRAM FOR TRANSPORT ANALYSIS . . .	F3.1.1
Section F4	XSDOSE: A MODULE FOR CALCULATING FLUXES AND DOSE RATES AT POINTS OUTSIDE A SHIELD	F4.1.1
Section F5	KENO-IV/S: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM	F5.1.1
Section F6	COUPLE: SCALE SYSTEM MODULE TO PROCESS PROBLEM-DEPENDENT CROSS SECTIONS AND NEUTRON SPECTRAL DATA FOR ORIGEN-S ANALYSES	F6.1.1
Section F7	ORIGEN-S: SCALE SYSTEM MODULE TO CALCULATE FUEL DEPLETION, ACTINIDE TRANSMUTATION, FISSION PRODUCT BUILDUP AND DECAY, AND ASSOCIATED RADIATION SOURCE TERMS	F7.1.1
Section F8	ICE-S: MODULE TO MIX MULTIGROUP CROSS SECTIONS	F8.1.1
Section F9	MORSE-SGC/S: A NEUTRON AND PHOTON MONTE CARLO TRANSPORT ANALYSIS WITH SUPER-GROUPING AND ADVANCED GEOMETRY FEATURES	F9.1.1
Section F10	HEATING6: A MULTIDIMENSIONAL HEAT CONDUCTION ANALYSIS WITH THE FINITE-DIFFERENCE FORMULATION	F10.1.1
Section F11	KENO-V.a: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM WITH SUPER-GROUPING	F11.1.1
Section F12	JUNEBUG-II: A THREE-DIMENSIONAL GEOMETRY PLOTTING CODE	F12.1.1
Section F13	HEATPLOT-S: A TEMPERATURE DISTRIBUTION PLOTTING PROGRAM FOR HEATING5/HEATING6.	F13.1.1
Section F14	REGPLOT6: A PLOTTING PROGRAM TO VERIFY HEATING6 INPUT DATA	F14.1.1
Section F15	PLORIGEN: A PLOTTING PROGRAM FOR ORIGEN-S OUTPUT (to be issued at later date)	F15.1.1

VOLUME 3: MISCELLANEOUS

		Page
Section M1	THE SCALE SYSTEM DRIVER	M1.1.1
Section M2	THE SCALE SYSTEM SUBROUTINE LIBRARY	M2.1.1
Section M3	SCALE FREE FORM READING ROUTINES	M3.1.1
Section M4	NEUTRON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M4.1.1
Section M5	NEUTRON-PHOTON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M5.1.1
Section M6	ORIGEN-S DATA LIBRARIES	M6.1.1
Section M7	MATERIAL PROPERTIES LIBRARIES . . . . . (to be issued at later date)	M7.1.1
Section M8	STANDARD COMPOSITION LIBRARY	M8.1.1
Section M9	MARS: A MULTIPLE ARRAY SYSTEM USING COMBINATORIAL GEOMETRY	M9.1.1
Section M10	FIDO INPUT SYSTEM	M10.1.1
Section M11	SCALE INTERACTIVE INPUT PROCESSOR	M11.1.1
Section M12	CESAR: A CRITICALITY EXPERIMENT STORAGE AND RETRIEVAL PROGRAM.	M12.1.1
Section M13	PICTURE: A PRINTER PLOT PACKAGE FOR MAKING 2-D PICTURES OF MARS GEOMETRIES.	M13.1.1
Section M14	COMPOZ DATA GUIDE.	M14.1.1

FOREWORD FOR SCALE MANUAL

This manual describes a system of computer codes for performing Standardized Computer Analyses for Licensing Evaluation (SCALE). The SCALE system was conceived and funded by the U. S. Nuclear Regulatory Commission to satisfy a strong need for a standardized method of analysis for the evaluation of nuclear fuel facility and package designs. Thus, the SCALE system will lead to a common understanding on how these designs will be evaluated by the NRC.

The overall goal of the SCALE project has been to develop easy-to-use analytical sequences which are automated to perform the necessary data processing (e.g., cross-section preparation) and manipulation of well-established computer codes (functional modules) required by the sequence. Thus, the user is able to select an analytical sequence characterized by the type of analysis (criticality, shielding, or heat transfer) to be performed and the geometric complexity of the system being analyzed. The user then prepares a single set of input for the control module corresponding to this analytical sequence. The control module input is in terms of easily visualized engineering parameters specified in a simplified, free-form format. The control modules use this information to derive additional parameters and prepare the input for each of the functional modules in the analytical sequence. Provisions have also been made to allow the user to execute the functional modules on a stand-alone basis. Stand-alone execution of the functional modules enables the SCALE system to be used for a wide range of nuclear and thermal applications.

In developing the SCALE system, an attempt has been made to adhere to certain criteria that would make SCALE an attractive computational tool for both the novice and experienced analyst. Some of the criteria are to:

a) develop a system driver which minimizes the computer core and running time,

b) develop control modules which require only a simplified set of input,

c) develop control modules which automate and couple the data processing and systems analysis into standard analytical sequences,

d) provide self-contained and thorough documentation for the entire system,

e) use well-established functional modules and data libraries that have a wide range of applicability and a proven record of reliable performance.

The overall objective of following these criteria was to develop SCALE as a system leading to analytical standardization.

The control modules available with SCALE are documented in Volume 1 of this manual. The SCALE system contains a wide assortment of control modules for performing criticality, shielding, and heat transfer analysis. Criticality Safety Analysis Sequences are referred to as CSAS1-CSASn and are described in Sections C1-Cn. Likewise, the shielding and heat transfer analysis sequences are called SAS1-SASn and HTAS1-HTASn, respectively and are described in Sections S1-Sn and H1-Hn, respectively.

Volume 2 of this manual provides documentation for each of the well-established functional modules used within the SCALE system. These sections are referred to as Sections F1-Fn. The functional modules include codes for data and/or cross section preparation (e.g. BONAMI-S and NITAWL-S), criticality analysis (e.g. XSDRNPM-S, KENO-IV/S, KENO-V), shielding analysis (e.g., XSDRNPM-S and MORSE-SGC/S), heat transfer analysis (HEATING6), and depletion analysis (ORIGEN-S). The "S" attached to the name of some functional modules indicates that the code was originally developed outside of SCALE and modified for use in the SCALE system. The exception to this is the HEATING6 code which is an upgrade of HEATING5.

Volume 3 of this manual provides documentation for the miscellaneous (but important) components of the SCALE system. These sections are referred to as Sections M1-Mn and include descriptions of the system driver, the subroutine library, the input systems, and the data, material composition, and property libraries. The data libraries include a variety of well-established cross section libraries, as well as some new libraries developed within the SCALE project.

The SCALE Manual is being released in a modular fashion; i.e., as each new component of the SCALE system is developed and released, a new section will be added to the SCALE Manual. Also, the documentation is released in looseleaf form so that as corrections and small revisions are made to the various modules, new or revised pages can be easily issued for the Manual. The List of Revisions issued with each release of new documentation provides a list of the NRC publication dates for each new or revised section and/or pages.

The SCALE system is being packaged and distributed by the Radiation Shielding Information Center (RSIC), located at Oak Ridge National Laboratory. A limited version of SCALE (SCALE-0) was made available to RSIC in July 1980. Significant revisions and additions have been made to SCALE and new versions were released as SCALE-1 in 1982 and SCALE-2 in 1983. This latest issuance of documentation should upgrade the existing documentation to match the SCALE-3 version of the system presently being issued by RSIC. Only IBM versions of SCALE are currently available from RSIC with the exception of SCALE-0 for which a CDC version is available. Requests for the latest SCALE code package can be mailed to:

Radiation Shielding Information Center
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

or telephoned to

Area Code 615; 574-6176 or to FTS 624-6176.

C. V. Parks
SCALE Project Leader
Oak Ridge National Laboratory

LIST OF REVISIONS for NUREG/CR-0200,
SCALE: A Modular Code System for Performing Standardized
Computer Analyses for Licensing Evaluation

Page/Section	New/Revised/Corrected*	Date
Table of Contents	Revised	December 1984
Foreword	Revised	December 1984
Sect. C1	Revised	January 1982
p. C1.A.5	Corrected	December 1984
Sect. C2	New	January 1982
Sect. C4	New	January 1982
title page	Revised	December 1984
p. C4.iii-vii	Revised	December 1984
Sect. C4.4	Revised	December 1984
Sect. S2	New	June 1983
title page	Revised	December 1984
p. S2.2.9	Revised	December 1984
p. S2.4.7	Revised	December 1984
p. S2.5.5	Revised	December 1984
Sect. S3	New	January 1982
p. S3.A.1	Revised	June 1983
Sect. H1	New	June 1983
title page	Revised	December 1984
p. H1.iii-xi	Revised	December 1984
p. H1.xiii	Deleted	December 1984
p. H1.2.15	Revised	December 1984
p. H1.2.20	Revised	December 1984
p. H1.2.21	New	December 1984
p. H1.3.4	Revised	December 1984
p. H1.3.6	Revised	December 1984
p. H1.3.11-14	Revised	December 1984
p. H1.4.3	Revised	December 1984
p. H1.4.15	Revised	December 1984
p. H1.4.17-18	Revised	December 1984
p. H1.4.20	Revised	December 1984
p. H1.4.25-26	Revised	December 1984
p. H1.4.30	Revised	December 1984
p. H1.4.31-32	Revised	December 1984
Sect. H1.5	Revised	December 1984
Sect. H1.6	Revised	December 1984
Sect. H1.A	Revised	December 1984
Sect. H1.B	Revised	December 1984
Sect. H1.C	Deleted	December 1984
Sect. F1	New	January 1982
p. F1.2.2	Corrected	June 1983
p. F1.2.5	Corrected	June 1983
p. F1.5.1	Corrected	June 1983
p. F1.5.2	Corrected	December 1984
p. F1.6.2	Corrected	July 1982
p. F1.6.11	Corrected	July 1982

Page/Section	New/Revised/Corrected*	Date
Sect. F2	New	January 1982
p. F2.2.3	Corrected	June 1983
p. F2.2.15	Revised	June 1983
pp.F2.3.33-34	Corrected	July 1982
p. F2.3.41	Corrected	July 1982
p. F2.4.4	Corrected	December 1984
Sect. F3	New	June 1983
p. F3.12.6	Corrected	December 1984
Sect. F4	New	June 1983
Sect. F5	New	January 1982
p. F5.6.2	Corrected	June 1983
p. F5.B.3	Corrected	December 1984
Sect. F6	New	January 1982
title page	Revised	December 1984
p. F6.4.6	Revised	December 1984
p. F6.5.14	Revised	December 1984
Sect. F7	New	January 1982
title page	Revised	December 1984
p. F7.iii	Revised	December 1984
p. F7.v-vi	Revised	June 1983
p. F7.ix	Revised	December 1984
p. F7.2.15	Revised	June 1983
p. F7.2.27-28	New	December 1984
p. F7.3.3-4	Revised	December 1984
p. F7.4.2	Revised	December 1984
p. F7.5.11	Revised	December 1984
p. F7.6.2	Revised	June 1983
p. F7.6.4	Revised	December 1984
p. F7.6.6	Revised	June 1983
p. F7.6.8	Revised	December 1984
p. F7.6.10-11	Revised	December 1984
p. F7.6.15-19	Revised	December 1984
p. F7.6.21	Revised	June 1983
p. F7.6.22-23	Revised	December 1984
p. F7.6.25-26	Revised	June 1983
p. F7.6.27-28	Revised	December 1984
p. F7.6.30	Revised	December 1984
p. F7.6.33-34	Revised	June 1983
p. F7.6.37-38	Revised	December 1984
p. F7.6.39-40	Revised	June 1983
p. F7.6.43	Revised	June 1983
p. F7.6.44	Revised	December 1984
p. F7.7.1	Revised	December 1984
p. F7.8.1	Revised	December 1984
p. F7.9.2	Revised	December 1984
Sect. F8	New	June 1983

Page/Section	New/Revised/Corrected*	Date
Sect. F9	New	January 1982
title page	Revised	December 1984
p. F9.A.1	Revised	December 1984
p. F9.A.12	Revised	December 1984
p. F9.A.13	Revised	December 1984
p. F9.B.6	Corrected	December 1984
p. F9.B.7-9	Revised	December 1984
p. F9.B.10	Deleted	December 1984
Sect. F10	New	January 1982
p. F10.iv	Revised	June 1983
p. F10.iv	Corrected	December 1984
p. F10.2.23	Corrected	December 1984
p. F10.3.38	Revised	June 1983
p. F10.5.9-12	Revised	June 1983
p. F10.5.9	Corrected	December 1984
p. F10.5.14	Corrected	December 1984
p. F10.5.35	Revised	June 1983
p. F10.5.38	Corrected	December 1984
p. F10.A.1-2	Corrected	December 1984
p. F10.A.4-5	Corrected	December 1984
p. F10.C.36	Revised	June 1983
Sect. F11	Revised	December 1984
Sect. F12	Revised	December 1984
Sect. F13	New	December 1984
Sect. F14	New	December 1984
Sect. M1	New	January 1982
Sect. M2	New	January 1982
p. M2.2.58	New	June 1983
p. M2.2.73	Revised	December 1984
Sect. M3	New	January 1982
Sect. M6	New	June 1983
title page	Revised	December 1984
p. M6.v	Revised	December 1984
p. M6.vii	Revised	December 1984
p. M6.4.1-2	Revised	December 1984
p. M6.7.4	Revised	December 1984
p. M6.7.6-10	Revised	December 1984
p. M6.8.3	Revised	December 1984
Sect. M8	Revised	June 1983
title page	Revised	December 1984
p. M8.2.2-13	Revised	December 1984
p. M8.4.1	Revised	December 1984

Page/Section	New/Revised/Corrected*	Date
Sect. M9	New	January 1982
title page	Revised	December 1984
p. M9.iv	Revised	December 1984
p. M9.vii	Revised	December 1984
p. M9.6.5-6	Revised	December 1984
p. M9.E.2-3	Corrected	December 1984
p. M9.E.4-6	Revised	December 1984
p. M9.E.14-15	Revised	December 1984
p. M9.E.16-19	New	December 1984
Sect. M9.G	New	December 1984
Sect. M10	New	January 1982
Sect. M11	New	January 1982
Sect. M12	New	December 1984
Sect. M13	New	December 1984
Sect. M14	New	December 1984

* New = First issuance of document section or page
Revised = Revision of document section or pages to reflect code updates or changes
Corrected = Correction to reflect error in previous issuance of document

TABLE OF CONTENTS
FOR SCALE MANUAL

		Page*
ACKNOWLEDGMENTS		vii
ABSTRACT		ix
FOREWORD		xi
LIST OF REVISIONS		xv
	VOLUME 1: CONTROL MODULES	
Section C1	CSAS1: A ONE-DIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C1.1.1
Section C2	CSAS2: A MULTIDIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C2.1.1
Section C3	CSAS3: AN OPTIMUM CONCENTRATION CRITICALITY SAFETY ANALYSIS MODULE (to be issued at later date)	C3.1.1
Section C4	CSAS4: AN ENHANCED CRITICALITY SAFETY ANALYSIS MODULE WITH AN OPTIMUM PITCH SEARCH OPTION	C4.1.1
Section S1	SAS1: A ONE-DIMENSIONAL SHIELDING ANALYSIS MODULE (to be issued at later date)	S1.1.1
Section S2	SAS2: A COUPLED ONE-DIMENSIONAL DEPLETION AND SHIELDING ANALYSIS MODULE	S2.1.1
Section S3	SAS3: AN AUTOMATED MONTE CARLO SHIELDING ANALYSIS MODULE	S3.1.1
Section S4	SAS4: A MONTE CARLO SHIELDING ANALYSIS MODULE WITH AUTOMATIC BIASING. (to be issued at later date)	S4.1.1
Section H1	HTAS1: A TWO-DIMENSIONAL HEAT TRANSFER ANALYSIS OF FUEL CASKS	H1.1.1

*All section page numbers refer to the page where the section introduction begins.

VOLUME 2: FUNCTIONAL MODULES

		Page
Section F1	BONAMI-S: RESONANCE SELF-SHIELDING BY THE BONDARENKO METHOD	F1.1.1
Section F2	NITAWL-S: RESONANCE SELF-SHIELDING BY THE NORDHEIM METHOD	F2.1.1
Section F3	XSDRNPM-S: A ONE-DIMENSIONAL DISCRETE-ORDINATES PROGRAM FOR TRANSPORT ANALYSIS . . .	F3.1.1
Section F4	XSDOSE: A MODULE FOR CALCULATING FLUXES AND DOSE RATES AT POINTS OUTSIDE A SHIELD	F4.1.1
Section F5	KENO-IV/S: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM	F5.1.1
Section F6	COUPLE: SCALE SYSTEM MODULE TO PROCESS PROBLEM-DEPENDENT CROSS SECTIONS AND NEUTRON SPECTRAL DATA FOR ORIGEN-S ANALYSES	F6.1.1
Section F7	ORIGEN-S: SCALE SYSTEM MODULE TO CALCULATE FUEL DEPLETION, ACTINIDE TRANSMUTATION, FISSION PRODUCT BUILDUP AND DECAY, AND ASSOCIATED RADIATION SOURCE TERMS	F7.1.1
Section F8	ICE-S: MODULE TO MIX MULTIGROUP CROSS SECTIONS	F8.1.1
Section F9	MORSE-SGC/S: A NEUTRON AND PHOTON MONTE CARLO TRANSPORT ANALYSIS WITH SUPER-GROUPING AND ADVANCED GEOMETRY FEATURES	F9.1.1
Section F10	HEATING6: A MULTIDIMENSIONAL HEAT CONDUCTION ANALYSIS WITH THE FINITE-DIFFERENCE FORMULATION	F10.1.1
Section F11	KENO-V.a: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM WITH SUPER-GROUPING	F11.1.1
Section F12	JUNEBUG-II: A THREE-DIMENSIONAL GEOMETRY PLOTTING CODE	F12.1.1
Section F13	HEATPLOT-S: A TEMPERATURE DISTRIBUTION PLOTTING PROGRAM FOR HEATING5/HEATING6.	F13.1.1
Section F14	REGPLOT6: A PLOTTING PROGRAM TO VERIFY HEATING6 INPUT DATA	F14.1.1
Section F15	PLORIGEN: A PLOTTING PROGRAM FOR ORIGEN-S OUTPUT (to be issued at later date)	F15.1.1

VOLUME 3: MISCELLANEOUS

		Page
Section M1	THE SCALE SYSTEM DRIVER	M1.1.1
Section M2	THE SCALE SYSTEM SUBROUTINE LIBRARY	M2.1.1
Section M3	SCALE FREE FORM READING ROUTINES	M3.1.1
Section M4	NEUTRON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M4.1.1
Section M5	NEUTRON-PHOTON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M5.1.1
Section M6	ORIGEN-S DATA LIBRARIES	M6.1.1
Section M7	MATERIAL PROPERTIES LIBRARIES . . . . . (to be issued at later date)	M7.1.1
Section M8	STANDARD COMPOSITION LIBRARY	M8.1.1
Section M9	MARS: A MULTIPLE ARRAY SYSTEM USING COMBINATORIAL GEOMETRY	M9.1.1
Section M10	FIDO INPUT SYSTEM	M10.1.1
Section M11	SCALE INTERACTIVE INPUT PROCESSOR	M11.1.1
Section M12	CESAR: A CRITICALITY EXPERIMENT STORAGE AND RETRIEVAL PROGRAM.	M12.1.1
Section M13	PICTURE: A PRINTER PLOT PACKAGE FOR MAKING 2-D PICTURES OF MARS GEOMETRIES.	M13.1.1
Section M14	COMPOZ DATA GUIDE.	M14.1.1

FOREWORD FOR SCALE MANUAL

This manual describes a system of computer codes for performing Standardized Computer Analyses for Licensing Evaluation (SCALE). The SCALE system was conceived and funded by the U. S. Nuclear Regulatory Commission to satisfy a strong need for a standardized method of analysis for the evaluation of nuclear fuel facility and package designs. Thus, the SCALE system will lead to a common understanding on how these designs will be evaluated by the NRC.

The overall goal of the SCALE project has been to develop easy-to-use analytical sequences which are automated to perform the necessary data processing (e.g., cross-section preparation) and manipulation of well-established computer codes (functional modules) required by the sequence. Thus, the user is able to select an analytical sequence characterized by the type of analysis (criticality, shielding, or heat transfer) to be performed and the geometric complexity of the system being analyzed. The user then prepares a single set of input for the control module corresponding to this analytical sequence. The control module input is in terms of easily visualized engineering parameters specified in a simplified, free-form format. The control modules use this information to derive additional parameters and prepare the input for each of the functional modules in the analytical sequence. Provisions have also been made to allow the user to execute the functional modules on a stand-alone basis. Stand-alone execution of the functional modules enables the SCALE system to be used for a wide range of nuclear and thermal applications.

In developing the SCALE system, an attempt has been made to adhere to certain criteria that would make SCALE an attractive computational tool for both the novice and experienced analyst. Some of the criteria are to:

a) develop a system driver which minimizes the computer core and running time,

b) develop control modules which require only a simplified set of input,

c) develop control modules which automate and couple the data processing and systems analysis into standard analytical sequences,

d) provide self-contained and thorough documentation for the entire system,

e) use well-established functional modules and data libraries that have a wide range of applicability and a proven record of reliable performance.

The overall objective of following these criteria was to develop SCALE as a system leading to analytical standardization.

The control modules available with SCALE are documented in Volume 1 of this manual. The SCALE system contains a wide assortment of control modules for performing criticality, shielding, and heat transfer analysis. Criticality Safety Analysis Sequences are referred to as CSAS1-CSASn and are described in Sections C1-Cn. Likewise, the shielding and heat transfer analysis sequences are called SAS1-SASn and HTAS1-HTASn, respectively and are described in Sections S1-Sn and H1-Hn, respectively.

Volume 2 of this manual provides documentation for each of the well-established functional modules used within the SCALE system. These sections are referred to as Sections F1-Fn. The functional modules include codes for data and/or cross section preparation (e.g. BONAMI-S and NITAWL-S), criticality analysis (e.g. XSDRNPM-S, KENO-IV/S, KENO-V), shielding analysis (e.g., XSDRNPM-S and MORSE-SGC/S), heat transfer analysis (HEATING6), and depletion analysis (ORIGEN-S). The "S" attached to the name of some functional modules indicates that the code was originally developed outside of SCALE and modified for use in the SCALE system. The exception to this is the HEATING6 code which is an upgrade of HEATING5

Volume 3 of this manual provides documentation for the miscellaneous (but important) components of the SCALE system. These sections are referred to as Sections M1-Mn and include descriptions of the system driver, the subroutine library, the input systems, and the data, material composition, and property libraries. The data libraries include a variety of well-established cross section libraries, as well as some new libraries developed within the SCALE project.

The SCALE Manual is being released in a modular fashion; i.e., as each new component of the SCALE system is developed and released, a new section will be added to the SCALE Manual. Also, the documentation is released in looseleaf form so that as corrections and small revisions are made to the various modules, new or revised pages can be easily issued for the Manual. The List of Revisions issued with each release of new documentation provides a list of the NRC publication dates for each new or revised section and/or pages.

The SCALE system is being packaged and distributed by the Radiation Shielding Information Center (RSIC), located at Oak Ridge National Laboratory. A limited version of SCALE (SCALE-0) was made available to RSIC in July 1980. Significant revisions and additions have been made to SCALE and new versions were released as SCALE-1 in 1982 and SCALE-2 in 1983. This latest issuance of documentation should upgrade the existing documentation to match the SCALE-3 version of the system presently being issued by RSIC. Only IBM versions of SCALE are currently available from RSIC with the exception of SCALE-0 for which a CDC version is available. Requests for the latest SCALE code package can be mailed to:

Radiation Shielding Information Center
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

or telephoned to

Area Code 615; 574-6176 or to FTS 624-6176.

C. V. Parks
SCALE Project Leader
Oak Ridge National Laboratory

LIST OF REVISIONS for NUREG/CR-0200,
SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

Page/Section	New/Revised/Corrected*	Date
Table of Contents	Revised	December 1984
Foreword	Revised	December 1984
Sect. C1	Revised	January 1982
p. C1.A.5	Corrected	December 1984
Sect. C2	New	January 1982
Sect. C4	New	January 1982
title page	Revised	December 1984
p. C4.iii-vii	Revised	December 1984
Sect. C4.4	Revised	December 1984
Sect. S2	New	June 1983
title page	Revised	December 1984
p. S2.2.9	Revised	December 1984
p. S2.4.7	Revised	December 1984
p. S2.5.5	Revised	December 1984
Sect. S3	New	January 1982
p. S3.A.1	Revised	June 1983
Sect. H1	New	June 1983
title page	Revised	December 1984
p. H1.iii-xi	Revised	December 1984
p. H1.xiii	Deleted	December 1984
p. H1.2.15	Revised	December 1984
p. H1.2.20	Revised	December 1984
p. H1.2.21	New	December 1984
p. H1.3.4	Revised	December 1984
p. H1.3.6	Revised	December 1984
p. H1.3.11-14	Revised	December 1984
p. H1.4.3	Revised	December 1984
p. H1.4.15	Revised	December 1984
p. H1.4.17-18	Revised	December 1984
p. H1.4.20	Revised	December 1984
p. H1.4.25-26	Revised	December 1984
p. H1.4.30	Revised	December 1984
p. H1.4.31-32	Revised	December 1984
Sect. H1.5	Revised	December 1984
Sect. H1.6	Revised	December 1984
Sect. H1.A	Revised	December 1984
Sect. H1.B	Revised	December 1984
Sect. H1.C	Deleted	December 1984
Sect. F1	New	January 1982
p. F1.2.2	Corrected	June 1983
p. F1.2.5	Corrected	June 1983
p. F1.5.1	Corrected	June 1983
p. F1.5.2	Corrected	December 1984
p. F1.6.2	Corrected	July 1982
p. F1.6.11	Corrected	July 1982

Page/Section	New/Revised/Corrected*	Date
Sect. F2	New	January 1982
p. F2.2.3	Corrected	June 1983
p. F2.2.15	Revised	June 1983
pp.F2.3.33-34	Corrected	July 1982
p. F2.3.41	Corrected	July 1982
p. F2.4.4	Corrected	December 1984
Sect. F3	New	June 1983
p. F3.12.6	Corrected	December 1984
Sect. F4	New	June 1983
Sect. F5	New	January 1982
p. F5.6.2	Corrected	June 1983
p. F5.B.3	Corrected	December 1984
Sect. F6	New	January 1982
title page	Revised	December 1984
p. F6.4.6	Revised	December 1984
p. F6.5.14	Revised	December 1984
Sect. F7	New	January 1982
title page	Revised	December 1984
p. F7.iii	Revised	December 1984
p. F7.v-vi	Revised	June 1983
p. F7.ix	Revised	December 1984
p. F7.2.15	Revised	June 1983
p. F7.2.27-28	New	December 1984
p. F7.3.3-4	Revised	December 1984
p. F7.4.2	Revised	December 1984
p. F7.5.11	Revised	December 1984
p. F7.6.2	Revised	June 1983
p. F7.6.4	Revised	December 1984
p. F7.6.6	Revised	June 1983
p. F7.6.8	Revised	December 1984
p. F7.6.10-11	Revised	December 1984
p. F7.6.15-19	Revised	December 1984
p. F7.6.21	Revised	June 1983
p. F7.6.22-23	Revised	December 1984
p. F7.6.25-26	Revised	June 1983
p. F7.6.27-28	Revised	December 1984
p. F7.6.30	Revised	December 1984
p. F7.6.33-34	Revised	June 1983
p. F7.6.37-38	Revised	December 1984
p. F7.6.39-40	Revised	June 1983
p. F7.6.43	Revised	June 1983
p. F7.6.44	Revised	December 1984
p. F7.7.1	Revised	December 1984
p. F7.8.1	Revised	December 1984
p. F7.9.2	Revised	December 1984
Sect. F8	New	June 1983

Page/Section	New/Revised/Corrected*	Date
Sect. F9	New	January 1982
title page	Revised	December 1984
p. F9.A.1	Revised	December 1984
p. F9.A.12	Revised	December 1984
p. F9.A.13	Revised	December 1984
p. F9.B.6	Corrected	December 1984
p. F9.B.7-9	Revised	December 1984
p. F9.B.10	Deleted	December 1984
Sect. F10	New	January 1982
p. F10.iv	Revised	June 1983
p. F10.iv	Corrected	December 1984
p. F10.2.23	Corrected	December 1984
p. F10.3.38	Revised	June 1983
p. F10.5.9-12	Revised	June 1983
p. F10.5.9	Corrected	December 1984
p. F10.5.14	Corrected	December 1984
p. F10.5.35	Revised	June 1983
p. F10.5.38	Corrected	December 1984
p. F10.A.1-2	Corrected	December 1984
p. F10.A.4-5	Corrected	December 1984
p. F10.C.36	Revised	June 1983
Sect. F11	Revised	December 1984
Sect. F12	Revised	December 1984
Sect. F13	New	December 1984
Sect. F14	New	December 1984
Sect. M1	New	January 1982
Sect. M2	New	January 1982
p. M2.2.58	New	June 1983
p. M2.2.73	Revised	December 1984
Sect. M3	New	January 1982
Sect. M6	New	June 1983
title page	Revised	December 1984
p. M6.v	Revised	December 1984
p. M6.vii	Revised	December 1984
p. M6.4.1-2	Revised	December 1984
p. M6.7.4	Revised	December 1984
p. M6.7.6-10	Revised	December 1984
p. M6.8.3	Revised	December 1984
Sect. M8	Revised	June 1983
title page	Revised	December 1984
p. M8.2.2-13	Revised	December 1984
p. M8.4.1	Revised	December 1984

Page/Section	New/Revised/Corrected*	Date
Sect. M9	New	January 1982
title page	Revised	December 1984
p. M9.iv	Revised	December 1984
p. M9.vii	Revised	December 1984
p. M9.6.5-6	Revised	December 1984
p. M9.E.2-3	Corrected	December 1984
p. M9.E.4-6	Revised	December 1984
p. M9.E.14-15	Revised	December 1984
p. M9.E.16-19	New	December 1984
Sect. M9.G	New	December 1984
Sect. M10	New	January 1982
Sect. M11	New	January 1982
Sect. M12	New	December 1984
Sect. M13	New	December 1984
Sect. M14	New	December 1984

* New = First issuance of document section or page
Revised = Revision of document section or pages to reflect code updates or changes
Corrected = Correction to reflect error in previous issuance of document

TABLE OF CONTENTS
FOR SCALE MANUAL

		Page*
ACKNOWLEDGMENTS		vii
ABSTRACT		ix
FOREWORD		xi
LIST OF REVISIONS		xv
	VOLUME 1: CONTROL MODULES	
Section C1	CSAS1: A ONE-DIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C1.1.1
Section C2	CSAS2: A MULTIDIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C2.1.1
Section C3	CSAS3: AN OPTIMUM CONCENTRATION CRITICALITY SAFETY ANALYSIS MODULE (to be issued at later date)	C3.1.1
Section C4	CSAS4: AN ENHANCED CRITICALITY SAFETY ANALYSIS MODULE WITH AN OPTIMUM PITCH SEARCH OPTION	C4.1.1
Section S1	SAS1: A ONE-DIMENSIONAL SHIELDING ANALYSIS MODULE (to be issued at later date)	S1.1.1
Section S2	SAS2: A COUPLED ONE-DIMENSIONAL DEPLETION AND SHIELDING ANALYSIS MODULE	S2.1.1
Section S3	SAS3: AN AUTOMATED MONTE CARLO SHIELDING ANALYSIS MODULE	S3.1.1
Section S4	SAS4: A MONTE CARLO SHIELDING ANALYSIS MODULE WITH AUTOMATIC BIASING. (to be issued at later date)	S4.1.1
Section H1	HTAS1: A TWO-DIMENSIONAL HEAT TRANSFER ANALYSIS OF FUEL CASKS	H1.1.1

*All section page numbers refer to the page where the section introduction begins.

VOLUME 2: FUNCTIONAL MODULES

		Page
Section F1	BONAMI-S: RESONANCE SELF-SHIELDING BY THE BONDARENKO METHOD	F1.1.1
Section F2	NITAWL-S: RESONANCE SELF-SHIELDING BY THE NORDHEIM METHOD	F2.1.1
Section F3	XSDRNPM-S: A ONE-DIMENSIONAL DISCRETE-ORDINATES PROGRAM FOR TRANSPORT ANALYSIS . . .	F3.1.1
Section F4	XSDOSE: A MODULE FOR CALCULATING FLUXES AND DOSE RATES AT POINTS OUTSIDE A SHIELD	F4.1.1
Section F5	KENO-IV/S: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM	F5.1.1
Section F6	COUPLE: SCALE SYSTEM MODULE TO PROCESS PROBLEM-DEPENDENT CROSS SECTIONS AND NEUTRON SPECTRAL DATA FOR ORIGEN-S ANALYSES	F6.1.1
Section F7	ORIGEN-S: SCALE SYSTEM MODULE TO CALCULATE FUEL DEPLETION, ACTINIDE TRANSMUTATION, FISSION PRODUCT BUILDUP AND DECAY, AND ASSOCIATED RADIATION SOURCE TERMS	F7.1.1
Section F8	ICE-S: MODULE TO MIX MULTIGROUP CROSS SECTIONS	F8.1.1
Section F9	MORSE-SGC/S: A NEUTRON AND PHOTON MONTE CARLO TRANSPORT ANALYSIS WITH SUPER-GROUPING AND ADVANCED GEOMETRY FEATURES	F9.1.1
Section F10	HEATING6: A MULTIDIMENSIONAL HEAT CONDUCTION ANALYSIS WITH THE FINITE-DIFFERENCE FORMULATION	F10.1.1
Section F11	KENO-V.a: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM WITH SUPER-GROUPING	F11.1.1
Section F12	JUNEBUG-II: A THREE-DIMENSIONAL GEOMETRY PLOTTING CODE	F12.1.1
Section F13	HEATPLOT-S: A TEMPERATURE DISTRIBUTION PLOTTING PROGRAM FOR HEATING5/HEATING6	F13.1.1
Section F14	REGPLOT6: A PLOTTING PROGRAM TO VERIFY HEATING6 INPUT DATA	F14.1.1
Section F15	PLORIGEN: A PLOTTING PROGRAM FOR ORIGEN-S OUTPUT (to be issued at later date)	F15.1.1

VOLUME 3: MISCELLANEOUS

		Page
Section M1	THE SCALE SYSTEM DRIVER	M1.1.1
Section M2	THE SCALE SYSTEM SUBROUTINE LIBRARY	M2.1.1
Section M3	SCALE FREE FORM READING ROUTINES	M3.1.1
Section M4	NEUTRON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M4.1.1
Section M5	NEUTRON-PHOTON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M5.1.1
Section M6	ORIGEN-S DATA LIBRARIES	M6.1.1
Section M7	MATERIAL PROPERTIES LIBRARIES . . . . . (to be issued at later date)	M7.1.1
Section M8	STANDARD COMPOSITION LIBRARY	M8.1.1
Section M9	MARS: A MULTIPLE ARRAY SYSTEM USING COMBINATORIAL GEOMETRY	M9.1.1
Section M10	FIDO INPUT SYSTEM	M10.1.1
Section M11	SCALE INTERACTIVE INPUT PROCESSOR	M11.1.1
Section M12	CESAR: A CRITICALITY EXPERIMENT STORAGE AND RETRIEVAL PROGRAM.	M12.1.1
Section M13	PICTURE: A PRINTER PLOT PACKAGE FOR MAKING 2-D PICTURES OF MARS GEOMETRIES.	M13.1.1
Section M14	COMPOZ DATA GUIDE.	M14.1.1

FOREWORD FOR SCALE MANUAL

This manual describes a system of computer codes for performing Standardized Computer Analyses for Licensing Evaluation (SCALE). The SCALE system was conceived and funded by the U. S. Nuclear Regulatory Commission to satisfy a strong need for a standardized method of analysis for the evaluation of nuclear fuel facility and package designs. Thus, the SCALE system will lead to a common understanding on how these designs will be evaluated by the NRC.

The overall goal of the SCALE project has been to develop easy-to-use analytical sequences which are automated to perform the necessary data processing (e.g., cross-section preparation) and manipulation of well-established computer codes (functional modules) required by the sequence. Thus, the user is able to select an analytical sequence characterized by the type of analysis (criticality, shielding, or heat transfer) to be performed and the geometric complexity of the system being analyzed. The user then prepares a single set of input for the control module corresponding to this analytical sequence. The control module input is in terms of easily visualized engineering parameters specified in a simplified, free-form format. The control modules use this information to derive additional parameters and prepare the input for each of the functional modules in the analytical sequence. Provisions have also been made to allow the user to execute the functional modules on a stand-alone basis. Stand-alone execution of the functional modules enables the SCALE system to be used for a wide range of nuclear and thermal applications.

In developing the SCALE system, an attempt has been made to adhere to certain criteria that would make SCALE an attractive computational tool for both the novice and experienced analyst. Some of the criteria are to:

a) develop a system driver which minimizes the computer core and running time,

b) develop control modules which require only a simplified set of input,

c) develop control modules which automate and couple the data processing and systems analysis into standard analytical sequences,

d) provide self-contained and thorough documentation for the entire system,

e) use well-established functional modules and data libraries that have a wide range of applicability and a proven record of reliable performance.

The overall objective of following these criteria was to develop SCALE as a system leading to analytical standardization.

The control modules available with SCALE are documented in Volume 1 of this manual. The SCALE system contains a wide assortment of control modules for performing criticality, shielding, and heat transfer analysis. Criticality Safety Analysis Sequences are referred to as CSAS1-CSASn and are described in Sections C1-Cn. Likewise, the shielding and heat transfer analysis sequences are called SAS1-SASn and HTAS1-HTASn, respectively and are described in Sections S1-Sn and H1-Hn, respectively.

Volume 2 of this manual provides documentation for each of the well-established functional modules used within the SCALE system. These sections are referred to as Sections F1-Fn. The functional modules include codes for data and/or cross section preparation (e.g. BONAMI-S and NITAWL-S), criticality analysis (e.g. XSDRNPM-S, KENO-IV/S, KENO-V), shielding analysis (e.g., XSDRNPM-S and MORSE-SGC/S), heat transfer analysis (HEATING6), and depletion analysis (ORIGEN-S). The "S" attached to the name of some functional modules indicates that the code was originally developed outside of SCALE and modified for use in the SCALE system. The exception to this is the HEATING6 code which is an upgrade of HEATING5.

Volume 3 of this manual provides documentation for the miscellaneous (but important) components of the SCALE system. These sections are referred to as Sections M1-Mn and include descriptions of the system driver, the subroutine library, the input systems, and the data, material composition, and property libraries. The data libraries include a variety of well-established cross section libraries, as well as some new libraries developed within the SCALE project.

The SCALE Manual is being released in a modular fashion; i.e., as each new component of the SCALE system is developed and released, a new section will be added to the SCALE Manual. Also, the documentation is released in looseleaf form so that as corrections and small revisions are made to the various modules, new or revised pages can be easily issued for the Manual. The List of Revisions issued with each release of new documentation provides a list of the NRC publication dates for each new or revised section and/or pages.

The SCALE system is being packaged and distributed by the Radiation Shielding Information Center (RSIC), located at Oak Ridge National Laboratory. A limited version of SCALE (SCALE-0) was made available to RSIC in July 1980. Significant revisions and additions have been made to SCALE and new versions were released as SCALE-1 in 1982 and SCALE-2 in 1983. This latest issuance of documentation should upgrade the existing documentation to match the SCALE-3 version of the system presently being issued by RSIC. Only IBM versions of SCALE are currently available from RSIC with the exception of SCALE-0 for which a CDC version is available. Requests for the latest SCALE code package can be mailed to:

Radiation Shielding Information Center
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

or telephoned to

Area Code 615; 574-6176 or to FTS 624-6176.

C. V. Parks
SCALE Project Leader
Oak Ridge National Laboratory

LIST OF REVISIONS for NUREG/CR-0200,
SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

Page/Section	New/Revised/Corrected*	Date
Table of Contents	Revised	December 1984
Foreword	Revised	December 1984
Sect. C1	Revised	January 1982
p. C1.A.5	Corrected	December 1984
Sect. C2	New	January 1982
Sect. C4	New	January 1982
title page	Revised	December 1984
p. C4.iii-vii	Revised	December 1984
Sect. C4.4	Revised	December 1984
Sect. S2	New	June 1983
title page	Revised	December 1984
p. S2.2.9	Revised	December 1984
p. S2.4.7	Revised	December 1984
p. S2.5.5	Revised	December 1984
Sect. S3	New	January 1982
p. S3.A.1	Revised	June 1983
Sect. H1	New	June 1983
title page	Revised	December 1984
p. H1.iii-xi	Revised	December 1984
p. H1.xiii	Deleted	December 1984
p. H1.2.15	Revised	December 1984
p. H1.2.20	Revised	December 1984
p. H1.2.21	New	December 1984
p. H1.3.4	Revised	December 1984
p. H1.3.6	Revised	December 1984
p. H1.3.11-14	Revised	December 1984
p. H1.4.3	Revised	December 1984
p. H1.4.15	Revised	December 1984
p. H1.4.17-18	Revised	December 1984
p. H1.4.20	Revised	December 1984
p. H1.4.25-26	Revised	December 1984
p. H1.4.30	Revised	December 1984
p. H1.4.31-32	Revised	December 1984
Sect. H1.5	Revised	December 1984
Sect. H1.6	Revised	December 1984
Sect. H1.A	Revised	December 1984
Sect. H1.B	Revised	December 1984
Sect. H1.C	Deleted	December 1984
Sect. F1	New	January 1982
p. F1.2.2	Corrected	June 1983
p. F1.2.5	Corrected	June 1983
p. F1.5.1	Corrected	June 1983
p. F1.5.2	Corrected	December 1984
p. F1.6.2	Corrected	July 1982
p. F1.6.11	Corrected	July 1982

Page/Section	New/Revised/Corrected*	Date
Sect. F9	New	January 1982
title page	Revised	December 1984
p. F9.A.1	Revised	December 1984
p. F9.A.12	Revised	December 1984
p. F9.A.13	Revised	December 1984
p. F9.B.6	Corrected	December 1984
p. F9.B.7-9	Revised	December 1984
p. F9.B.10	Deleted	December 1984
Sect. F10	New	January 1982
p. F10.iv	Revised	June 1983
p. F10.iv	Corrected	December 1984
p. F10.2.23	Corrected	December 1984
p. F10.3.38	Revised	June 1983
p. F10.5.9-12	Revised	June 1983
p. F10.5.9	Corrected	December 1984
p. F10.5.14	Corrected	December 1984
p. F10.5.35	Revised	June 1983
p. F10.5.38	Corrected	December 1984
p. F10.A.1-2	Corrected	December 1984
p. F10.A.4-5	Corrected	December 1984
p. F10.C.36	Revised	June 1983
Sect. F11	Revised	December 1984
Sect. F12	Revised	December 1984
Sect. F13	New	December 1984
Sect. F14	New	December 1984
Sect. M1	New	January 1982
Sect. M2	New	January 1982
p. M2.2.58	New	June 1983
p. M2.2.73	Revised	December 1984
Sect. M3	New	January 1982
Sect. M6	New	June 1983
title page	Revised	December 1984
p. M6.v	Revised	December 1984
p. M6.vii	Revised	December 1984
p. M6.4.1-2	Revised	December 1984
p. M6.7.4	Revised	December 1984
p. M6.7.6-10	Revised	December 1984
p. M6.8.3	Revised	December 1984
Sect. M8	Revised	June 1983
title page	Revised	December 1984
p. M8.2.2-13	Revised	December 1984
p. M8.4.1	Revised	December 1984

Page/Section	New/Revised/Corrected*	Date
Sect. F2	New	January 1982
p. F2.2.3	Corrected	June 1983
p. F2.2.15	Revised	June 1983
pp.F2.3.33-34	Corrected	July 1982
p. F2.3.41	Corrected	July 1982
p. F2.4.4	Corrected	December 1984
Sect. F3	New	June 1983
p. F3.12.6	Corrected	December 1984
Sect. F4	New	June 1983
Sect. F5	New	January 1982
p. F5.6.2	Corrected	June 1983
p. F5.B.3	Corrected	December 1984
Sect. F6	New	January 1982
title page	Revised	December 1984
p. F6.4.6	Revised	December 1984
p. F6.5.14	Revised	December 1984
Sect. F7	New	January 1982
title page	Revised	December 1984
p. F7.iii	Revised	December 1984
p. F7.v-vi	Revised	June 1983
p. F7.ix	Revised	December 1984
p. F7.2.15	Revised	June 1983
p. F7.2.27-28	New	December 1984
p. F7.3.3-4	Revised	December 1984
p. F7.4.2	Revised	December 1984
p. F7.5.11	Revised	December 1984
p. F7.6.2	Revised	June 1983
p. F7.6.4	Revised	December 1984
p. F7.6.6	Revised	June 1983
p. F7.6.8	Revised	December 1984
p. F7.6.10-11	Revised	December 1984
p. F7.6.15-19	Revised	December 1984
p. F7.6.21	Revised	June 1983
p. F7.6.22-23	Revised	December 1984
p. F7.6.25-26	Revised	June 1983
p. F7.6.27-28	Revised	December 1984
p. F7.6.30	Revised	December 1984
p. F7.6.33-34	Revised	June 1983
p. F7.6.37-38	Revised	December 1984
p. F7.6.39-40	Revised	June 1983
p. F7.6.43	Revised	June 1983
p. F7.6.44	Revised	December 1984
p. F7.7.1	Revised	December 1984
p. F7.8.1	Revised	December 1984
p. F7.9.2	Revised	December 1984
Sect. F8	New	June 1983

Page/Section	New/Revised/Corrected*	Date
Sect. M9	New	January 1982
title page	Revised	December 1984
p. M9.iv	Revised	December 1984
p. M9.vii	Revised	December 1984
p. M9.6.5-6	Revised	December 1984
p. M9.E.2-3	Corrected	December 1984
p. M9.E.4-6	Revised	December 1984
p. M9.E.14-15	Revised	December 1984
p. M9.E.16-19	New	December 1984
Sect. M9.G	New	December 1984
Sect. M10	New	January 1982
Sect. M11	New	January 1982
Sect. M12	New	December 1984
Sect. M13	New	December 1984
Sect. M14	New	December 1984

* New = First issuance of document section or page
Revised = Revision of document section or pages to reflect code updates or changes
Corrected = Correction to reflect error in previous issuance of document

TABLE OF CONTENTS
FOR SCALE MANUAL

		Page*
ACKNOWLEDGMENTS		vii
ABSTRACT		ix
FOREWORD		xi
LIST OF REVISIONS		xv
	VOLUME 1: CONTROL MODULES	
Section C1	CSAS1: A ONE-DIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C1.1.1
Section C2	CSAS2: A MULTIDIMENSIONAL CRITICALITY SAFETY ANALYSIS MODULE	C2.1.1
Section C3	CSAS3: AN OPTIMUM CONCENTRATION CRITICALITY SAFETY ANALYSIS MODULE (to be issued at later date)	C3.1.1
Section C4	CSAS4: AN ENHANCED CRITICALITY SAFETY ANALYSIS MODULE WITH AN OPTIMUM PITCH SEARCH OPTION	C4.1.1
Section S1	SAS1: A ONE-DIMENSIONAL SHIELDING ANALYSIS MODULE (to be issued at later date)	S1.1.1
Section S2	SAS2: A COUPLED ONE-DIMENSIONAL DEPLETION AND SHIELDING ANALYSIS MODULE	S2.1.1
Section S3	SAS3: AN AUTOMATED MONTE CARLO SHIELDING ANALYSIS MODULE	S3.1.1
Section S4	SAS4: A MONTE CARLO SHIELDING ANALYSIS MODULE WITH AUTOMATIC BIASING. (to be issued at later date)	S4.1.1
Section H1	HTAS1: A TWO-DIMENSIONAL HEAT TRANSFER ANALYSIS OF FUEL CASKS	H1.1.1

*All section page numbers refer to the page where the section introduction begins.

VOLUME 2: FUNCTIONAL MODULES

		Page
Section F1	BONAMI-S: RESONANCE SELF-SHIELDING BY THE BONDARENKO METHOD	F1.1.1
Section F2	NITAWL-S: RESONANCE SELF-SHIELDING BY THE NORDHEIM METHOD	F2.1.1
Section F3	XSDRNPM-S: A ONE-DIMENSIONAL DISCRETE-ORDINATES PROGRAM FOR TRANSPORT ANALYSIS . . .	F3.1.1
Section F4	XSDOSE: A MODULE FOR CALCULATING FLUXES AND DOSE RATES AT POINTS OUTSIDE A SHIELD	F4.1.1
Section F5	KENO-IV/S: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM	F5.1.1
Section F6	COUPLE: SCALE SYSTEM MODULE TO PROCESS PROBLEM-DEPENDENT CROSS SECTIONS AND NEUTRON SPECTRAL DATA FOR ORIGEN-S ANALYSES	F6.1.1
Section F7	ORIGEN-S: SCALE SYSTEM MODULE TO CALCULATE FUEL DEPLETION, ACTINIDE TRANSMUTATION, FISSION PRODUCT BUILDUP AND DECAY, AND ASSOCIATED RADIATION SOURCE TERMS	F7.1.1
Section F8	ICE-S: MODULE TO MIX MULTIGROUP CROSS SECTIONS	F8.1.1
Section F9	MORSE-SGC/S: A NEUTRON AND PHOTON MONTE CARLO TRANSPORT ANALYSIS WITH SUPER-GROUPING AND ADVANCED GEOMETRY FEATURES	F9.1.1
Section F10	HEATING6: A MULTIDIMENSIONAL HEAT CONDUCTION ANALYSIS WITH THE FINITE-DIFFERENCE FORMULATION	F10.1.1
Section F11	KENO-V.a: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM WITH SUPER-GROUPING	F11.1.1
Section F12	JUNEBUG-II: A THREE-DIMENSIONAL GEOMETRY PLOTTING CODE	F12.1.1
Section F13	HEATPLOT-S: A TEMPERATURE DISTRIBUTION PLOTTING PROGRAM FOR HEATING5/HEATING6.	F13.1.1
Section F14	REGPLOT6: A PLOTTING PROGRAM TO VERIFY HEATING6 INPUT DATA	F14.1.1
Section F15	PLORIGEN: A PLOTTING PROGRAM FOR ORIGEN-S OUTPUT (to be issued at later date)	F15.1.1

VOLUME 3: MISCELLANEOUS

		Page
Section M1	THE SCALE SYSTEM DRIVER	M1.1.1
Section M2	THE SCALE SYSTEM SUBROUTINE LIBRARY	M2.1.1
Section M3	SCALE FREE FORM READING ROUTINES	M3.1.1
Section M4	NEUTRON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M4.1.1
Section M5	NEUTRON-PHOTON CROSS SECTION LIBRARIES . . . . . (to be issued at later date)	M5.1.1
Section M6	ORIGEN-S DATA LIBRARIES	M6.1.1
Section M7	MATERIAL PROPERTIES LIBRARIES . . . . . (to be issued at later date)	M7.1.1
Section M8	STANDARD COMPOSITION LIBRARY	M8.1.1
Section M9	MARS: A MULTIPLE ARRAY SYSTEM USING COMBINATORIAL GEOMETRY	M9.1.1
Section M10	FIDO INPUT SYSTEM	M10.1.1
Section M11	SCALE INTERACTIVE INPUT PROCESSOR	M11.1.1
Section M12	CESAR: A CRITICALITY EXPERIMENT STORAGE AND RETRIEVAL PROGRAM.	M12.1.1
Section M13	PICTURE: A PRINTER PLOT PACKAGE FOR MAKING 2-D PICTURES OF MARS GEOMETRIES.	M13.1.1
Section M14	COMPOZ DATA GUIDE.	M14.1.1

FOREWORD FOR SCALE MANUAL

This manual describes a system of computer codes for performing Standardized Computer Analyses for Licensing Evaluation (SCALE). The SCALE system was conceived and funded by the U. S. Nuclear Regulatory Commission to satisfy a strong need for a standardized method of analysis for the evaluation of nuclear fuel facility and package designs. Thus, the SCALE system will lead to a common understanding on how these designs will be evaluated by the NRC.

The overall goal of the SCALE project has been to develop easy-to-use analytical sequences which are automated to perform the necessary data processing (e.g., cross-section preparation) and manipulation of well-established computer codes (functional modules) required by the sequence. Thus, the user is able to select an analytical sequence characterized by the type of analysis (criticality, shielding, or heat transfer) to be performed and the geometric complexity of the system being analyzed. The user then prepares a single set of input for the control module corresponding to this analytical sequence. The control module input is in terms of easily visualized engineering parameters specified in a simplified, free-form format. The control modules use this information to derive additional parameters and prepare the input for each of the functional modules in the analytical sequence. Provisions have also been made to allow the user to execute the functional modules on a stand-alone basis. Stand-alone execution of the functional modules enables the SCALE system to be used for a wide range of nuclear and thermal applications.

In developing the SCALE system, an attempt has been made to adhere to certain criteria that would make SCALE an attractive computational tool for both the novice and experienced analyst. Some of the criteria are to:

a) develop a system driver which minimizes the computer core and running time,

b) develop control modules which require only a simplified set of input,

c) develop control modules which automate and couple the data processing and systems analysis into standard analytical sequences,

d) provide self-contained and thorough documentation for the entire system,

e) use well-established functional modules and data libraries that have a wide range of applicability and a proven record of reliable performance.

The overall objective of following these criteria was to develop SCALE as a system leading to analytical standardization.

The control modules available with SCALE are documented in Volume 1 of this manual. The SCALE system contains a wide assortment of control modules for performing criticality, shielding, and heat transfer analysis. Criticality Safety Analysis Sequences are referred to as CSAS1-CSASn and are described in Sections C1-Cn. Likewise, the shielding and heat transfer analysis sequences are called SAS1-SASn and HTAS1-HTASn, respectively and are described in Sections S1-Sn and H1-Hn, respectively.

Volume 2 of this manual provides documentation for each of the well-established functional modules used within the SCALE system. These sections are referred to as Sections F1-Fn. The functional modules include codes for data and/or cross section preparation (e.g. BONAMI-S and NITAWL-S), criticality analysis (e.g. XSDRNPM-S, KENO-IV/S, KENO-V), shielding analysis (e.g., XSDRNPM-S and MORSE-SGC/S), heat transfer analysis (HEATING6), and depletion analysis (ORIGEN-S). The "S" attached to the name of some functional modules indicates that the code was originally developed outside of SCALE and modified for use in the SCALE system. The exception to this is the HEATING6 code which is an upgrade of HEATING5.

Volume 3 of this manual provides documentation for the miscellaneous (but important) components of the SCALE system. These sections are referred to as Sections M1-Mn and include descriptions of the system driver, the subroutine library, the input systems, and the data, material composition, and property libraries. The data libraries include a variety of well-established cross section libraries, as well as some new libraries developed within the SCALE project.

The SCALE Manual is being released in a modular fashion; i.e., as each new component of the SCALE system is developed and released, a new section will be added to the SCALE Manual. Also, the documentation is released in looseleaf form so that as corrections and small revisions are made to the various modules, new or revised pages can be easily issued for the Manual. The List of Revisions issued with each release of new documentation provides a list of the NRC publication dates for each new or revised section and/or pages.

The SCALE system is being packaged and distributed by the Radiation Shielding Information Center (RSIC), located at Oak Ridge National Laboratory. A limited version of SCALE (SCALE-0) was made available to RSIC in July 1980. Significant revisions and additions have been made to SCALE and new versions were released as SCALE-1 in 1982 and SCALE-2 in 1983. This latest issuance of documentation should upgrade the existing documentation to match the SCALE-3 version of the system presently being issued by RSIC. Only IBM versions of SCALE are currently available from RSIC with the exception of SCALE-0 for which a CDC version is available. Requests for the latest SCALE code package can be mailed to:

Radiation Shielding Information Center
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

or telephoned to

Area Code 615; 574-6176 or to FTS 624-6176.

C. V. Parks
SCALE Project Leader
Oak Ridge National Laboratory

LIST OF REVISIONS for NUREG/CR-0200,
SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

Page/Section	New/Revised/Corrected*	Date
Table of Contents	Revised	December 1984
Foreword	Revised	December 1984
Sect. C1	Revised	January 1982
p. C1.A.5	Corrected	December 1984
Sect. C2	New	January 1982
Sect. C4	New	January 1982
title page	Revised	December 1984
p. C4.iii-vii	Revised	December 1984
Sect. C4.4	Revised	December 1984
Sect. S2	New	June 1983
title page	Revised	December 1984
p. S2.2.9	Revised	December 1984
p. S2.4.7	Revised	December 1984
p. S2.5.5	Revised	December 1984
Sect. S3	New	January 1982
p. S3.A.1	Revised	June 1983
Sect. H1	New	June 1983
title page	Revised	December 1984
p. H1.iii-xi	Revised	December 1984
p. H1.xiii	Deleted	December 1984
p. H1.2.15	Revised	December 1984
p. H1.2.20	Revised	December 1984
p. H1.2.21	New	December 1984
p. H1.3.4	Revised	December 1984
p. H1.3.6	Revised	December 1984
p. H1.3.11-14	Revised	December 1984
p. H1.4.3	Revised	December 1984
p. H1.4.15	Revised	December 1984
p. H1.4.17-18	Revised	December 1984
p. H1.4.20	Revised	December 1984
p. H1.4.25-26	Revised	December 1984
p. H1.4.30	Revised	December 1984
p. H1.4.31-32	Revised	December 1984
Sect. H1.5	Revised	December 1984
Sect. H1.6	Revised	December 1984
Sect. H1.A	Revised	December 1984
Sect. H1.B	Revised	December 1984
Sect. H1.C	Deleted	December 1984
Sect. F1	New	January 1982
p. F1.2.2	Corrected	June 1983
p. F1.2.5	Corrected	June 1983
p. F1.5.1	Corrected	June 1983
p. F1.5.2	Corrected	December 1984
p. F1.6.2	Corrected	July 1982
p. F1.6.11	Corrected	July 1982

Page/Section	New/Revised/Corrected*	Date
Sect. F2	New	January 1982
p. F2.2.3	Corrected	June 1983
p. F2.2.15	Revised	June 1983
pp.F2.3.33-34	Corrected	July 1982
p. F2.3.41	Corrected	July 1982
p. F2.4.4	Corrected	December 1984
Sect. F3	New	June 1983
p. F3.12.6	Corrected	December 1984
Sect. F4	New	June 1983
Sect. F5	New	January 1982
p. F5.6.2	Corrected	June 1983
p. F5.B.3	Corrected	December 1984
Sect. F6	New	January 1982
title page	Revised	December 1984
p. F6.4.6	Revised	December 1984
p. F6.5.14	Revised	December 1984
Sect. F7	New	January 1982
title page	Revised	December 1984
p. F7.iii	Revised	December 1984
p. F7.v-vi	Revised	June 1983
p. F7.ix	Revised	December 1984
p. F7.2.15	Revised	June 1983
p. F7.2.27-28	New	December 1984
p. F7.3.3-4	Revised	December 1984
p. F7.4.2	Revised	December 1984
p. F7.5.11	Revised	December 1984
p. F7.6.2	Revised	June 1983
p. F7.6.4	Revised	December 1984
p. F7.6.6	Revised	June 1983
p. F7.6.8	Revised	December 1984
p. F7.6.10-11	Revised	December 1984
p. F7.6.15-19	Revised	December 1984
p. F7.6.21	Revised	June 1983
p. F7.6.22-23	Revised	December 1984
p. F7.6.25-26	Revised	June 1983
p. F7.6.27-28	Revised	December 1984
p. F7.6.30	Revised	December 1984
p. F7.6.33-34	Revised	June 1983
p. F7.6.37-38	Revised	December 1984
p. F7.6.39-40	Revised	June 1983
p. F7.6.43	Revised	June 1983
p. F7.6.44	Revised	December 1984
p. F7.7.1	Revised	December 1984
p. F7.8.1	Revised	December 1984
p. F7.9.2	Revised	December 1984
Sect. F8	New	June 1983

Page/Section	New/Revised/Corrected*	Date
Sect. F9	New	January 1982
title page	Revised	December 1984
p. F9.A.1	Revised	December 1984
p. F9.A.12	Revised	December 1984
p. F9.A.13	Revised	December 1984
p. F9.B.6	Corrected	December 1984
p. F9.B.7-9	Revised	December 1984
p. F9.B.10	Deleted	December 1984
Sect. F10	New	January 1982
p. F10.iv	Revised	June 1983
p. F10.iv	Corrected	December 1984
p. F10.2.23	Corrected	December 1984
p. F10.3.38	Revised	June 1983
p. F10.5.9-12	Revised	June 1983
p. F10.5.9	Corrected	December 1984
p. F10.5.14	Corrected	December 1984
p. F10.5.35	Revised	June 1983
p. F10.5.38	Corrected	December 1984
p. F10.A.1-2	Corrected	December 1984
p. F10.A.4-5	Corrected	December 1984
p. F10.C.36	Revised	June 1983
Sect. F11	Revised	December 1984
Sect. F12	Revised	December 1984
Sect. F13	New	December 1984
Sect. F14	New	December 1984
Sect. M1	New	January 1982
Sect. M2	New	January 1982
p. M2.2.58	New	June 1983
p. M2.2.73	Revised	December 1984
Sect. M3	New	January 1982
Sect. M6	New	June 1983
title page	Revised	December 1984
p. M6.v	Revised	December 1984
p. M6.vii	Revised	December 1984
p. M6.4.1-2	Revised	December 1984
p. M6.7.4	Revised	December 1984
p. M6.7.6-10	Revised	December 1984
p. M6.8.3	Revised	December 1984
Sect. M8	Revised	June 1983
title page	Revised	December 1984
p. M8.2.2-13	Revised	December 1984
p. M8.4.1	Revised	December 1984

Page/Section	New/Revised/Corrected*	Date
Sect. M9	New	January 1982
title page	Revised	December 1984
p. M9.iv	Revised	December 1984
p. M9.vii	Revised	December 1984
p. M9.6.5-6	Revised	December 1984
p. M9.E.2-3	Corrected	December 1984
p. M9.E.4-6	Revised	December 1984
p. M9.E.14-15	Revised	December 1984
p. M9.E.16-19	New	December 1984
Sect. M9.G	New	December 1984
Sect. M10	New	January 1982
Sect. M11	New	January 1982
Sect. M12	New	December 1984
Sect. M13	New	December 1984
Sect. M14	New	December 1984

* New = First issuance of document section or page
Revised = Revision of document section or pages to reflect code updates or changes
Corrected = Correction to reflect error in previous issuance of document

NUREG/CR-0200
VOLUME 1
ORNL/NUREG/CSD-2/V1/R3

SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

REVISIONS AND ERRATA SHEETS FOR VOLUME 1: CONTROL MODULES

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
Under Interagency Agreement 40-549-75

C1.A.3. Nuclides Having Resonance Data

The list of nuclides having resonance data differs from one master cross-section library to another as shown in Table C1.A-2. The temperature of all materials containing any of these nuclides should be specified by the user. NITAWL will then use the resonance parameters to properly account for the Doppler broadening of the cross section data.

The six nuclides denoted by an (*) in the HANSEN-ROACH library have Bondarenko data in lieu of resonance parameters. Regrettably,the Bondarenko factors for these nuclides are available only at a single temperature. Thus, for example, the HANSEN-ROACH library should not be used to determine the Doppler coefficient of reactivity.

Table C1.A-2. Nuclides Having Resonance Data

Resonance nuclides found on the HANSEN-ROACH library:
25055, 47107, 47109, 49113, 49115, 66164, 77175,
71176, 74182, 74183, 74184, 74186, 75185, 75187,
79197, 90232*, 91233, 92233*, 92234, 92235*, 92236,
92238*, 93237, 94238*, 94239, 94240*, 94241, 94242,
95241, 95243, 96244
Resonance nuclides found on the 27GROUPNDF4 and 218GROUPNDF4 libraries:
11023, 25055, 26000, 27059, 29000, 35790, 35810, 40302,
41093, 42000, 47107, 47109, 49113, 49115, 55133, 64000,
66164, 77175, 77176, 73181, 74182, 74183, 74184,
74186, 75185, 75187, 79197, 90232, 91233, 92233,
92234, 92235, 92236, 92238, 93237, 94238, 94239,
94240, 94241, 94242, 95241, 95243, 96244
Resonance nuclides found on the 123GROUPGMTH library:
90232, 92234, 92236, 92238, 94240, 94242

*denotes nuclides having Bondarenko data in lieu of resonance parameters

C1.A.4. Nuclides With Multiple Sets of Thermal Scattering Data

The list of nuclides having thermal scattering cross-section data at more than one temperature differs from one cross-section library to another as shown in Table C1.A-3. Likewise, the temperatures at which thermal scattering data is available for a given nuclide may also differ from one library to the next as shown in Table C1.A-3.

Since scattering matrices for nuclides in media at elevated temperatures are generally fuller than those at lower temperatures, the user should enter a rough estimate of the temperature of all materials containing any of these nuclides. From the various sets of thermal scattering data available on the master cross-section library, NITAWL will produce a working library containing the data for the temperature closest to that specified by the user.

Lastly, it should be noted that those nuclides found in the Standard Composition Library (see Sect. M8) but not listed in Table C1.A-3 all have thermal scattering data at room temperature (293°K).

Table C1.A-3. Nuclides With Multiple Sets of Thermal Scattering Data

Cross-Section Library Specified by User	Nuclide I.D. No.	Temperatures (°K) for Which Different Sets of Thermal Scattering X-Section Data is Available
HANSEN-ROACH	5010	293, 550
	5011	293, 550
27GROUPNDF4	1001	293, 550
and	1002	293, 550
218GROUPNDF4	4009	296, 900, 1000, 1200
	5010	293, 550
	5011	293, 550
	6012	293, 900, 1000, 1200
123GROUPGMTH	1001	295, 345
	1002	294, 345, 361
	3006	294, 800, 900, 1000
	3007	294, 800, 850, 900, 950, 1000
	4009	294, 800, 850, 900, 950, 1000
	5011	300, 600, 900, 1200
	6012	589, 627, 675, 800, 850, 900, 950, 1000
	8016	294, 345, 361, 900, 1165
	9019	294, 800, 850, 900, 950, 1000
	11023	294, 900
	13027	627, 900
	24000	295, 627, 900
	26000	295, 627, 900
	28000	295, 627, 900
	29000	295, 900

NUREG/CR-0200
Volume 1, Section C4
ORNL/NUREG/CSD-2/V1/R3

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

CSAS4: AN ENHANCED CRITICALITY SAFETY ANALYSIS MODULE WITH AN OPTIMUM PITCH SEARCH OPTION

N. F. Landers
L. M. Petrie

Manuscript Completed: October 1981
Date Published: October 1981
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

TABLE OF CONTENTS

LIST OF FIGURES C4.v

LIST OF TABLES C4.vii

ACKNOWLEDGMENTS C4.ix

ABSTRACT C4.xi

C4.1 INTRODUCTION TO CSAS4 ..... C4.1.1
C4.2 TECHNIQUES C4.2.1
C4.3 LOGICAL PROGRAM FLOW C4.3.1

C4.3.1 INITIATION OF CRITICALITY SAFETY ANALYSIS SEQUENCE NO. 4 ..... C4.3.2
C4.3.2 OVERALL FLOW DIAGRAM C4.3.5
C4.3.3 THE DATA PREPROCESSOR C4.3.9
C4.3.4 THE MATERIAL INFORMATION PROCESSOR C4.3.12
C4.3.5 OVERALL CONTROL FOR THE MATERIAL INFORMATION PROCESSOR C4.3.13
C4.3.6 GENERATION OF MATERIAL INFORMATION PROCESSOR MIXING TABLE C4.3.15
C4.3.7 GENERATION OF RESONANCE DATA C4.3.17
C4.3.8 KENO V DATA PROCESSOR C4.3.19
C4.3.9 CONTROL OF THE DATA INPUT FOR KENO V C4.3.22
C4.3.10 READ KENO V ARRAY DATA C4.3.24
C4.3.11 READ KENO V GEOMETRY DATA C4.3.25
C4.3.12 READ KENO V BIASING DATA C4.3.26
C4.3.13 GENERATE REMAINING KENO V DATA C4.3.27
C4.3.14 CHECK THE KENO V GEOMETRY DATA C4.3.28
C4.3.15 THE SEARCH DATA PROCESSOR C4.3.29
C4.3.16 MODIFY, THE SEARCH MODULE C4.3.32
C4.3.17 REFERENCES C4.3.34

C4.4 INPUT DATA GUIDE C4.4.1

C4.4.1 INPUT DATA SUMMARY C4.4.1
C4.4.2 INPUT DATA REQUIREMENTS C4.4.22
C4.4.3 MATERIAL INFORMATION PROCESSOR DATA C4.4.22
C4.4.4 STANDARD COMPOSITIONS SPECIFICATION DATA C4.4.24
C4.4.5 UNIT CELL SPECIFICATION FOR LATTICECELL PROBLEMS C4.4.30
C4.4.6 GEOMETRY SPECIFICATION FOR MULTIREGION PROBLEMS C4.4.33
C4.4.7 OPTIONAL PARAMETER DATA C4.4.36
C4.4.8 SUMMARY OF KENO-V DATA C4.4.39
C4.4.9 SEARCH DATA C4.4.48

C4.5 NOTES FOR USERS C4.5.1

C4.5.1 RESONANCE TREATMENT C4.5.1
C4.5.2 MIXTURES C4.5.1
C4.5.3 COMPLETION OF THE SEARCH C4.5.2
C4.5.4 KENO V DATA C4.5.2

C4.6 DESCRIPTION OF OUTPUT C4.6.1

C4.6.1 HEADER PAGE C4.6.1
C4.6.2 PROBLEM PARAMETERS C4.6.2
C4.6.3 PROBLEM COMPOSITION DESCRIPTION C4.6.2
C4.6.4 PROBLEM GEOMETRY C4.6.3
C4.6.5 OPTIONAL PARAMETER DATA C4.6.4
C4.6.6 XSDRN MESH INTERVALS C4.6.5
C4.6.7 PROCESSING ACCOUNTING INFORMATION C4.6.5
C4.6.8 SEARCH PASS ITERATION DATA C4.6.6
C4.6.9 SEARCH CONVERGENCE INFORMATION C4.6.9

C4.7 WARNING AND ERROR MESSAGES C4.7.1

C4.7.1 MATERIAL INFORMATION PROCESSOR MESSAGES C4.7.1
C4.7.2 DATA PREPROCESSOR MESSAGES C4.7.11
C4.7.3 MESSAGES FROM CRITICALITY SAFETY SEQUENCE NO. 4 ..... C4.7.16
C4.7.4 STOP CODES C4.7.33

APPENDICES

C4.A ALPHABETICAL INDEX OF SUBROUTINES C4.A.1
C4.B ALPHABETICAL INDEX OF COMMONS C4.B.1
C4.C SAMPLE PROBLEMS C4.C.1
C4.D SAMPLE OUTPUT C4.D.1

C4.D.1 CONTROL MODULE OUTPUT C4.D.1
C4.D.2 COMPLETE SAMPLE PROBLEM OUTPUT C4.D.12

LIST OF FIGURES

C4.3.1 Control modules available in criticality safety analysis sequence no. 4 ..... C4.3.3

C4.3.2 Overall flow of criticality safety analysis sequence no. 4 ..... C4.3.5

C4.3.3 Flow chart of the data preprocessor C4.3.9

C4.3.4 Initiation of the material information processor C4.3.12

C4.3.5 Overall control for the material information processor C4.3.14

C4.3.6 Flow chart for generating the mixing table C4.3.16

C4.3.7 Flow chart for generating resonance data C4.3.17

C4.3.8 Flow chart of routines to process KENO V data C4.3.19

C4.3.9 Flow chart of KENO V data reading routines C4.3.22

C4.3.10 Flow chart for reading KENO V array data C4.3.24

C4.3.11 Flow chart of routines to read the KENO V geometry data C4.3.25

C4.3.12 Flow chart for reading KENO V biasing data C4.3.26

C4.3.13 Flow chart for generating remaining KENO V data C4.3.27

C4.3.14 Flow chart for checking KENO V geometry data C4.3.28

C4.3.15 Flow chart of the search data processor C4.3.30

C4.3.16 Flow chart of the search module, MODIFY C4.3.32

C4.4.1 Outline of CSAS4 input data C4.4.3

C4.4.2 Arrangement of materials in a symmetric slab cell C4.4.32

C4.4.3 Arrangement of materials in an asymmetric slab cell C4.4.32

C4.6.2 Example of problem parameters C4.6.2

C4.6.3 Example of problem composition description C4.6.3

C4.6.4 Sample output for latticecell C4.6.4

C4.6.5 Sample output for multiregion geometry C4.6.4

C4.6.6 Example of optional parameter output C4.6.5

C4.6.7 Example of XSDRN mesh intervals C4.6.5

C4.6.8 Example of processing accounting information C4.6.6

C4.6.9 Example of search pass iteration data C4.6.6

C4.6.10 Example of search convergence information C4.6.7

LIST OF TABLES

Table C4.3.1 Description of units utilized by criticality safety analysis sequence no. 4 ..... C4.3.1

Table C4.4.1 Functional modules executed by CSAS control modules C4.4.1

Table C4.4.2 I/O units on which cross-section libraries are written C4.4

Table C4.4.3 I/O units generated by CSAS4 and associated modules C4.4.2

Table C4.4.4 Tables of input data requirements C4.4.5

Table C4.4.5 Summary of KENO-Va input data C4.4.12

Table C4.4.6 Summary of SEARCH input data C4.4.21

Table C4.4.7 Summary of data input requirements C4.4.22

Table C4.4.8 Outline of material information processor data C4.4.23

Table C4.4.9 Outline of standard composition specification data C4.4.25

Table C4.4.10 Unit cell specification for latticecell problems C4.4.31

Table C4.4.11 Geometry specification for multiregion problems C4.4.34

Table C4.4.12 Optional parameter data C4.4.37

Table C4.4.13 Outline of KENO-V data C4.4.39

Table C4.4.14 Summary of KENO-V data C4.4.40

Table C4.4.15 Outline of search type specification C4.4.48

Table C4.4.16 Outline of individual search commands C4.4.51

ABSTRACT

CSAS4 and CSAS4X were developed to provide an optimum pitch search capability for three-dimensional problems in the SCALE system. They have been upgraded to increase their search capabilities and to interface with KENO V.a rather than KENO V. This expands the geometrical capabilities to allow arrays of arrays, holes, variable chords for hemicylinders and hemispheres, and in-line printer plots.

Several control sequences are available within the CSAS4-CSAS4X family. These control sequences activate the cross-section processing codes BONAMI-S and NITAWL-S to provide resonance-corrected cross sections. Control modules ending with "X" utilize XSDRNPM-S to provide a cell-weighted mixture cross section. ICE-S is used by CSASI and CSASIX to provide a Monte Carlo formatted mixed cross-section library. KENO V.a is used to calculate the effective neutron multiplication factor (k-effective) of the system. The search capability is achieved by repeatedly activating the modules MODIFY to alter the pitch or other dimensions and KENO V.a to calculate the k-effective for the new pitch or dimensions.

The control sequences described within this document were developed simultaneously and are closely related. These sequences are listed below with the modules they invoke.

CSASI	no search	BONAMI-S	NITAWL-S		ICE-S	
CSASIX	no search	BONAMI-S	NITAWL-S	XSDRNPM-S	ICE-S	
CSASN	no search	BONAMI-S	NITAWL-S			
CSAS1X	no search	BONAMI-S	NITAWL-S	XSDRNPM-S		
CSAS25	no search	BONAMI-S	NITAWL-S		KENO V.a	
CSAS2X	no search	BONAMI-S	NITAWL-S	XSDRNPM-S	KENO V.a	
CSAS4	search	BONAMI-S	NITAWL-S		KENO V.a	MODIFY
CSAS4X	search	BONAMI-S	NITAWL-S	XSDRNPM-S	KENO V.a	MODIFY

C4.4 INPUT DATA GUIDE

This section describes the input data required for Criticality Safety Analysis Sequence No. 4 (CSAS4X). Several subsets of the CSAS4X sequence are available to achieve several different levels of processing. These are listed in Table C4.4.1 with the functional modules they invoke. Note that modules with an "X" in their names create cell-weighted libraries by using XSDRNPM-S.

Table C4.4.1 Functional modules executed by CSAS control modules

Control Module	Search Function	Functional modules executed by the control module					
CSASN	no search	BONAMI-S	NITAWL-S				
CSAS1X	no search	BONAMI-S	NITAWL-S	XSDRNPM-S			
CSASI	no search	BONAMI-S	NITAWL-S		ICE-S		
CSASIX	no search	BONAMI-S	NITAWL-S	XSDRNPM-S	ICE-S		
CSAS25	no search	BONAMI-S	NITAWL-S			KENO V	
CSAS2X	no search	BONAMI-S	NITAWL-S	XSDRNPM-S		KENO V	
CSAS4	search	BONAMI-S	NITAWL-S			KENO V	MODIFY*
CSAS4X	search	BONAMI-S	NITAWL-S	XSDRNPM-S		KENO V	MODIFY*

*MODIFY is a control module.

Each sequence generates various libraries that may be saved for future use. Table C4.4.2 lists the various I/O units that may be saved and their definitions. Table C4.4.3 lists those I/O units containing cross-section libraries that are generated by each sequence.

Table C4.4.2 I/O units on which cross-section libraries are written

Unit	Type of Data	Creating Module
2	resonance-corrected mixed working library	ICE-S
3	resonance-corrected cell-weighted working library	XSDRNPM-S
4	resonance-corrected working library	NITAWL-S
14	resonance-corrected mixed cross-section MORSE/KENO library If the invoked sequence contains an X in its name, this library contains a cell-weighted mixture identified as mixture 500.	ICE-S

The input data for these CSAS modules are composed of three broad categories of data. The first (Material Information Processor, including Standard Compositions Data and Geometry Specification) specifies the cross-section library and defines the composition of each mixture and the cell geometry that is used to process the cross sections. This data block is necessary for CSASN, CSAS1X, CSASI, and CSASIX. The second category of data is the KENO-V input data, which are used to specify the geometric and boundary conditions that represent the physical configuration of the problem. Both data blocks are necessary for CSAS25 and CSAS2X. The last category of data is the search data. All three data blocks are required for CSAS4 and CSAS4X.

Table C4.4.3 I/O units generated by CSAS4 and associated modules

Module Name	Primary Product	Unit numbers of generated cross-section libraries 2	3	4	14
CSASN	resonance-corrected working library			X	
CSAS1X	XSDRNPM eigenvalue calculation		X	X	
CSASI	resonance-corrected mixed cross-section MORSE/KENO library	X		X	X
CSASIX	resonance-corrected cell-weighted mixed cross-section MORSE/KENO library	X	X	X	X
CSAS25	KENO V k-effective	X		X	X
CSAS2X	KENO V k-effective using homogenized cell	X	X	X	X
CSAS4	dimension alterations	X		X	X
CSAS4X	dimension alterations using homogenized cell	X	X	X	X

All data are entered in free form, allowing alphanumeric data, floating point data, and integer data to be entered in an unstructured manner. All 80 columns of any card may be used. Data can usually start or end in any column. Each data entry must be followed by one or more blanks (a comma can also be used for numeric entries) to terminate the data entry. Integers may be entered for floating values. For example, 10 will be interpreted as 10.0. Imbedded blanks are not allowed within a data entry unless an E precedes a single blank as in an unsigned exponent in a floating point number. For example, 1.0E 4 would be correctly interpreted as 1.0×10^4.

Multiple entries of the same data value can be achieved by entering the number of repeats, followed by either R, *, or $, followed by the data value to be repeated. Imbedded blanks are not allowed between the number of repeats and the repeat flag. For example, 5R12, 5*12, 5$12, or 5R 12, etc., will enter five successive 12's in the input data. Multiple zeros can be specified as nZ where n is the number of zeroes to be repeated. The SCALE free-form reading routines are described in more detail in Sect. M3.

The purpose of this section is to define the input data in discrete subsections relating to a particular type of data. Tables of the input data are included in each subsection, and the entries are described in more detail after the tables illustrated in Fig. C4.4.1. Section C4.4.1 contains a summary of all the tables that are used to describe the input data. Sections C4.4.3 through C4.4.7 describe the material information processor data, Sect. C4.4.8 describes the KENO-V data, and Sect. C4.4.9 describes the search data.

Resonance-corrected cross sections utilize Dancoff corrections based on the unit cell specification. Because only one unit cell can be specified in the data, resonance corrections can only be applied to one mixture via conventional methods. However, if multiple resonance-corrected mixtures are required, the necessary data can be entered via the OPTIONAL PARAMETER DATA described in Sect. C4.4.7 and Tables C4.4.4 and C4.4.12.

To check the input data without actually processing the cross sections, the words "PARM=CHECK" or "PARM=CHK" should be entered, starting in column 11 of the analytical sequence specification for IBM versions. For example, =CSAS4 PARM=CHK would cause the input data for CSAS4 to be checked and appropriate error messages to be printed. If plots are specified in the data, they will be printed. This feature allows the user to debug and verify the input data while using a minimum of computer time. Many problems can be checked in ten seconds or less.

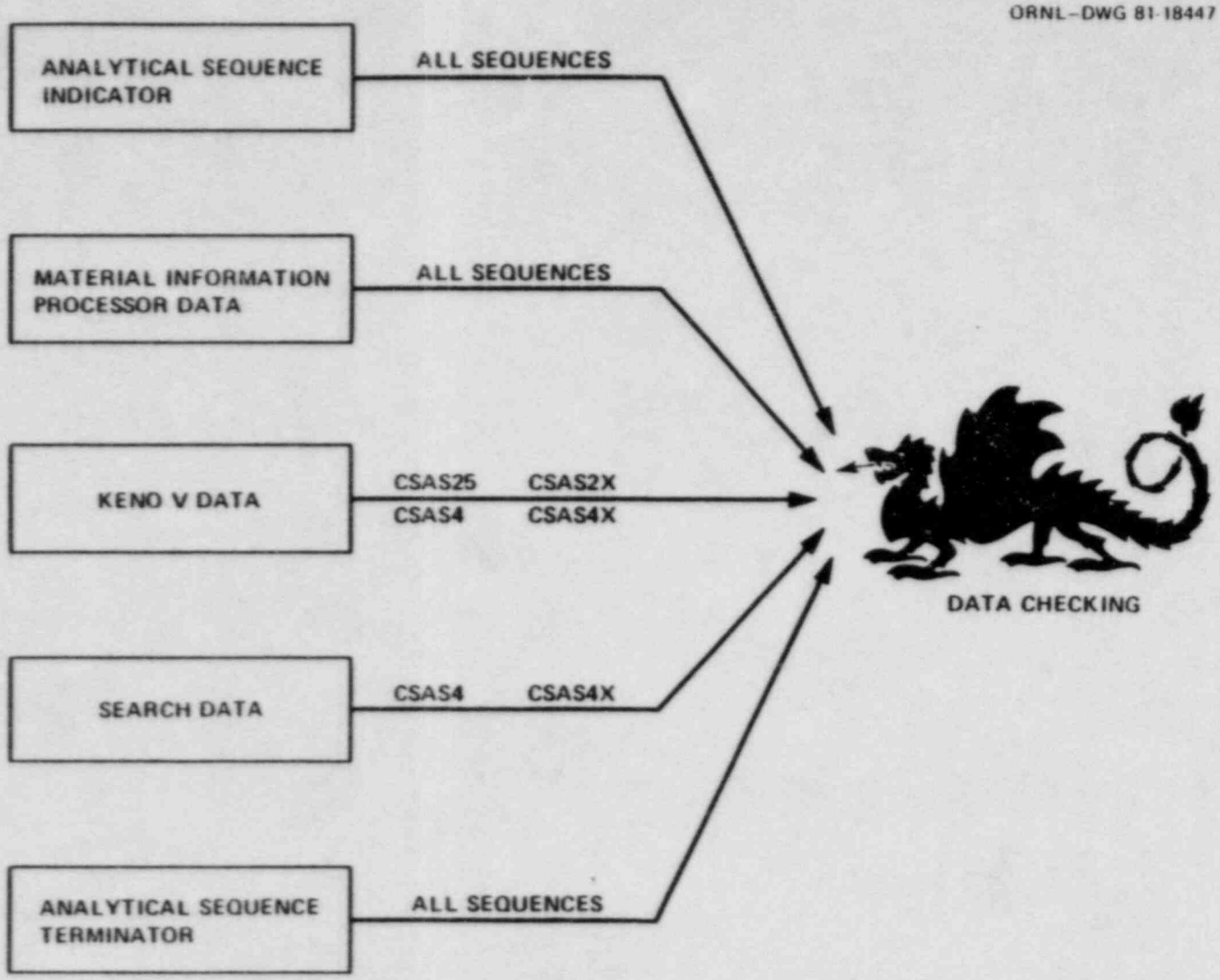


Fig. C4.4.1. Outline of CSAS4 input data

C4.4.1 INPUT DATA SUMMARY

The input data for Criticality Safety Analysis Sequence No. 4 are illustrated in Fig. C4.4.1, and summarized in Table C4.4.4. This table is provided as an input data guide for users who are familiar with CSAS data requirements. The KENO-V data are included in this table in very general terms. Table C4.4.5 contains a summary of the KENO-V input data. They are described in detail in Sect. F11.4 with additional information contained in Sect. F11.5. Table C4.4.6 contains a summary of the SEARCH data for CSAS4 and CSAS4X.

Table C4.4.4 Tables of input data requirements
(this table extends through page C4.4.11)

Summary of data input requirements (see Sect. C4.4.2)

	Type of Data	Analytical Sequence							
1	Analytical sequence specification (IBM versions)	=CSAS4	=CSAS4X	=CSAS25	=CSAS2X	=CSASIX	=CSASIX	=CSASN	=CSASI
2	Material information processor data	yes	yes	yes	yes	yes	yes	yes	yes
3	KENO V data	yes	yes	yes	yes	no	no	no	no
4	Search data	yes	yes	no	no	no	no	no	no
5	END for the analytical sequence (IBM versions)	yes	yes	yes	yes	yes	yes	yes	yes

Outline of material information processor data (see Sect. C4.4.3)

Date Position	Type of Data	Data Entry	Comments
1	TITLE	Enter a title	80 characters
2	Cross-section library name	HANSEN-ROACH 27GROUPNDF4 123GROUPGMTH 218GROUPNDF4 etc.	These are the currently available libraries. See Section C1.A.1
3	Type of calculation	INFHOMMEDIUM LATTICECELL MULTIREGION	These are the available options. See the explanation following this table.
4	Standard compositions specification data	Enter the appropriate data.	Terminate this data block with END COMP. See Table C4.4.4.
5	Cell geometry specification	Enter the appropriate data. (omit for INFHOMMEDIUM)	Omit for INFHOMMEDIUM. See Table C4.4.5 for LATTICECELL See Table C4.4.7 for MULTIREGION.
6	Optional parameter data	Enter the desired data	Precede this data block by MORE DATA, if more parameter data are to be entered. Otherwise, omit these data entirely. See Sect. C4.4.7.

Table C4.4.4 (continued through page C4.4.11)

Outline of standard compositions specifications data (see Sect. C4.4.4)

Entry Number	Variable Name	Type of data	Entry Requirement	Comments
1	SC	Standard composition component name	Always	Enter once for each standard composition. Enter the alphanumeric description from Table M8.2.1. Additional allowed names include those beginning with arbitrary materials, and SOLN for solutions.
A1	ROTH	Theoretical density of material (g/cc)	ARBM	Enter once for each standard composition component that is an arbitrary material.
A2	NEL	Number of elements in the material	ARBM	Enter once for each standard composition component that is an arbitrary material.
A3	IVIS	Multiple isotope indicator	ARBM	Enter once for each standard composition component that is an arbitrary material. Enter 0 for a material that does not contain multiple isotope elements. Enter 1 if the material contains a multiple isotope element. An arbitrary material cannot have more than one multiple isotope element. If necessary, several arbitrary materials may be used in a single mixture.
A4	ICP	Compound indicator	ARBM	Enter once for each standard composition component that is an arbitrary material. Enter 1 for a compound, 0 for alloys, mixtures, etc.
A5	IRS	Resonance indicator	ARBM	Enter once for each standard composition component that is an arbitrary material. Enter 0 if all the nuclides in the material are nonresonance nuclides; enter 1 if one or more have resonance data. See Section C1.A.3, Table C1.A-2.
A6	NCZA	ID number (from far right column of Table M8.2.1)	ARBM	Repeat the sequence A6 and A7 for each element in the arbitrary material before entering item 2. Enter the number from the far right column of Table M8.2.1. Only elements can be used in an arbitrary material.
A7	ATPM	Number of atoms of this element per molecule of arbitrary material OR Weight percent of this element in this arbitrary material	ARBM & ICP=1 ARBM & ICP=0	Repeat the sequence A6 and A7 for each element in arbitrary material before entering item 2. Do not enter a value unless ICP=1. Repeat the sequence A6 and A7 for each element in the arbitrary material before entering item 2. Do not enter a value unless ICP=0.

Table C4.4.4 (continued through page C4.4.11)

Outline of standard compositions specification data (continued)

Entry Number	Variable Name	Type of data	Entry Requirement	Comments
2	MX	Mixture ID number	Always	Enter once for each standard composition component.
S1	FD	Fuel density (grams of U or Pu per liter of solution)	SOLN	Enter once for a solution.
S2	AML	Acid molarity of the solution	SOLN	Enter once for a solution. AML=0 if there is no acid in the solution.
3	VF	Volume fraction (ratio of actual to theoretical density of the material)	See comment column	Default value is 1.0. This item can be omitted if items 4, 5, 6a, and 6b are also omitted.
4	ADEN	Number density (atoms/barn-cm) for the nuclide	VF=0	Enter only if VF=0.0.
5	TEMP	Temperature, in degrees K	See comment column	Default value is 293 K. This item can be omitted if items 6a and 6b are also omitted.
6a	IZA	Isotope's ZA number	VF=0	Enter for each isotope in the standard composition component. Omit if VF=0. Items 6a and 6b are entered in pairs until each isotope in the component is defined.
6b	WTP	Weight percent of the isotope	VF=0	Enter for each isotope in the standard composition component. Omit if VF=0.0. Items 6a and 6b are entered in pairs until each isotope in the component is defined.
7	END	Terminate a standard composition	Always	Enter once for each standard composition component. This terminates the data for a standard composition component. Enter END to terminate the component. Repeat items 1-7 until all the mixtures have been defined.
	END COMP	terminate the data block	Terminus	Enter once for a problem. Enter the words END COMP when all the standard composition components have been described.

Note: At least two blanks should separate the last entry 7 from the END COMP.

Table C4.4.4 (continued through page C4.4.11)

Unit cell specification for latticecell problems (see Sect. C4.4.5)

Entry Number	Variable Name	Type of Data	Entry Requirement	Data Entry	Comments
1	CTP	Type of lattice	Always		Describes the type of lattice or array configuration.
				SQUAREPITCH	Use for cylindrical rods in a square pitch.
				TRIANGPITCH	Use for cylindrical rods in a triangular pitch.
				SPHSQUAREP	Use for spherical pellets in a cubic lattice.
				SPHTRIANGP	Use for spherical pellets in a bi-centered or face-centered hexagonal close-packed lattice.
				SYMMSLABCELL	Use for a symmetric array of slabs.
				ASYMSLABCELL	Use for a periodic, but asymmetric, array of slabs.
2	PITCH	Array pitch	Always	Appropriate dimension	This is the center-to-center spacing (cm) between fuel lumps. For asymmetric slab cell, enter the distance from the center of one moderator to the center of the other moderator (cm).
3	FUELOD	Outside dimension of fuel	Always	Appropriate dimension	Outside diameter of fuel (cm) or the thickness of the fuel in a slab.
4	MFUEL	Fuel mixture number	Always	Mixture no.	Mixture number representing the fuel.
5	MMOD	Moderator mixture number	Always	Mixture no.	Mixture number representing the moderator.
6	MMOD2	2nd moderator mix. no.	ASYMSLABCELL	Mixture no.	Mixture number representing the second moderator.
7	TKMOD2	2nd moderator thickness	ASYMSLABCELL	Thickness	Thickness of the second moderator (cm).
8	CLADOD	Outside diameter of clad	If clad	Clad OD	OMIT IF NO CLAD. For a slab, CLADOD is the sum of thickness of the fuel, gap, and clad.
9	MCLAD	Clad mixture number	If clad	Mixture no.	OMIT IF NO CLAD. Mixture number representing the clad.
10	CLADID	Inside diameter of clad	If gap	Clad ID	OMIT IF NO GAP between the fuel and clad.
11	MGAP	Gap mixture number	If gap	Mixture no.	OMIT IF NO GAP between the fuel and clad. A mixture number of zero is often used.
12	END	Terminate latticecell data	Always	END	Terminate latticecell input data by entering the word END.

Table C4.4.4 (continued through page C4.4.11)

Geometry specification for multiregion problems (see Sect. C4.4.6)

Entry Number	Variable Name	Type of Data	Entry Requirement	Data Entry	Comments
1	CS	Type of geometry	Always		Available geometry types are listed below.
				SLAB	Use for slab geometry.
				CYLINDRICAL	Use for cylindrical geometry.
				SPHERICAL	Use for spherical geometry.
				BUCKLEDSLAB	Use for slab geometry with a buckling correction for the two transverse directions.
				BUCKLEDCYL	Use for cylindrical geometry with a buckling correction
2	BR	Right boundary condition	Required for BUCKLEDSLAB & BUCKLEDCYL Optional for other geometries		Default is VACUUM. Describes the right/outside boundary condition.
				VACUUM	This provides a nonreturn condition at the boundary.
				REFLECTED	Do not use for cylindrical or spherical.
				PERIODIC	Do not use for cylindrical or spherical.
				WHITE	This provides isotropic return at the boundary.
3	BL	Left boundary condition	Required for BUCKLEDSLAB & BUCKLEDCYL Optional for other geometries		Default is REFLECTED. Describes the left boundary condition.
				VACUUM	This provides a nonreturn condition at the boundary.
				REFLECTED	Recommended for cylindrical or spherical.
				PERIODIC	Do not use for cylindrical or spherical.
				WHITE	This provides isotropic return at the boundary.
4	ORGN	Location of left boundary on the x-axis		Appropriate dimension	Default is 0.0. Must not be changed for cylindrical or spherical geometry.
5	DY	Buckling height (cm)	BUCKLEDSLAB & BUCKLEDCYL	Appropriate dimension	OMIT FOR SLAB, CYLINDRICAL, and SPHERICAL. This corresponds to one of the transverse dimensions of an actual 3-D slab assembly, or to the length of a finite cylinder.
6	DZ	Buckling depth	BUCKLEDSLAB	Appropriate dimension	OMIT UNLESS BUCKLEDSLAB WAS SPECIFIED. This is the buckling depth corresponding to the second transverse dimension of a 3-D slab assembly.
7	END	End geometry parameters	Always	END	Enter the word END.
8	MXZ	Mixture number in the zone	Always	Mixture no.	Repeat items 8-10 until all zones are defined. Enter the mixture number for this zone.
9	RZ	Outside radius of the zone	Always	Appropriate dimension	Repeat items 8-10 until all zones are defined. Enter the outside dimension of the zone (cm).
10	XMOD	External moderator index	Always		Repeat items 8-10 until all zones are defined.
				NOEXTERMOD	No moderatering materials in the adjacent zones.
				ONEEXTERMOD	A moderating material is present in one adjacent zone.
				TWOEXTERMOD	Moderating materials are present in two adjacent zones.
	END ZONE	Terminate zone data		END ZONE	Enter when all zones have been defined by repeating items 8 thru 10 for each zone.

Table C4.4.4 (continued through page C4.4.11)

Summary of available optional parameter data (see Sect. C4.4.7)

Entry Number	Keyword Name	Type of Data	Comments
1	MORE DATA	Input flag	This signals that optional parameter data are to be entered.
2	ISN=	Order of angular quadrature	The default value is 8. This allows using another value.
3	SZF=	Spatial mesh size factor	The default value is 1.0 SZF<1.0 gives a finer mesh SZF>1.0 gives a coarser mesh
4	IIM=	Max. number of inner iterations	The default value is 20. This allows using another value.
5	ICM=	Max. number of outer iterations	The default value is 25. This allows using another value.
6	EPS=	Overall convergence criteria	The default value is 0.0001. This allows using another value.
7	PTC=	Point convergence criteria	The default value is 0.0001. This allows using another value.
8	BKL=	Buckling factor	The default value is 1.420892. Use ONLY for a multiregion problem that specifies BUCKLEDSLAB or BUCKLEDCYL.
9	IUS=	Upscatter scaling factor	The default value is zero. IUS=0 doesn't utilize upscatter scaling. IUS=1 uses upscatter scaling to accelerate the solution and/or speed convergence.
10	RES=	Resonance data	Enter the mixture number, geometry type (SLAB, CYLINDER, SPHERE) and the thickness of the slab or radius of the sphere or cylinder, in cm.
11	DAN(mm)=	Dancoff factor for the specified mixture	Enter the mixture number mm to which the Dancoff factor applies inside the parentheses; enter the Dancoff factor after the equal sign Repeat items 10 and 11 for all resonance mixtures used in the problem that are not treated in the lattice cell or multiregion description.
12	END	Terminus	Terminate the optional parameter data

Table C4.4.4 (continued through page C4.4.11)

Outline of KENO-V data (see Table C4.4.5 and Sect. C4.4.8)

Type of Data	Starting Flag	Comments	Termination Flag
Title	None	80 columns, must be entered first	none
Parameters	READ PARM	Enter desired parameter data	END PARM
Biasing or weighting	READ BIAS	Enter desired biasing data	END BIAS
Geometry	READ GEOM	Enter desired geometry data	END GEOM
Array data or unit orientation	READ ARRAY	Enter desired array data	END ARRAY
Boundary conditions or albedoes	READ BNDS	Enter desired albedo data	END BNDS
Start data or initial source	READ START	Enter desired start data	END START

Outline of SEARCH* data (see Table C4.4.6 and Sect. C4.4.9)

READ SEARCH	enter desired search data	END SEARCH

*Search data are required for CSAS4 and CSAS4X

Table C4.4.5 Summary of KENO-V input data (continued through page C4.4.19)

Summary of parameter data

TITLE The title must be entered first (80 columns) See Section F11.4.3

PARAMETERS Format: READ PARAM enter parameter data here END PARAM
If parameters are entered, they must follow the title. See Sections F11.4.3, F11.5.2 and F11.5.3.

KEY	STD.	DEFINITION	KEY	STD.	DEFINITION	KEY	STD.	DEFINITION	KEY	STD.	DEFINITION
RND=	given	random number	RUN=	YES	execute problem	MKH=	NO	matrix by hole	XSC=	14	mixed xsecs
TME=	30.min	execution time(min)	FLX=	NO	fluxes	CKH=	NO	cofactor k by hole	ALB=	75	albedo
TBA=	0.5min	batch time (min)	FDN=	NO	fission densities	FMH=	NO	fiss. prod. by hole	WTS=	80	weights
WTA=	0.5	average weight	ADJ=	NO	adjoint calculation	HHL=	NO	MKH at highest level	LIB=	0	working xsecs
WTH=	3.0	wt. for splitting	AMX=	NO	all mixture xsecs	MKA=	NO	matrix by array	SKT=	16	scratch
WTL=	1/WTH	Russian Roulette wt.	XAP=	NO	xsec angles & probs.	CKA=	NO	cofactor k by array	RST=	0	read restart
GEN=	103	No. of generations	XS1=	NO	1-D xsecs	FMA=	NO	fiss. prob. by array	WRS=	0	write restart
NPG=	300	No. per generation	XS2=	NO	2-D xsecs	HAL=	NO	MKA at higest level			
NSK=	3	generations skipped	FEI=	NO	fission spectrum	PLT=	YES	printer plots tracks			
RES=	0	gens. between restart	F1D=	NO	extra 1-D xsecs	BUG=	NO	debug print			
NBK=	GEN*25	neutron bank positions	FAR=	NO	fiss & abs.	TRK=	NO	print neutron tracks			
XNB=	0	extra bank entries	MKP=	NO	matrix by location	PWT=	NO	print avg. weight			
NFB=	NPG	fission bank positions	CKP=	NO	cofactor k by loc.	PGM=	NO	unprocessed geometry			
XFB=	0	extra bank entries	FMP=	NO	fiss. prod. by loc.	SMU=	NO	self-multiplication			
X1D=	0	No. of extra 1-D's	MKU=	NO	matrix by unit	NUB=	NO	neutrons per fission			
LNG=	4000000	words of storage	CKU=	NO	cofactor k by unit	PAX=	NO	albedo-xsec array			
BEG=	1	restart at this gen.	FMU=	NO	fiss. prod. by unit						
NBB=	200	blocks for d.a. unit									
NLB=	789	length of d.a. block									

Table C4.4.5 (continued through page C4.4.19)

Summary of array data

ARRAY

Format: READ ARRAY array parameters data type orientation data END ARRAY
See Sections P11.5.5, P11.5.6 and P11.5.7

Repeat the sequence ARRAY PARAMETERS DATA TYPE ORIENTATION DATA for each array used in the problem.

ARRAY PARAMETERS

KEYWORD	DEFAULT	DEFINITION
ARA=	1	No. defining the array
NUX=	1	No. of units in X direction
NUY=	1	No. of units in Y direction
NUZ=	1	No. of units in Z direction
GBL=	maxara	The global, universe or overall array number**
CON=	none	delim comment delim optional comment is a maximum of 132 characters.

** Can be defaulted by the code. If specified, it need be entered only once per problem.

DATA TYPE
FILL
LOOP

ORIENTATION DATA FOR FILL

Enter unit numbers to define every position in the array. When entering data utilizing the options in this table, the count field and option field must be adjacent with no imbedded blanks. The operand field may be separated from the option field by one or more blanks. Fill data for an array ends with END FILL.

COUNT FIELD	OPTION FIELD	OPERAND FIELD	COMMENTS
		j	stores j at the current position in the array
i	R	j	stores j in the next i positions in the array
i	*	j	stores j in the next i positions in the array
i	S	j	stores j in the next i positions in the array
	F	j	fills remainder of the array with unit No. j starting with the current array position
	A	j	sets the current position in the array to j
i	S		increments current position in the array by i allows skipping i positions. The value of i may be positive or negative.
i	Q	j	repeats the previous j entries i times. The default value of i is 1.
i	M	j	repeats the previous j entries i times, inverting the sequence each time. The default value of i is 1.
i	B	j	Starting with the entry at -i from the current position, store entries in inverse order until position -(i+j) is reached. Default value of i=1.
i	P	j	alternately stores j and -j in the next i positions of the array.
i	I	j k	provides i entries linearly interpolated between j and k, and the end points (i.e. a total of i+2 points.) j and k must be separated by at least one blank.
	T		terminates the data reading for the array.

ORIENTATION DATA FOR LOOP

Enter the unit number and nine numbers that define the position(s) of that unit. Data for each of these ten entries are repeated until every position in the array has been defined. Data for an array is terminated by entering END LOOP.

ENTER DATA IN THE FORM:

DATA ENTRY	COMMENTS
LTYPE	The unit or box type. LTYPE must be greater than 1.
IX1	Starting position in the X direction. IX1 must be at least 1 and no larger then the value entered for NUX.
IX2	Ending position in the X direction. IX2 must be at least 1 and no larger than the value of NUX.
INCX	The number of units by which increments are made in the X direction.
IY1	The starting position in the Y direction. IY1 must be at least 1 and less than the value entered for NUY.
IY2	Ending position in the Y direction. IY2 must be at least 1 and no larger than the value of NUY.
INCY	The number of units by which increments are made in the positive Y direction.
IZ1	Starting position in the Z direction. IZ1 must be at least 1 and no larger than NUZ. in the positive Y direction.
IZ2	Ending position in the Z direction. IZ2 must be at least 1 and no larger than NUZ.
INCZ	The number of units by which increments are made in the positive Z direction.

Table C4.4.5 (continued through page C4.4.19)

Summary of biasing data

BIAS (weighting) Format: READ BIAS keyword correlation data auxiliary END BIAS
See Sections F11.4.7 and F11.5.8

KEYWORD	DESCRIPTION
ID=	CORRELATION DATA will be read next.
	id material ID. Enter from table at right. to use weighting data from the library.
	ibgn beginning bias ID
	iend ending bias ID
WT=	AUXILIARY DATA will be read next.
WTS=	AUXILIARY DATA will be read next.
	wttitl material title
	id material ID.
	nsets number of sets of group structures
	REPEAT THKINC, NUMINC, NGPWT, WTAVG NSETS TIMES
	thkinc thickness per increment
	numinc number of increments
	ngpwt number of energy groups for this set of wts
	wtavg enter numinc* ngpwt values of wtavg.

MATERIAL	ID	ENERGY GROUPS	THICKNESS/INCREMENT
concrete	301	16, 27, 123	5 cm
paraffin	400	16, 27, 123	3 cm
water	500	16, 27, 123, 218	3 cm
graphite	6100	16, 27, 123	20 cm

For CORRELATION DATA, the Material ID is chosen from material ID column above (the keyword is ID=).
For AUXILIARY DATA, the material ID is chosen by the user and the keyword is WT= or WTS=.

Beginning and ending bias ID's are defined by the user. The geometry specification that has the bias ID equal to the beginning bias ID utilizes the wt avg's from the first interval of material ID.

Table C4.4.5 (continued through page C4.4.19)

Summary of boundary condition data

BNDS (albedo or boundary conditions)

Format: READ BNDS face code albedo name END BNDS
See Section F11.4.7

The sequence FACE CODE ALBEDO NAME is entered as many times as necessary to define the appropriate albedo boundary conditions. The default for all faces is vacuum.

FACE CODES FOR ENTERING BOUNDARY (ALBEDO) CONDITIONS

FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION
+XB=	positive X face	XFC=	both X faces	+YX=	positive X and Y faces	&XY=	positive Y and X faces
&XB=	positive X face	YFC=	both Y faces	&XY=	positive X and Y faces	-XY=	negative X and Y faces
-XB=	negative X face	&FC=	both X faces	&YX=	positive X and Y faces	-XX=	negative X and X faces
+YB=	positive Y face	+FC=	all positive faces	+XX=	positive X and X faces	-YX=	negative Y and X faces
&YB=	positive Y face	&FC=	all positive faces	+&X=	positive X and X faces	YXF=	all X and Y faces
-YB=	negative Y face	-FC=	all negative faces	&XX=	positive X and X faces	XXF=	all X and X faces
+XB=	positive X face	XYF=	all X and Y faces	&&X=	positive X and X faces	XYF=	all Y and X faces
&XB=	positive X face	XXF=	all X and X faces	+YX=	positive Y and X faces	-YX=	negative X and Y faces
-XB=	negative X face	YXF=	all Y and X faces	+&Y=	positive Y and X faces	-&X=	negative X and X faces
ALL=	all 6 faces	+XY=	positive X and Y faces	&YX=	positive Y and X faces	-XY=	negative Y and X faces

ALBEDO NAMES AVAILABLE ON THE KENO V ALBEDO LIBRARY, FOR USE WITH THE FACE CODES

ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION
DPOH2O DPOH2O DP0 DPO	12 inch double P0 water differential albedo with 4 incident angles	CONC-4 CON4 CONC4	4 inch concrete differential albedo with 4 incident angles	VACUUN VOID VACU VAC	vacuum condition
H2O WATER	12 inch water differential albedo with 4 incident angles	CONC-8 CON8 CONC8	8 inch concrete differential albedo with 4 incident angles	SPECULAR MIRROR MIRR SPEC SPE MIR	mirror image reflection
PARAFFIN PARA WAX	12 inch paraffin differential albedo with 4 incident angles	CONC-12 CON12 CONC12	12 inch concrete differential albedo with 4 incident angles		
CARBON GRAPHITE C	200 cm. carbon differential albedo with 4 incident angles	CONC-16 CON16 CONC16	16 inch concrete differential albedo with 4 incident angles	PERIODIC PERI PER	periodic boundary condition
ETHYLENE POLY CH2	12 inch polyethylene differential albedo with 4 incident angles	CONC-24 CON24 CONC24	24 inch concrete differential albedo with 4 incident angles		

Table C4.4.5 (continued through page C4.4.19)

Summary of geometry data

GEOMETRY (region)

Format: READ GEOM enter geometry region data here END GEOM
See Sections F11.4.4, F11.5.1.2, F11.5.6 and F11.5.7

GEOMETRY REGION DATA consists of SIMPLE GEOMETRY REGION DATA and EXTENDED GEOMETRY REGION DATA.

ENTER GEOMETRY REGION DATA IN THE FOLLOWING FORM:

OPTIONAL GLOBAL SPECIFICATION
UNIT n
OPTIONAL GEOMETRY COMMENT
SIMPLE GEOMETRY REGION DATA
  and/or
EXTENDED GEOMETRY REGION DATA

* *

ENTER SIMPLE GEOMETRY REGION DATA IN THE FOLLOWING FORM:

GLOBAL Enter only to specify this unit as the global unit.
UNIT n
COM=delim comment delim This optional comment can be up to 132 characters.
                         It must begin and end with a delimiter.
fgeom mix no. bias ID dimensions optional origin data (ORIGIN coordinates) optional chord data (CHORD)

Enter as many geometry description specifications as necessary to describe the unit and as many units as necessary to describe the system.

SIMPLE GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA	TYPE 4 DATA	TYPE 5 DATA	TYPE 6 DATA
fgeom	SPHERE HEMISPHERE HEMISPHERE+X HEMISPHERE-X HEMISPHERE+Y HEMISPHERE-Y HEMISPHERE+Z HEMISPHERE-Z	XCYLINDER XHEMICYL+Y XHEMICYL-Y XHEMICYL+Z XHEMICYL-Z	YCYLINDER YHEMICYL+X YHEMICYL-X YHEMICYL+Z YHEMICYL-Z	CYLINDER ZCYLINDER ZHEMICYL+X ZHEMICYL-X ZHEMICYL+Y ZHEMICYL-Y	CUBE	CUBOID
dimensions	R (radius)	R +H -H	R +H -H	R +H -H	+X -X	+X -X +Y -Y +Z -Z
optional origin coord.	Enter the X Y Z coord. of origin	Enter the Y Z coord. of center-line	Enter the X Z coord. of center-line	Enter the X Y coord. of center-line	omit	omit
optional chord data**	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	omit	omit

**Chord data is not applicable for SPHERE XCYLINDER YCYLINDER CYLINDER or ZCYLINDER

Table C4.4.5 (continued through page C4.4.19)

Summary of geometry data (continued)

GEOMETRY (region) (cont.)

ENTER EXTENDED GEOMETRY DATA IN THE FOLLOWING FORM:

fgeom ref. ID bias ID thickness per region origin coordinates nreg

EXTENDED GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA
fgeom	ARRAY CORE COREBDY COREBNDS COREBOUN	HOLE	REPLICATE REFLECTOR
ref. ID	array no.	emplaced unit number	mixture no. in generated regions
bias ID	omit for ARRAY	omit	first bias ID
thick./reg.	omit	omit	variable**
origin coord.	Enter the X Y Z coord. of most neg. pt. of array	Enter the X Y Z coord. of origin	omit
nreg	omit	omit	No. of regions to be generated

**The number of dimensions to be entered is the same as the region preceding the replicate or reflector specification because the generated regions have that shape. The value of the dimensions is the thickness of each generated region of material on that surface.

Table C4.4.5 (continued through page C4.4.19)

Summary of plot data

PLOT Format: READ PLOT plot parameters END PLOT
plot parameters must be entered for each plot that is to be made.
See Sections F11.4.11 and F11.5.9

KEYWORD	DEFAULT	DEFINITION
TTL=	prev. title	delim ptitl delim A one character delimiter signals the beginning and end of the title. ptitl is the plot title.
PIC=	MAT	Type of picture: MIXTURE, UNIT NO. or BIAS ID NO.
		MIXTURE ------- MAT
		MIX
		MIXT
		MIXTURE
		MEDI
		MEDIA
		UNIT NO. ------- BOX
		BOXT
		BOXTYPE
		UNT
		UNIT
		UNITTYPE
		BIAS ID NO. ---- IMP
		BIAS
		BIASID
		WTS
		WEIG
		WEIGHTS
		WGT
		WGTS
XUL=	prev. plot	X coord. of upper left corner of plot
YUL=	prev. plot	Y coord. of upper left corner of plot
ZUL=	prev. plot	Z coord. of upper left corner of plot
XLR=	prev. plot	X coord. of lower right corner of plot
YLR=	prev. plot	Y coord. of lower right corner of plot
ZLR=	prev. plot	Z coord. of lower right corner of plot

KEYWORD	DEFAULT	DEFINITION
UAX=	prev. plot 0 IF VAX OR WAX is read	X component of direction cosine for the AX axis of the plot (across)
VAX=	prev. plot 0 IF UAX OR WAX is read	Y component of direction cosine for the AX axis of the plot (across)
WAX=	prev. plot 0 IF UAX OR VAX is read	Z component of direction cosine for the AX axis of the plot (across)
UDN=	prev. plot 0 IF VDN OR WDN is read	X component of direction cosine for the DN axis of the plot (down)
VDN=	prev. plot 0 IF UDN OR WDN is read	Y component of direction cosine for the DN axis of the plot (down)
WDN=	prev. plot 0 IF UDN OR VDN is read	Z component of direction cosine for the DN axis of the plot (down)
DLX=		horizontal spacing between points on plot
DLD=		vertical spacing between points on plot
MAX=		no. of intervals to be printed across page
MDN=		no. of intervals to be printed down page
LPI=	8	lines per inch printed down the page
MCH=	CHRS**	delim CHRS delim a one character delimiter signals the beginning and end of the character string.
RUN=	YES	YES allows the problem to execute NO terminates problem after data checking
PLT=	YES	YES allows the plot(s) to be made NO allows reading the plot data without making a plot.

PLOT ORIGIN:

(1) SINGLE UNIT-coincides with origin of geometry description.

(2) BARE ARRAY-at the most negative point of the array, (lower left-hand back corner of the global array).

(3) REFLECTED ARRAY- coincides with the origin of the CORE or ARRAY description of the global array.

**default values of CHRS are given below.

MEDIA	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CHRS		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M
MEDIA	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42			
CHRS	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	#	,	$	-	+	)	|			
MEDIA	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58							
CHRS	&	>	:	;	.	-	%	*	"	=	!	(	@	<	/	0							

Table C4.4.5 (continued through page C4.4.19)

Summary of starting data

START Format: READ START enter start data here END START
The default value of start type is zero.
See Section P11.4.8

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
0	none	NST	uniform
		XSM	
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
1	NST	XSM	cosine
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
2	NST	XSM	cosine with
	NXS	XSP	fraction in
	NYS	YSM	specified
	NZS	YSP	unit
	FCT	ZSM	
		ZSP	
		RFL	
		PSP	

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
3	NST	KFS	spike
	TFX	PSP	
	TFY		
	TFZ		
	NXS		
	NYS		
	NZS		
4	NST	KFS	multiple
	TFX	PSP	spikes
	TFY		
	TFZ		
	NBX		
5	NST	PSP	in specified
	NBX		units
6	NST	NXS	arbitrary
	TFX	NYS	points
	TFY	NZS	
	TFZ	KFS	
	LNU*	PS6	
		PSP	

*LNU must be the last entry for each set of start 6 data. The LNU of each successive set of data must be larger than the last.

KEYWORD	DEFAULT	DEFINITION
NST=	0	start type
TFX=	0.0	X coordinate
TFY=	0.0	Y coordinate
TFZ=	0.0	Z coordinate
NXS=	0	X index of unit pos.
NYS=	0	Y index of unit pos.
NZS=	0	Z index of unit pos.
KFS=		fission spectra
LNU=	0	number of last neutron
NBX=	0	source unit number
FCT=	0	fraction
XSM=	-X	-X of source cuboid
XSP=	+X	+X of source cuboid
YSM=	-Y	-Y of source cuboid
YSP=	+Y	+Y of source cuboid
ZSM=	-Z	-Z of source cuboid
ZSP=	+Z	+Z of source cuboid
RFL=	NO	start in reflector
PS6=	NO	print start 6 input
PSP=	NO	print starting points

Table C4.4.6 Summary of SEARCH input data

Outline of Search Type Specification			
Data Position	Type of Data	Data Entry	Comments
1.	Search Descriptor	OPTIMUM OPT OPTM	Initiates a search for the maximum value of k-effective.
		CRITICAL CRTL	Initiates a search for a specified value of k-effective.
		MINIMUM MINM MIN	Initiates a search for the minimum value of k-effective.
2.	Search Type	PITCH PIT PTCH	Vary the pitch
		DIMENSION DIM DMSN	Vary one or more dimensions in one or more regions.
		CONCENTRATION CON	NOT IMPLEMENTED
3.	Optional search parameters		Optional search parameters allow changing default values. Any or all may be entered in any order.
3a.	No. of search passes	PAS=	Enter the keyword PAS= followed by the desired number of search passes. default=10.
3b.	No. of search parameters	NPM=	Enter the keyword NPM= followed by the number of search parameters. Present capability is limited to 1. default=1.
3c.	Search convergence tolerance	EPS=	Enter the keyword EPS= followed by the desired convergence tolerance. default=.005
3d.	Desired value of k-effective	KEF=	Enter the keyword KEF= followed by the desired value of k-effective. default =1.000 DO NOT ENTER FOR OPTIMUM OR MINIMUM SEARCHES.
4.	Additional search data	MORE	Enter the delimiter MORE This delimiter ends the optional search parameters and initiates the individual search commands found in Table C4.4.16.

Table C4.4.6 (continued)

Outline of individual search commands (continued)

Entry Number	Keyword Name	Type of Data	Entry Requirements	Comments
1	MAXPITCH=	Maximum allowed pitch	PITCH	This is used to specify the maximum allowed pitch for the search. The default value is the pitch corresponding to -5*the parameter at the minimum possible pitch.
2	MINPITCH=	Minimum allowed pitch	PITCH	This is used to specify the minimum allowed pitch for the search. The default value is the minimum possible pitch (i.e. the pitch at which the units in the array touch).
3a	ALTER ALTR ALT CHANGE CHNG MODIFY MOD MDFY	Begin a new search command	DIMENSION PITCH	These words are used to specify that modifications will be made to the geometry according to subsequent commands (items 5 through 8 as required to specify the desired changes).
3b	MAINTAIN MNTN	Begin a new search command	DIMENSION PITCH	The spacing (thickness) of the specified geometry regions will be maintained when the interior regions grow or shrink.
3c	KEEP HOLD	Begin a new search command	DIMENSION PITCH	This command resets the specified geometry to the original input specifications.
4	PAR=	Parameter number	DIMENSION PITCH	Enter the parameter number the current command (ALTER, MAINTAIN, KEEP) applies to. The present capability is limited. Default=1 and should not be overridden.
5	UNIT=	Unit the current command applies to	DIMENSION PITCH	Enter the unit in which regions are to be altered. Default=1.
6	REG=	First region to be altered in the unit	DIMENSION PITCH	Enter the first or only region in the unit which the search constants(items 8a, b, c and/or d) apply to.
7	TO	Last region to be altered in the unit	DIMENSION PITCH	Enter the last region in the unit which the search constants (items 8a, b, c and/or d) apply to. Default is the first region. Note--Item 6 must be entered in order to alter a single region in a unit. Both items 6 and 7 must be entered in order to alter more than one region in a unit.
8a	ALL=	Search constant for all surfaces (faces) of the region(s).	DIMENSION PITCH	Enter a value for the search constants for the specified regions. This value will be applied to all surfaces of the region(s).
8b	+X=	Search constant for +X face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the +X face of a cuboid.
	-X=	Search constant for -X face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the -X face of a cuboid.
	+Y=	Search constant for +Y face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the +Y face of a cuboid.
	-Y=	Search constant for -Y face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the -Y face of a cuboid.
	+Z=	Search constant for +Z face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the +Z face of a cuboid.
	-Z=	Search constant for -Z face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the -Z face of a cuboid.
8c	+R=	Search constant for radius	DIMENSION PITCH	Enter a value for the search constant for the radius of a sphere or a cylinder.
	+H=	Search constant for + height	DIMENSION PITCH	Enter a value for the search constant for the + height of a cylinder.
	-H=	Search constant for - height	DIMENSION PITCH	Enter a value for the search constant for the - height of a cylinder.
8d	CH=	Search constant for chord	DIMENSION PITCH	Enter a value for the search constant for the chord face of a hemisphere or hemicylinder.
9	+CON=	Maximum constraint	DIMENSION PITCH	Enter the maximum constraint for the current parameter. default=+10E10
0	-CON=	Minimum constraint	DIMENSION PITCH	Enter the minimum constraint for the current parameter. default=-10E10

C4.4.2 INPUT DATA REQUIREMENTS

Table C4.4.7 summarizes the input data required by Criticality Safety Analysis Sequence No. 4. The types of data needed are briefly discussed after the table.

Table C4.4.7 Summary of data input requirements

	Type of Data	Analytical Sequence							
1	Analytical Sequence Specification (IBM versions)	=CSAS1X	=CSASI	=CSASIX	=CSASN	=CSAS25	=CSAS2X	=CSAS4	=CSAS4X
2	Material Information Processor Data	yes	yes	yes	yes	yes	yes	yes	yes
3	KENO V data	no	no	no	no	yes	yes	yes	yes
4	Search data	no	no	no	no	no	no	yes	yes
5	END for the Analytical Sequence (IBM versions)	yes	yes	yes	yes	yes	yes	yes	yes

1. ANALYTICAL SEQUENCE SPECIFICATION...For IBM versions, enter the desired sequence name. The available options are: =CSAS4, =CSAS4X, =CSAS25, =CSAS2X, =CSAS1X, =CSASI, =CSASIX, =CSASN. Enter PARM=CHK in column 11 to check the data.
2. MATERIAL INFORMATION PROCESSOR DATA...These data are required for processing the cross sections for use in the problem. These data are described in detail in Sects. C4.4.3 through C4.4.7.
3. KENO-V DATA...These data are described briefly in Sect. C4.4.8. A detailed description is found in Sect. F11.4 with additional information in Sect. F11.5.
4. SEARCH DATA...Search data are required for CSAS4 and CSAS4X. A detailed description is given in Sect. C4.4.9. The data required for a simple optimum pitch search are:

   READ SEARCH OPTIMUM PITCH END SEARCH

5. END FOR ANALYTICAL SEQUENCE...For IBM versions, enter END, starting in column 1.

C4.4.3 MATERIAL INFORMATION PROCESSOR DATA

The Material Information Processor reads the standard compositions specification data and the cell geometry specifications. It then produces the mixing table and unit cell information necessary for processing the cross sections. Table C4.4.8 outlines the input data for the Material Information Processor.

The input data for the Material Information Processor are identical to the input for CSAS1, Sect. C1.6, with the following exceptions:

- The analytical sequence card is considered to be part of the overall data input.
- Items 2, 3, 4, 6, and 7 from Sect. C1.6 are omitted from the parameter card.
- The words END COMP are entered to signal the completion of the standard compositions specification data.
- The multiregion zone description data are terminated by the words END ZONE.
- If optional control parameters are entered, the words MORE DATA must precede the first optional parameter data entry.

Table C4.4.8 Outline of material information processor data

Data Position	Type of data	Data entry	Comments
1	Title	Enter a title.	80 characters
2	Cross-section library name	HANSEN-ROACH, 27GROUPNDF4, 123GROUPGMTH, 218GROUPNDF4, etc.	These are the currently available libraries. See Sect. C1.A.1.
3	Type of calculation	INFHOMMEDIUM LATTICECELL MULTIREGION	These are the available options. See the explanation in Sect. C4.4.3.
4	Standard compositions specification data	Enter the appropriate data.	Terminate this data block with END COMP. See Table C4.4.4.
5	Cell geometry specification	Enter the appropriate data. (Omit for INFHOMMEDIUM.)	Omit for INFHOMMEDIUM. See Table C4.4.5 for LATTICECELL. See Table C4.4.7 for MULTIREGION.
6	Optional parameter data	Enter the desired data.	Precede this data block by MORE DATA if more parameter data are to be entered. Otherwise, omit these data entirely. See Sect. C4.4.7.

The types of data in Table C4.4.8 are discussed briefly here.

1. TITLE...An 80-character title is required. The title is the first 80 characters of the Material Information Processor data.
2. CROSS-SECTION LIBRARY NAME...This item specifies the cross-section library that is to be used in the calculation. See Sect. C1.A.1 for a list of the available libraries.
3. TYPE OF CALCULATION...The options are INFHOMMEDIUM, LATTICECELL, and MULTIREGION.

   INFHOMMEDIUM is used for an infinite homogeneous medium. It can be used to provide cross sections to KENO V if the KENO-V geometry does not involve a fuel assembly or the geometry cannot be described in one dimension. The cross sections are treated as if each mixture is infinite. Thus, the self-shielding calculations will not account for any geometrical effects.

LATTICECELL is used when the geometry can be described as a lattice. It is especially suited for arrays of cylindrical rods or spherical pellets. The use of LATTICECELL requires the entry of additional geometry information (cell geometry specification), which is described in Table C4.4.10. When the problem consists of an array of fuel bundles or similar geometry, the cross sections are corrected for both geometric and resonance self-shielding. If cell-weighted cross sections are desired, CSAS2X or CSAS4X must be used. A flux disadvantage factor is then applied to the cross sections of the materials that are used in the unit cell. The KENO-V geometry description may represent a fuel bundle discretely, or as a homogeneous region. By using CSAS2X or CSAS4X, a fuel bundle can be represented as a single homogeneous region by specifying a mixture number of 500 on the KENO-V geometry card that defines the overall size and shape of the bundle. This utilizes the cell-weighted cross sections that represent the spatial behavior of the unit cell. The use of CSAS2X or CSAS4X precludes using individual mixtures from the unit cell in the KENO-V geometry.

MULTIREGION may be used to define a geometric configuration that is more complicated than that allowed by LATTICECELL. It can also be used for a system involving large geometric regions where geometry effects may be minimal. The cross sections utilized in the unit cell are corrected for resonance self-shielding, but the geometric correction uses a Dancoff factor of 0, thus ignoring the effects due to a lattice geometry. The cell-weighted mixture that is generated for CSAS2X and CSAS4X results from homogenizing the unit cell. The additional geometry information (Cell Geometry Specification) required for a MULTIREGION problem is described in Table C4.4.11.

4. STANDARD COMPOSITIONS...See Sect. C4.4.4, and Table C4.4.9, for a description of the standard compositions specification data. These data are required for every problem.
5. CELL GEOMETRY SPECIFICATION...See Sect. C4.4.5, and Table C4.4.10, for an explanation of the data associated with LATTICECELL problems. Section C4.4.6 explains the data required for a MULTIREGION problem.
6. OPTIONAL PARAMETER DATA...Omit these data unless they are needed. If data are to be entered, see Sect. C4.4.7, and Table C4.4.12, for assistance.

C4.4.4 STANDARD COMPOSITIONS SPECIFICATION DATA

The standard compositions specification data are used to define mixtures that are to be utilized in the problem. These data can be considered to be a mixing table that defines mixtures in terms of standard compositions specification data (available from Table M8.2.1). The required input for the standard compositions specification data varies, depending on the standard composition component name. However, every standard composition specification must include the following:

1. A standard composition component name (SC),
2. The mixture that contains this component (MX),
3. A terminator for the standard compositions specification data (enter the word END).

Enter as many standard composition specifications as necessary to define all the mixtures needed for the problem. When all the necessary mixtures have been described, enter the words END COMP to signal the completion of this type of data.

The data required to define a standard composition specification are contained in Table C4.4.9. "ARBM" is used to denote entries required only for arbitrary materials, and "SOLN" is used for entries required only for solutions.

Table C4.4.9 Outline of standard composition specification data

Entry Number	Variable Name	Type of data	Entry Requirement	Comments
1	SC	Standard composition component name	Always	Enter once for each standard composition. Enter the alphanumeric description from Table M8.2.1. Additional allowed names include those beginning with arbitrary materials, and SOLN for solutions.
A1	ROTH	Theoretical density of material (g/cc)	ARBM	Enter once for each standard composition component that is an arbitrary material.
A2	NEL	Number of elements in the material	ARBM	Enter once for each standard composition component that is an arbitrary material.
A3	IVIS	Multiple isotope indicator	ARBM	Enter once for each standard composition component that is an arbitrary material. Enter 0 for a material that does not contain multiple isotope elements. Enter 1 if the material contains a multiple isotope element. An arbitrary material cannot have more than one multiple isotope element. If necessary, several arbitrary materials may be used in a single mixture.
A4	ICP	Compound indicator	ARBM	Enter once for each standard composition component that is an arbitrary material. Enter 1 for a compound, 0 for alloys, mixtures, etc.
A5	IRS	Resonance indicator	ARBM	Enter once for each standard composition component that is an arbitrary material. Enter 0 if all the nuclides in the material are nonresonance nuclides; enter 1 if one or more have resonance data. See Section C1.A.3, Table C1.A-2.
A6	NCZA	ID number (from far right column of Table M8.2.1)	ARBM	Repeat the sequence A6 and A7 for each element in the arbitrary material before entering item 2. Enter the number from the far right column of Table M8.2.1. Only elements can be used in an arbitrary material.
A7	ATPM	Number of atoms of this element per molecule of arbitrary material OR Weight percent of this element in this arbitrary material	ARBM & ICP=1 ARBM & ICP=0	Repeat the sequence A6 and A7 for each element in arbitrary material before entering item 2. Do not enter a value unless ICP=1. Repeat the sequence A6 and A7 for each element in the arbitrary material before entering item 2. Do not enter a value unless ICP=0.
2	MX	Mixture ID number	Always	Enter once for each standard composition component.
S1	FD	Fuel density (grams of U or Pu per liter of solution)	SOLN	Enter once for a solution.
S2	AML	Acid molarity of the solution	SOLN	Enter once for a solution. AML=0 if there is no acid in the solution.
3	VF	Volume fraction (ratio of actual to theoretical density of the material)	See comment column	Default value is 1.0. This item can be omitted if items 4, 5, 6a, and 6b are also omitted.
4	ADEN	Number density (atoms/barn-cm) for the nuclide	VF=0	Enter only if VF=0.0.
5	TEMP	Temperature, in degrees K	See comment column	Default value is 293 K. This item can be omitted if items 6a and 6b are also omitted.
6a	IZA	Isotope's ZA number	VF=0	Enter for each isotope in the standard composition component. Omit if VF=0. Items 6a and 6b are entered in pairs until each isotope in the component is defined.
6b	WTP	Weight percent of the isotope	VF=0	Enter for each isotope in the standard composition component. Omit if VF=0.0. Items 6a and 6b are entered in pairs until each isotope in the component is defined.
7	END	Terminate a standard composition	Always	Enter once for each standard composition component. This terminates the data for a standard composition component. Enter END to terminate the component. Repeat items 1-7 until all the mixtures have been defined.
	END COMP	terminate the data block	Terminus	Enter once for a problem. Enter the words END COMP when all the standard composition components have been described.

Note: At least two blanks should separate the last entry 7 from the END COMP.

1. SC STANDARD COMPOSITION COMPONENT NAME...The names of the standard compositions components (the alphanumeric identifiers) may be (1) chosen from Table M8.2.1, the list of elements, compounds, and alloys found in the Standard Composition Library, (2) they may be chosen from the table of available solutions, Sect. M8.3, or (3) they may be designated as an arbitrary material (ARBM). An error message will be printed if the user enters an invalid standard composition component name.

   For standard compositions taken from the Standard Composition Library, certain data such as the volume fraction or theoretical density and other engineering type data may need to be entered. Table C1.A.2 lists the ID's for nuclides that require a resonance self-shielding calculation. For standard compositions containing more than one isotope of an element (such as UO_2), the user is free to specify the weight percent for each isotope, such that they total 100%.

   Solutions require the specification of the density of the heavy metal in the solution and the acid molarity of the solution. The alphanumeric specification of a solution starts with the four characters "SOLN." Some common fissile solutions are SOLNUO2(NO3)2, SOLNPU(NO3)4, and SOLNUO2F2.

   See Sect. M8.3 for a complete listing of the available solutions.

   Arbitrary materials require the user to provide all the information normally found in the Standard Composition Library. The arbitrary material option allows specifying a compound or alloy not available in the Standard Composition Library.

   An arbitrary material name must start with the four characters "ARBM." A maximum of twelve characters is allowed for the Standard Composition Component name. Imbedded blanks are not allowed in a standard composition component name.

   An example of an arbitrary material specification is given below:

   Consider a mixture of boral having 35 wt % B_4C, 65 wt % Al and an overall density of 2.64 g/cc. If neither BORAL nor B4C were available in the Standard Composition Library, the components BORON, C, and AL could be used to describe the boral. The necessary volume fractions would have to be determined as shown in Sect. C1.B. By utilizing the arbitrary material option, both B_4C and Al can be described as a component of the same mixture (assume it is desired to be mixture 3), both have a density of 2.64 g/cc and the volume fractions are 0.35 for the B_4C and 0.65 for the Al. The resulting standard compositions specification data do not require any calculations and would be:

   ```
   ARBMTL-B4C  2.64 2 1 1 0 5000 4 6012 1 3 0.35 END
   ARBMTL-AL   2.64 1 0 0 0 13027 100.0 3 0.65 END
   ```

A1. ROTH THEORETICAL DENSITY OF MATERIAL...Enter ONLY for arbitrary materials. The density of the arbitrary material is entered in units of g/cc.

A2. NEL NUMBER OF ELEMENTS IN THE MATERIAL...Enter ONLY for arbitrary materials. Enter the number of components from the Standard Composition Library that are to be used to define this arbitrary material. In the above example, the ARBMTL-B4C contains two components, the B and the C. Therefore, NEL is 2 for the B_4C arbitrary material. The ARBMTL-AL contains only one component, so NEL is 1 for it.

A3. IVIS MULTIPLE ISOTOPE INDICATOR...Enter ONLY for an arbitrary material. An arbitrary material cannot have more than one multiple isotope element. If more than one is needed, define another arbitrary material. Enter a zero if none of the elements in the arbitrary material have multiple isotopes. In the above example, the boron is a multiple isotope element, so IVIS is 1 for the B_4C arbitrary material. The arbitrary material describing the aluminum contains no multiple isotope elements, so IVIS is zero for it.

A4. ICP COMPOUND INDICATOR...Enter ONLY for arbitrary materials. A compound is a substance containing more than one constituent element and having properties different from those which their constituents had as elementary substances. The composition of a pure compound is perfectly definite and is always the same, regardless of the method of formation. Examples of compounds are: H_2O, B_4C, CO_2, $Ca(OH)_2$, C_2H_6, Fe_2S_3, Cr_2O_3, etc.

ICP is 1 for a compound. ICP is 0 for alloys (stainless steel, monel, etc.), and conglomerates or mixtures such as concrete.

In the B_4C arbitrary material example, ICP is 1; in the aluminum arbitrary material example, ICP is 0.

A5. IRS RESONANCE INDICATOR...Enter ONLY for arbitrary materials. If any of the nuclides in the arbitrary material have resonance data associated with them, set IRS to 1. If all the nuclides are nonresonance nuclides, set IRS to zero. Table M8.2.1 lists the nuclides that have resonance data associated with them.

IRS is zero for both the B_4C and Al arbitrary material examples.

A6. NCZA ID NUMBER... Enter ONLY for arbitrary materials. This is the "ZA" ID number for the element or isotope. Usually, NCZA=A+1000*Z, where A is the atomic mass or weight of the nuclide, and Z is the atomic number. For example, NCZA is 1001 for hydrogen, 8016 for oxygen, and 92235 for ^{235}U. For elements for which the natural abundance is acceptable, NCZA can be entered as 1000*Z. For example, 92000 for uranium or 5000 for boron.

If the arbitrary material contains an element that has multiple isotopes, that element MUST be the first element specified in the arbitrary material. Furthermore, that element MUST be listed in the Isotope Distribution Table, Table M8.4.1. Uranium (92000), plutonium (94000), and boron (5000) are multiple isotope elements.

For example: ARBMTL-B4C 2.64 2 1 1 0 5000 4 6012 1 3 0.35 END is correct; ARBMTL-B4C 2.64 2 1 1 0 6012 1 5000 4 3 0.35 END is incorrect because the multiple isotope element is not listed first.

REPEAT THE SEQUENCE NCZA and ATPM (A6 and A7) FOR EVERY ELEMENT IN THE ARBITRARY MATERIAL.

A7. ATPM (1) NUMBER OF ATOMS PER MOLECULE...Enter ONLY for arbitrary materials. If the arbitrary material is not a compound (entry A4, ICP is 0), go to (2) below. If the arbitrary material is a compound (entry A4, ICP is 1), enter the number of atoms of this element per molecule of arbitrary material. Repeat the sequence A6 and A7 for every element in the arbitrary material.

(2) WEIGHT PERCENT OF THIS ELEMENT IN THE ARBITRARY MATERIAL...Enter ONLY for arbitrary materials. If the arbitrary material is a compound (entry A4, ICP is 1), omit this entry. If the arbitrary material is not a compound (entry A4, ICP is 0), enter the weight percent of this element in the arbitrary material. If ICP is 0, the sum of all the weight percents for the arbitrary material MUST be 100.0. Repeat the sequence A6 and A7 for every element in the arbitrary material before going on to item 2.

2. MX MIXTURE ID NUMBER...A mixture number is required on every standard composition specification. It defines the mixture that contains the material defined by the standard compositions specifications data. The mixture numbers are utilized in the Unit Cell Specification for LATTICECELL problems, the Geometry Specification for MULTIREGION problems, and the KENO V geometry data.

S1. FD FUEL DENSITY...Enter ONLY for a solution. The Standard Composition Component name for a solution starts with the four characters "SOLN." The available solution names are listed in Sect. M8.3. Enter the fuel density in units of grams of heavy metal (fissile material) per liter of solution.

S2. AML ACID MOLARITY...Enter ONLY for a solution. If there is no acid in the solution, enter zero. If acid is present in the solution, enter the molarity of the solution.

3. VF VOLUME FRACTION...The default value of the volume fraction is 1.0. It can be omitted if items 4, 5, 6a, and 6b are also omitted. The volume fraction can be interpreted as:

1. the volume fraction of this standard composition component in the mixture,
2. the density of the standard composition component in this application, divided by the theoretical density listed in the Standard Composition Library (Sect. M8), or
3. a product of (1) and (2).

For example, assume an homogenized mixture representing the water moderator and zircalloy cladding around a fuel pin is to be described. If the volume of the clad is 5.32 cc and the volume of the water moderator is 44.68 cc, the mixture can be described using H_2O with a volume fraction of 0.8936 and ZIRCALLOY with a volume fraction of 0.1064.

For UO_2 fuel at 95% theoretical density, the volume fraction is 0.95. For the coolant water in an operating PWR, the volume fraction could be 0.71 (cf., Table C1.A.1).

To describe a mixed oxide fuel pin with a density of 10.5 g/cc, that is 17.8 wt % PuO_2 and 82.2 wt % UO_2, two standard compositions specifications must be entered: one for the PuO_2 component and one for the UO_2 component. From the Standard Composition Library, Sect. M8, the theoretical density of the PuO_2 is 11.46 g/cc, and that of the UO_2 is 10.96 g/cc. Therefore, the volume fraction for the PuO_2 is 0.1631 (i.e., 0.178*10.50/11.46), and the volume fraction for the UO_2 is 0.7875 (i.e., 0.822*10.50/10.96).

Occasionally, a special mixture will be desired for which the determination of the component volume fractions will not be so straightforward. Consider a mixture of boral having 35 wt % B_4C, 65 wt% Al, and an overall density of 2.64 g/cc. Assume that neither BORAL nor B_4C are in the Standard Composition Library. The volume fractions corresponding to BORON, C, and AL may be computed as illustrated in Appendix C1.B. Note that the calculation is complicated by the fact that B_4C is a compound and boron has two isotopes. It might be simpler to enter B_4C and Al as arbitrary materials as in the earlier example. IF THE VOLUME FRACTION IS SET TO 0.0, THE USER CAN ENTER THE NUMBER DENSITY FOR THE NUCLIDE (in atoms per barn-cm) FOR ITEM 4. THIS OPTION CANNOT BE USED FOR STANDARD COMPOSITIONS THAT REPRESENT MULTIPLE NUCLIDES.

4. ADEN NUMBER DENSITY...Enter a value for the number density ONLY if the volume fraction (VF) was entered as zero. The number density is entered in units of atoms per barn-cm.

5. TEMP TEMPERATURE...The default value of the temperature is 293°K. The temperature can be omitted if items 6a and 6b are also omitted.

The temperature is used for Doppler broadening and/or the selection of the proper set of thermal scattering data.

The resonance self-shielding calculation performed by NITAWL uses the Nordheim Integral Method and can account for Doppler broadening of the resonances at any specified temperature. If this material is known to contain a resonance nuclide, the user should enter a rough estimate of the temperature of the material. To determine whether resonance data are available for any of the nuclides in a material,

1. check Table M8.2.1 if item 1 is not a solution, SOLN;
2. check Table M8.3.1 if item 1 denotes a solution, SOLN.

For each of the light nuclides ($A \leq 20$ a.m.u.), the master cross-section library may contain one or more sets of thermal scattering data, each set corresponding to a different temperature. Scattering matrices for nuclides in media at elevated temperatures are generally fuller than those at lower temperatures. Tables M8.2.1 and M8.3.1 indicate the isotopes, elements, compounds, solutions, and alloys for which multiple sets of thermal scattering data are currently available. If multiple sets of thermal scattering data are available, the user should enter a rough estimate of the temperature of the material. The code will then use the set of cross-section data that is most appropriate.

6a. IZA ISOTOPES ZA NUMBER...Enter a value for each isotope in the standard composition component, item 1. Do not enter a value if the volume fraction, VF, is zero (item 3).

The ZA number of the isotope is entered if the user wishes to specify the isotopic distribution. This is done by entering 6a and 6b for each isotope until all the desired isotopes have been described. The "ZA" ID number is (A+1000*Z), where A is the atomic mass or weight of the isotope, and Z is the atomic number.

Items 6a and 6b can be skipped if the default values listed in the Isotope Distribution Table, Table M8.4.1, are acceptable.

6b. WTP WEIGHT PERCENT OF THE ISOTOPE...Do not enter a value if the volume fraction, VF, item 3 is zero. If item 6a is entered, a value must also be entered for 6b. The weight percent of the isotope is the percent of this isotope in the element.

The user need not enter items 6a and 6b for each isotope listed for the element in Table M8.4.1. The only requirement is that the sum of items 6b for the element must sum to 100.0. Thus, uranium could be specified as 92235 3.2 92238 96.8 to represent 3.2% enriched uranium and ignore the 234 and 236 isotopes. However, 92235 3.0 and 92238 96.8 would not be correct because the 3.0 and 96.8 do not sum to 100.0. 92235 3.0 92234 0.2 92238 96.8 would be correct.

REPEAT THE SEQUENCE 6a and 6b until all the isotopes of the element have been defined as desired.

7. END The word END is entered to terminate the input data for a standard composition component.

END COMP The words END COMP are entered only once for a problem. Their purpose is to signal the end of all the standard compositions specification data that will be used in the problem. At least two blanks should separate item 7 from the END COMP. If item 7 is followed by a name, a single blank between that name and the END COMP is sufficient.

C4.4.5 UNIT CELL SPECIFICATION FOR LATTICECELL PROBLEMS

This section describes the additional geometry data that must be entered for a LATTICECELL problem. A lattice cell calculation involves one of the following arrangements, specified from the center to the outside: (1) fuel/moderator, (2) fuel/clad/moderator, or (3) fuel/gap/clad/moderator. The geometric arrangements are limited to infinitely long cylindrical rods in a square or triangular lattice, spheres in a cubic or close-packed lattice, a symmetric array of slabs, or an asymmetric array of slabs.

These data are utilized by the code to define a unit cell that is used to determine the geometric and resonance self-shielding corrections that will be applied to the cross sections. Cell-weighted cross sections are created only if CSAS1X, CSASIX, CSAS2X, or CSAS4X are used. These cell-weighted cross sections have a flux disadvantage factor applied to them. Because of this flux disadvantage factor, mixtures used in the unit cell description should not be used later in the KENO description.

The geometry data required to specify a LATTICECELL problem are contained in Table C4.4.10. The individual entries are explained in the text following the table.

Table C4.4.10 Unit cell specification for latticecell problems

Entry Number	Variable Name	Type of data	Entry Requirement	Data Entry	Comments
1	CTP	Type of lattice	Always		Describes the type of lattice or array configuration.
				SQUAREPITCH	Use for cylindrical rods in a square pitch.
				TRIANGPITCH	Use for cylindrical rods in a triangular pitch.
				SPHSQUAREP	Use for spherical pellets in a cubic lattice.
				SPHTRIANGP	Use for spherical pellets in a bi-centered or face-centered hexagonal close-packed lattice.
				SYMMSLABCELL	Use for a symmetric array of slabs.
				ASYMSLABCELL	Use for a periodic, but asymmetric, array of slabs.
2	PITCH	Array pitch	Always	Appropriate dimension	The center-to-center spacing (cm) between fuel lumps. For asymmetric slab cell, enter the distance from the center of one moderator to the center of the other moderator (cm).
3	FUELOD	Outside dimension of fuel	Always	Appropriate dimension	Outside diameter of fuel (cm), or the thickness of the fuel in a slab.
4	MFUEL	Fuel mixture number	Always	Mixture no.	Mixture number representing the fuel.
5	MMOD	Moderator mixture number	Always	Mixture no.	Mixture number representing the moderator.
6	MMOD2	2nd moderator mixture number	ASYMSLABCELL	Mixture no.	Mixture number representing the second moderator
7	TKMOD2	2nd moderator thickness	ASYMSLABCELL	Thickness	Thickness of the second moderator (cm).
8	CLADOD	Outside diameter of clad	If clad	Clad OD	OMIT IF NO CLAD. For a slab, CLADOD is the sum of thickness of the fuel, gap, and clad.
9	MCLAD	Clad mixture number	If clad	Mixture no.	OMIT IF NO CLAD. Mixture number representing the clad.
10	CLADID	Inside diameter of clad	If gap	Clad ID	OMIT IF NO GAP between the fuel and clad.
11	MGAP	Gap mixture number	If gap	Mixture no.	OMIT IF NO GAP between the fuel and clad. A mixture number of zero is often used.
12	END	Terminate latticecell data	Always	END	Terminate latticecell input data by entering the word END.

1. CTP TYPE OF LATTICE...This defines the type of lattice or array configuration. Any one of the following alphanumeric descriptions may be used. Note that the alphanumeric description must be separated from subsequent data entries by one or more blanks.

SQUAREPITCH is used for an array of cylinders arranged in a square lattice.

TRIANGPITCH is used for an array of cylinders arranged in a triangular lattice.

SPHSQUAREP is used for an array of spheres arranged in a cubic lattice.

SPHTRIANGP is used for spheres arranged in a close-packed arrangement (dodecahedron lattice).

SYMMSLABCELL is used for a symmetric array of slabs as illustrated in Fig. C4.4.2.

ASYMSLABCELL is used for a periodic, but asymmetric array of slabs as illustrated in Fig. C4.4.3. For such an asymmetric lattice cell, the possible arrangements are fuel/moderator, fuel/clad/moderator, or fuel/gap/clad/moderator.

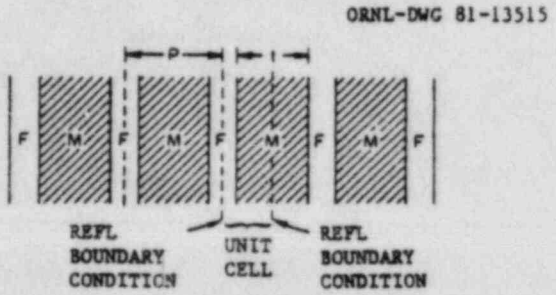


Fig. C4.4.2 Arrangement of materials in a symmetric slab cell

In the above figure, the arrangement is fuel/moderator, fuel/moderator, etc. F denotes the fuel, M denotes the moderator, P is the pitch, and t is the thickness of the moderator.

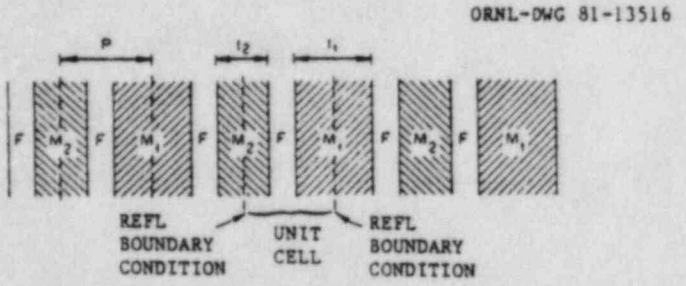


Fig. C4.4.3. Arrangement of materials in an asymmetric slab cell

In this figure, F denotes the fuel, M2 is the second moderator, M1 is the first moderator, P is the pitch, t1 is the thickness of the first moderator, and t2 is the thickness of the second moderator. The arrangement in this figure is F/M2/F/M1.

2. PITCH — ARRAY PITCH...This is the center-to-center spacing between the fuel lumps (rods, pellets, or slabs) in cm. For an ASYMSLABCELL, enter the distance from the center of one moderator to the center of the other moderator. For Figs. C4.4.2 and C4.4.3, enter the value of P.

3. FUELOD — OUTSIDE DIMENSION OF FUEL...This is the outside diameter of the fuel in cm. In slab geometry, enter the thickness of the fuel (for Fig. C4.4.2, enter the difference between the pitch and the moderator thickness, (P-t); for Fig. C4.4.3, enter the difference between the pitch and half the sum of the thickness of the first and second moderators, (P-(t1+t2)/2).

4. MFUEL — FUEL MIXTURE NUMBER...This is the mixture number representing the fuel.

5. MMOD — MODERATOR MIXTURE NUMBER...This is the mixture number representing the moderator. (M for Fig. C4.4.2, or M1 for Fig. C4.4.3) Note: A void cannot be used for MMOD at the present level of development.

6. MMOD2 — SECOND MODERATOR MIXTURE NUMBER...This is the mixture number representing the second moderator. OMIT unless entry 1, CTP, is ASYMSLABCELL.

7. TKMOD2 — SECOND MODERATOR THICKNESS...Enter ONLY for ASYMSLABCELL. This is the thickness of the second moderator in cm.

8. CLADOD — OUTSIDE DIAMETER OF CLAD...Enter ONLY if a clad is present. Enter the thickness of the clad in cm. For a slab, CLADOD is the sum of the fuel thickness, twice the gap thickness, and twice the clad thickness.

9. MCLAD — CLAD MIXTURE NUMBER...Enter ONLY if a clad is present. Enter the mixture number that represents the clad.

10. CLADID — INSIDE DIAMETER OF CLAD...Enter ONLY if a gap is present between the fuel and the clad. Omit if there is no clad. If a gap is present, enter the inside diameter of the clad in cm. This corresponds to the outside diameter of the gap. In slab geometry, CLADID is the sum of the fuel thickness and twice the gap thickness.

11. MGAP — GAP MIXTURE NUMBER...Enter ONLY if a gap is present between the fuel and the clad. Omit if there is no clad. Enter the mixture number representing the gap. Zero is often used.

12. END — The word END is entered to terminate the LATTICECELL data.

C4.4.6 GEOMETRY SPECIFICATION FOR MULTIREGION PROBLEMS

These data are entered only if the problem is defined as a multiregion problem. It describes the additional geometry data that are required for a MULTIREGION problem. A multiregion problem can be used to define a geometric arrangement that is more complicated than is allowed by a lattice cell. It can also be used for large geometric regions where the geometry effects for the cross sections are minimal. BE FULLY AWARE THAT CROSS SECTIONS GENERATED USING MULTIREGION IGNORE THE LATTICE GEOMETRY EFFECTS IN DOING THE RESONANCE CORRECTIONS.

The additional data required for a MULTIREGION problem are given in Table C4.4.11 and explained in the following text.

Table C4.4.11 Geometry specification for multiregion problems

Entry Number	Variable Name	Type of data	Entry Requirement	Data Entry	Comments
1	CS	Type of geometry	Always		Available geometry types are listed below.
				SLAB	Use for slab geometry.
				CYLINDRICAL	Use for cylindrical geometry.
				SPHERICAL	Use for spherical geometry.
				BUCKLEDSLAB	Use for slab geometry with a buckling correction
				BUCKLEDCYL	Use for cylindrical geometry with a buckling correction
2	BR	Right boundary condition	Required for BUCKLEDSLAB		Default is VACUUM. Describes the right/outside boundary condition.
			BUCKLEDCYL	VACUUM	This provides a non-return condition at the boundary.
			Optional	REFLECTED	Do not use for cylindrical or spherical.
			for other	PERIODIC	Do not use for cylindrical or spherical.
			geometries	WHITE	This provides isotropic return at the boundary.
3	BL	Left boundary condition	Required for BUCKLEDSLAB		Default is REFLECTED. Describes the left boundary condition.
			BUCKLEDCYL	VACUUM	This provides a nonreturn condition at the boundary.
			Optional	REFLECTED	Recommended for cylindrical or spherical.
			for other	PERIODIC	Do not use for cylindrical or spherical.
			geometries	WHITE	This provides isotropic return at the boundary.
4	ORGN	Location of left boundary on the x-axis		Appropriate dimension	Default is 0.0. Must not be changed for cylindrical or spherical geometry.
5	DY	Buckling height (cm)	BUCKLEDSLAB & BUCKLEDCYL	Appropriate dimension	OMIT FOR SLAB, CYLINDRICAL, and SPHERICAL. This corresponds to one of the transverse dimensions of an actual 3-D slab assembly or to the length of a finite cylinder.
6	DZ	Buckling depth	BUCKLDSLAB	Appropriate dimension	OMIT UNLESS BUCKLEDSLAB WAS SPECIFIED. This is the buckling depth corresponding to the second transverse dimension of a 3-D slab assembly.
7	END	End geometry parameters	Always	END	Enter the word END.
8	MXZ	Mixture number in the zone	Always	Mixture no.	Repeat items 8-10 until all zones are defined. Enter the mixture number for this zone.
9	RZ	Outside radius of the zone	Always	Appropriate dimension	Repeat items 8-10 until all zones are defined. Enter the outside dimension of the zone (cm).
10	XMOD	External moderator index	Always		Repeat items 8-10 until all zones are defined.
				NOEXTERMOD	No moderating materials in the adjacent zones.
				ONEEXTERMOD	A moderating material is present in one adjacent zone.
				TWOEXTERMOD	Moderating materials are present in two adjacent zones.
	END ZONE	Terminate zone area		END ZONE	Enter when all zones have been defined by repeating items 8-10 for each zone.

1. CS — TYPE OF GEOMETRY...The type of geometry must always be specified for a MULTIREGION problem. The cross-section corrections do not account for lattice effects. The available options are listed below.

   SLAB...This is used to describe a slab geometry.

   CYLINDRICAL...This is used to describe cylindrical geometry.

   SPHERICAL...This is used to describe spherical geometry.

   BUCKLEDSLAB...This is used for slab geometry with a buckling correction for the two transverse directions.

   BUCKLEDCYL...This is used for cylindrical geometry with a buckling correction in the axial direction.

2. BR — RIGHT BOUNDARY CONDITION...This is defaulted to vacuum. A value for BR MUST be entered for BUCKLEDSLAB and BUCKLEDCYL, but is optional for SLAB, CYLINDRICAL, or SPHERICAL. The available options and their qualifications are listed below:

   VACUUM...This imposes a vacuum at the boundary of the system.

   REFLECTED...This imposes mirror image reflection at the boundary. Do not use for CYLINDRICAL or SPHERICAL.

   PERIODIC...This imposes periodic reflection at the boundary. Do not use for CYLINDRICAL or SPHERICAL.

   WHITE...This causes isotropic return at the boundary.

3. BL — LEFT BOUNDARY CONDITION...This is defaulted to reflected. A value for BL MUST be entered for BUCKLEDSLAB, but is optional for SLAB, CYLINDRICAL, or SPHERICAL. The available options and their qualifications are listed below:

   VACUUM...This imposes a vacuum at the boundary of the system.

   REFLECTED...This imposes mirror image reflection at the boundary. For CYLINDRICAL or SPHERICAL, this is the only valid boundary condition because the left boundary corresponds to the centerline.

   PERIODIC...This imposes periodic reflection at the boundary.

   WHITE...This causes isotropic return at the boundary.

4. ORGN — LOCATION OF LEFT BOUNDARY ON THE ORIGIN...The default value of ORGN is 0.0. This is the only value allowed for CYLINDRICAL or SPHERICAL geometry. Enter the location of the left boundary on the x-axis (in cm).

5. DY — BUCKLING HEIGHT...Enter ONLY for BUCKLEDSLAB or BUCKLEDCYL. This is the buckling height in cm. It corresponds to one of the transverse dimensions of an actual 3-D assembly or the length of a finite cylinder.

6. DZ — BUCKLING DEPTH...Enter ONLY for BUCKLEDSLAB. This is the buckling depth in cm. It corresponds to the second transverse dimension of an actual 3-D assembly.

7. END The word END is entered to terminate these data before entering the zone description data. At least two blanks are required between items 1 and 7 if they are the only entries.

   The zone description data are entered at this point. Items 8, 9, and 10 are entered for each zone, and the sequence is repeated until all the desired zones have been described. To terminate the data, enter the words END ZONE.

8. MXZ MIXTURE NUMBER IN THE ZONE...Enter the mixture number of the material that is present in the zone. Enter a zero for a void. Repeat the sequence of items 8 through 10 for each zone.

9 RZ OUTSIDE RADIUS OF THE ZONE...Enter the outside dimension of the zone in cm. In slab geometry, RZ is the location of the zone's right boundary on the x-axis. Repeat the sequence of items 8 through 10 for each zone.

10. XMOD 10. XMOD EXTERNAL MODERATOR INDEX...This describes the conditions existing in the zones adjacent to this one. The available options are listed below.

    NOEXTERMOD...This means there are no moderating materials in an adjacent zone.

    ONEEXTERMOD...This means one of the adjacent zones contains a moderating material.

    TWOEXTERMOD...This indicates that both adjacent zones contain a moderator.

    Repeat the sequence of items 8 through 10 for each zone. When all the zones have been described, enter the words END ZONE.

END ZONE is used to terminate the multiregion zone data. Enter the words END ZONE when all the zones have been described.

C4.4.7 OPTIONAL PARAMETER DATA

Most of the parameter data for Criticality Safety Analysis Sequence No. 4 are determined by the control module and cannot be adjusted. However, certain parameters can be altered. If the default parameters are acceptable, this section of input data should be omitted in its entirety. Non-default values for one or more of the parameters can be used by entering the words MORE DATA followed by the desired keyworded parameters and their associated values. One or more of the parameters can be entered in any order. Default values are used for parameters that are not entered. Each parameter is entered by spelling its name, followed immediately by an equal sign and the value to be entered. There should not be a blank between the parameter name and the equal sign. Each parameter specification must be separated from the rest by at least one blank. For example:

MORE DATA ISN=16 EPS=0.00008 END

would result in using an S16 angular quadrature set and tightening the convergence criteria to 0.00008. Note that the END is not the end for the analytical sequence, item 5 of Table C4.4.7.

Table C4.4.12 describes the optional parameters that can be entered as data.

Table C4.4.12 Optional parameter data

Entry Number	Keyword Name	Type of Data	Comments
1	MORE DATA		This signals that optional parameter data are to be entered.
2	ISN=	Order of angular quadrature	The default value is 8. This allows using another value.
3	SZF=	Spatial mesh size factor	The default value is 1.0 SZF<1.0 gives a finer mesh SZF>1.0 gives a coarser mesh
4	IIM=	Max. number of inner iterations	The default value is 20. This allows using another value.
5	ICM=	Max. number of outer iterations	The default value is 25. This allows using another value.
6	EPS=	Overall convergence criteria	The default value is 0.0001. This allows using another value.
7	PTC=	Point convergence criteria	The default value is 0.0001. This allows using another value.
8	BKL=	Buckling factor	The default value is 1.420892. Use ONLY for a multiregion problem that specifies BUCKLEDSLAB or BUCKLEDCYL.
9	IUS=	Upscatter scaling factor	The default value is zero. IUS=0 doesn't utilize upscatter scaling. IUS=1 uses upscatter scaling to accelerate the solution and/or speed convergence.
10	RES=	Resonance data	Enter the mixture number, geometry type (SLAB, CYLINDER, SPHERE) and the thickness of the slab or radius of the sphere or cylinder, in cm.
11	DAN(mm)=	Dancoff factor for the specified mixture	Enter the mixture number mm to which the Dancoff factor applies inside the parentheses; enter the Dancoff factor after the equal sign Repeat items 10 and 11 for all resonance mixtures used in the problem that are not treated in the lattice cell or multiregion description.
12	END	Terminus	Terminate the optional parameter data

SEARCH DATA
READ SEARCH and is terminated by entering END SEARCH.

1. MORE DATA...These words, followed by one or more blanks, are entered ONLY if optional parameter data are to be entered. They must precede the first entry of the optional parameter data.

2. ISN ORDER OF ANGULAR QUADRATURE...Quadrature sets are geometry-dependent quantities that are defaulted to order 8 for the Criticality Safety Analysis Sequences. See Sect. C1.2.6 for a detailed explanation.

3. SZF SPATIAL MESH SIZE FACTOR...The automatic mesh generator used in this sequence is discussed in Sect. C1.2.5. The size of the largest mesh interval can be adjusted by entering a value for SZF. The default value is 1.0. A value less than 1.0 yields a finer mesh; a value greater than 1.0 yields a coarser mesh.

4. IIM MAXIMUM NUMBER OF INNER ITERATIONS...This is the maximum number of inner iterations to be used in the XSDRNPM calculation. The default value is 20.

5. ICM MAXIMUM NUMBER OF OUTER ITERATIONS...This is the maximum number of outer iterations to be used in the XSDRNPM calculation. The default value is 25. If the calculation reaches the outer iteration limit, a larger value should be used.

6. EPS OVERALL CONVERGENCE CRITERIA...This is used by XSDRNPM after each outer iteration to determine if the problem has converged. The default value of EPS is 0.0001. A value less than 0.0001 tightens the convergence criteria; a larger value loosens the convergence criteria. See Sect. C1.2.8 for additional information.

7. PTC POINTWISE CONVERGENCE CRITERIA...This is the point flux convergence criteria used by XSDRNPM to determine if convergence has been achieved after an inner iteration. The default value is 0.0001. A smaller value tightens convergence; a larger value loosens it.

8. BKL BUCKLING FACTOR...A buckling factor should be used ONLY for a multiregion BUCKLEDSLAB or BUCKLEDCYL problem. Because cylinders are assumed to be infinitely long and slabs are assumed to be infinite in both transverse directions, the analytic sequence may tend to overestimate the total flux for a finite system. A buckling correction can be used to approximate the leakage from the system in the transverse direction(s). The extrapolation distance factor, BKL, is defaulted to 1.420892. See Sect. C1.2.7 for additional details.

9. IUS UPSCATTER SCALING FLAG...This option allows the use of upscatter scaling to accelerate the solution or force convergence. The default value is zero, in which case upscatter scaling is not used. IUS=1 facilitates upscatter scaling. Guidelines are not available to indicate when upscatter scaling is needed. Some problems will not converge with it and some will not converge without it.

10. RES RESONANCE DATA...This allows overriding the default resonance data for materials that are not used in the unit cell. The default resonance data are infinite homogeneous media which may not accurately represent the geometry (for example, a thin clad or container region). The resonance data are entered in the form:

    RES= mixture number geometry type thickness.

    The mixture number is the material for which the resonance data apply, the geometry type defines the third entry of the 3* array in NITAWL, and the thickness is the fourth entry of the 3* array in NITAWL. The thickness is the thickness of a slab or the radius of a sphere or cylinder in cm. This entire data sequence should be entered for each material for which the default data are inadequate.

11. DAN(mm)= DANCOFF FACTOR for the specified mixtures. This allows entering a Dancoff factor to be used in the resonance correction for the specified mixture. The Dancoff data are entered in the form:

    DAN(mm) = Dancoff factor

Note that the parentheses must be entered as part of the data. The mixture number, mm, to which the Dancoff factor applies, must be enclosed in the parentheses. The Dancoff factor to be used is entered after the equal sign.

Items 10 and 11 should be entered for each resonance mixture that is not specified in the lattice cell or multiregion data. See Sect. C4.5.1 for additional information.

12. END The word END is entered to terminate the optional parameter data.

C4.4.8 SUMMARY OF KENO-V DATA

KENO-V data are required for CSAS25, CSAS2X, CSAS4, and CSAS4X. A brief outline of the KENO-V data is given in Table C4.4.13. The input data are summarized in Table C4.4.14. The input data for KENO V are discussed in detail in Sects. F11.4 and F11.5. Data need not be supplied for entries whose default value is acceptable. KENO-V mixing table data cannot be entered or used in CSAS because they are automatically supplied by the control module.

Table C4.4.13 Outline of KENO-V data

Type of Data	Starting Flag	Comments	Termination Flag
Title	None	80 columns, must be entered first	none
Parameters	READ PARM	Enter desired parameter data	END PARM
Biasing or weighting	READ BIAS	Enter desired biasing data	END BIAS
Geometry	READ GEOM	Enter desired geometry data	END GEOM
Array data or unit orientation	READ ARRAY	Enter desired array data	END ARRAY
Boundary conditions or albedoes	READ BNDS	Enter desired albedo data	END BNDS
Start data or initial source	READ START	Enter desired start data	END START
Plot data	READ PLOT	Enter desired plot data	END PLOT

Table C4.4.14 Summary of KENO V data

Summary of parameter data

TITLE The title must be entered first (80 columns) See Section F11.4.3

PARAMETERS Format: READ PARAM enter parameter data here END PARAM
If parameters are entered, they must follow the title. See Sections F11.4.3, F11.5.2 and F11.5.3.

KEY	STD.	DEFINITION
RND=	given	random number
THE=	30.min	execution time(min)
TBA=	0.5min	batch time (min)
WTA=	0.5	average weight
WTK=	3.0	wt. for splitting
WTL=	1/WTH	Russian Roulette wt.
GEN=	103	No. of generations
NPG=	300	No. per generation
NSK=	3	generations skipped
RES=	0	gens. between restart
NBK=	GEN+25	neutron bank positions
XNB=	0	extra bank entries
NFB=	NPG	fission bank positions
XFB=	0	extra bank entries
X1D=	0	No. of extra 1-D's
LNG=	4000000	words of storage
BEG=	1	restart at this gen.
NB8=	200	blocks for d.a. unit
NL8=	789	length of d.a. block

KEY	STD.	DEFINITION
RUN=	YES	execute problem
FLX=	NO	fluxes
FDN=	NO	fission densities
ADJ=	NO	adjoint calculation
AMX=	NO	all mixture xsecs
XAP=	NO	xsec angles & probs.
XS1=	NO	1-D xsecs
XS2=	NO	2-D xsecs
FKI=	NO	fission spectrum
F1D=	NO	extra 1-D xsecs
FAR=	NO	fiss & abs.
MKP=	NO	matrix by location
CKP=	NO	cofactor k by loc.
FMP=	NO	fiss. prod. by loc.
MKU=	NO	matrix by unit
CKU=	NO	cofactor k by unit
FMU=	NO	fiss. prod. by unit

KEY	STD.	DEFINITION
MKH=	NO	matrix by hole
CKH=	NO	cofactor k by hole
FMH=	NO	fiss. prod. by hole
HHL=	NO	MKH at highest level
MKA=	NO	matrix by array
CKA=	NO	cofactor k by array
FMA=	NO	fiss. prob. by array
HAL=	NO	MKA at higest level
PLT=	YES	printer plots tracks
BUG=	NO	debug print
TRK=	NO	print neutron tracks
PWT=	NO	print avg. weight
PGM=	NO	unprocessed geometry
SMU=	NO	self-multiplication
NUB=	NO	neutrons per fission
PAX=	NO	albedo-xsec array

KEY	STD.	DEFINITION
XSC=	19	mixed xsecs
ALB=	79	albedo
WTS=	80	weights
LIB=	0	working xsecs
SKT=	16	scratch
RST=	0	read restart
WRS=	0	write restart

Table C4.4.14 (continued)

Summary of array data

ARRAY

Format: READ ARRAY array parameters data type orientation data END ARRAY
See Sections F11.5.5, F11.5.6 and F11.5.7

Repeat the sequence ARRAY PARAMETERS DATA TYPE ORIENTATION DATA for each array used in the problem.

ARRAY PARAMETERS

KEYWORD	DEFAULT	DEFINITION
ARA=	1	No. defining the array
NUX=	1	No. of units in X direction
NUY=	1	No. of units in Y direction
NUZ=	1	No. of units in Z direction
GBL=	maxara	The global, universe or overall array number**
COM=	none	delim comment delim optional comment is a maximum of 132 characters.

** Can be defaulted by the code. If specified, it need be entered only once per problem.

DATA TYPE

FILL
LOOP

ORIENTATION DATA FOR FILL

Enter unit numbers to define every position in the array. When entering data utilizing the options in this table, the count field and option field must be adjacent with no imbedded blanks. The operand field may be separated from the option field by one or more blanks. Fill data for an array ends with END FILL.

COUNT FIELD	OPTION FIELD	OPERAND FIELD	COMMENTS
		j	stores j at the current position in the array
i	R	j	stores j in the next i positions in the array
i	*	j	stores j in the next i positions in the array
i	$	j	stores j in the next i positions in the array
	F	j	fills remainder of the array with unit No. j starting with the current array position
	A	j	sets the current position in the array to j
i	S		increments current position in the array by i allows skipping i positions. The value of i may be positive or negative.
i	Q	j	repeats the previous j entries i times. The default value of i is 1.
i	M	j	repeats the previous j entries i times, inverting the sequence each time. The default value of i is 1.
i	B	j	Starting with the entry at -i from the current position, store entries in inverse order until position -(i+j) is reached. Default value of i=1.
i	P	j	alternately stores j and -j in the next i positions of the array.
i	I	j k	provides i entries linearly interpolated between j and k, and the end points (i.e. a total of i+2 points.) j and k must be separated by at least one blank.
	T		terminates the data reading for the array.

ORIENTATION DATA FOR LOOP

Enter the unit number and nine numbers that define the position(s) of that unit. Data for each of these ten entries are repeated until every position in the array has been defined. Data for an array is terminated by entering END LOOP.

ENTER DATA IN THE FORM:

DATA ENTRY	COMMENTS
LTYPE	The unit or box type. LTYPE must be greater than 1.
IX1	Starting position in the X direction. IX1 must be at least 1 and no larger then the value entered for NUX.
IX2	Ending position in the X direction. IX2 must be at least 1 and no larger than the value of NUX.
INCX	The number of units by which increments are made in the X direction.
IY1	The starting position in the Y direction. IY1 must be at least 1 and less than the value entered for NUY.
IY2	Ending position in the Y direction. IY2 must be at least 1 and no larger than the value of NUY.
INCY	The number of units by which increments are made in the positive Y direction.
IZ1	Starting position in the Z direction. IZ1 must be at least 1 and no larger than NUZ. in the positive Y direction.
IZ2	Ending position in the Z direction. IZ2 must be at least 1 and no larger than NUZ.
INCZ	The number of units by which increments are made in the positive Z direction.

Table C4.4.14 (continued)

Summary of biasing data

BIAS . Format: READ BIAS keyword correlation data auxiliary END BIAS
(weighting) See Sections F11.4.7 and F11.5.8

KEYWORD	DESCRIPTION
ID=	CORRELATION DATA will be read next.
	id material ID. Enter from table at right. to use weighting data from the library.
	ibgn beginning bias ID
	iend ending bias ID
WT=	AUXILIARY DATA will be read next.
WTS=	AUXILIARY DATA will be read next.
	wttitl material title
	id material ID.
	nsets number of sets of group structures
	REPEAT THKINC, NUMINC, NGPWT, WTAVG NSETS TIMES
	thkinc thickness per increment
	numinc number of increments
	ngpwt number of energy groups for this set of wts
	wtavg enter numinc* ngpwt values of wtavg.

MATERIAL	ID	ENERGY GROUPS	THICKNESS/INCREMENT
concrete	301	16, 27, 123	5 cm
paraffin	400	16, 27, 123	3 cm
water	500	16, 27, 123, 218	3 cm
graphite	6100	16, 27, 123	20 cm

For CORRELATION DATA, the Material ID is chosen from material ID column above (the keyword is ID=).
For AUXILIARY DATA, the material ID is chosen by the user and the keyword is WT= or WTS=.

Beginning and ending bias ID's are defined by the user. The geometry specification that has the bias ID equal to the beginning bias ID utilizes the wt avg's from the first interval of material ID.

Table C4.4.14 (continued)

Summary of boundary condition data

BNDS (albedo or boundary conditions)

Format: READ BNDS face code albedo name END BNDS
See Section F11.4.7

The sequence FACE CODE ALBEDO NAME is entered as many times as necessary to define the appropriate albedo boundary conditions. The default for all faces is vacuum.

FACE CODES FOR ENTERING BOUNDARY (ALBEDO) CONDITIONS

FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION
+XB=	positive X face	XFC=	both X faces	+YX=	positive X and Y faces	&EY=	positive Y and E faces
&XB=	positive X face	YFC=	both Y faces	&XY=	positive X and Y faces	-XY=	negative X and Y faces
-XB=	negative X face	EFC=	both E faces	&YX=	positive X and Y faces	-XE=	negative X and E faces
+YB=	positive Y face	+FC=	all positive faces	+XE=	positive X and E faces	-YE=	negative Y and E faces
&YB=	positive Y face	&FC=	all positive faces	+EX=	positive X and E faces	YXF=	all X and Y faces
-YB=	negative Y face	-FC=	all negative faces	&XE=	positive X and E faces	EXF=	all X and E faces
+EB=	positive E face	XYF=	all X and Y faces	&EX=	positive X and E faces	EYF=	all Y and E faces
&EB=	positive E face	XEF=	all X and E faces	+YE=	positive Y and E faces	-YX=	negative X and Y faces
-EB=	negative E face	YEF=	all Y and E faces	+EY=	positive Y and E faces	-EX=	negative X and E faces
ALL=	all 6 faces	+XY=	positive X and Y faces	&YE=	positive Y and E faces	-EY=	negative Y and E faces

ALBEDO NAMES AVAILABLE ON THE KENO V ALBEDO LIBRARY, FOR USE WITH THE FACE CODES

ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION
DP0H2O DPOH2O DP0 DPO	12 inch double P0 water differential albedo with 4 incident angles	CONC-4 CON4 CONC4	4 inch concrete differential albedo with 4 incident angles	VACUUM VOID VACU VAC	vacuum condition
H2O WATER	12 inch water differential albedo with 4 incident angles	CONC-8 CON8 CONC8	8 inch concrete differential albedo with 4 incident angles	SPECULAR MIRROR MIRR SPEC	mirror image reflection
PARAFFIN PARA WAX	12 inch paraffin differential albedo with 4 incident angles	CONC-12 CON12 CONC12	12 inch concrete differential albedo with 4 incident angles	SPE MIR	
CARBON GRAPHITE C	200 cm. carbon differential albedo with 4 incident angles	CONC-16 CON16 CONC16	16 inch concrete differential albedo with 4 incident angles	PERIODIC PERI PER	periodic boundary condition
ETHYLENE POLY CH2	12 inch polyethylene differential albedo with 4 incident angles	CONC-24 CON24 CONC24	24 inch concrete differential albedo with 4 incident angles		

Table C4.4.14 (continued)

Summary of geometry data

GEOMETRY (region)

Format: READ GEOM enter geometry region data here END GEOM
See Sections F11.4.4, F11.5.1.2, F11.5.6 and F11.5.7

GEOMETRY REGION DATA consists of SIMPLE GEOMETRY REGION DATA and EXTENDED GEOMETRY REGION DATA.

ENTER GEOMETRY REGION DATA IN THE FOLLOWING FORM:

OPTIONAL GLOBAL SPECIFICATION
UNIT n
OPTIONAL GEOMETRY COMMENT
SIMPLE GEOMETRY REGION DATA
and/or
EXTENDED GEOMETRY REGION DATA

* *

ENTER SIMPLE GEOMETRY REGION DATA IN THE FOLLOWING FORM:

GLOBAL Enter only to specify this unit as the global unit.
UNIT n
COM=delim comment delim This optional comment can be up to 132 characters. It must begin and end with a delimiter.
fgeom mix no. bias ID dimensions optional origin data (ORIGN coordinates) optional chord data (CHORD)

Enter as many geometry description specifications as necessary to describe the unit and as many units as necessary to describe the system.

SIMPLE GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA	TYPE 4 DATA	TYPE 5 DATA	TYPE 6 DATA
fgeom	SPHERE HEMISPHERE HEMISPHERE+X HEMISPHERE-X HEMISPHERE+Y HEMISPHERE-Y HEMISPHERE+Z HEMISPHERE-Z	XCYLINDER XHEMICYL+Y XHEMICYL-Y XHEMICYL+Z XHEMICYL-Z	YCYLINDER YHEMICYL+X YHEMICYL-X YHEMICYL+Z YHEMICYL-Z	CYLINDER ZCYLINDER ZHEMICYL+X ZHEMICYL-X ZHEMICYL+Y ZHEMICYL-Y	CUBE	CUBOID
dimensions	R (radius)	R +H -H	R +H -H	R +H -H	+X -X	+X -X +Y -Y +Z -Z
optional origin coord.	Enter the X Y Z coord. of origin	Enter the Y Z coord. of center-line	Enter the X Z coord. of center-line	Enter the X Y coord. of center-line	omit	omit
optional chord data**	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	omit	omit

**Chord data is not applicable for SPHERE XCYLINDER YCYLINDER CYLINDER or ZCYLINDER

Table C4.4.14 (continued)

Summary of geometry data (continued)

GEOMETRY (region) (cont.)

ENTER EXTENDED GEOMETRY DATA IN THE FOLLOWING FORM:

fgeom ref. ID bias ID thickness per region origin coordinates nreg

EXTENDED GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA
fgeom	ARRAY CORE COREBDY COREBNDS COREBOUN	HOLE	REPLICATE REFLECTOR
ref. ID	array no.	emplaced unit number	mixture no. in generated regions
bias ID	omit for ARRAY	omit	first bias ID
thick./reg.	omit	omit	variable**
origin coord.	Enter the X Y Z coord. of most neg. pt. of array	Enter the X Y Z coord. of origin	omit
nreg	omit	omit	No. of regions to be generated

**The number of dimensions to be entered is the same as the region preceding the replicate or reflector specification because the generated regions have that shape. The value of the dimensions is the thickness of each generated region of material on that surface.

Table C4.4.14 (continued)

Summary of plot data

PLOT Format: READ PLOT plot parameters END PLOT
plot parameters must be entered for each plot that is to be made.
See Sections P11.4.11 and P11.5.9

KEYWORD	DEFAULT	DEFINITION
TTL=	prob. title	delim ptitl delim A one character delimiter signals the beginning and end of the title. ptitl is the plot title.
PIC=	MAT	Type of picture: MIXTURE, UNIT NO. or BIAS ID NO.
		MIXTURE ------- MAT
		MIX
		MIXT
		MIXTURE
		MEDI
		MEDIA
		UNIT NO. ------- BOX
		BOXT
		BOXTYPE
		UNT
		UNIT
		UNITTYPE
		BIAS ID NO. ---- IMP
		BIAS
		BIASID
		WTS
		WEIG
		WEIGHTS
		WGT
		WGTS
XUL=	prev. plot	X coord. of upper left corner of plot
YUL=	prev. plot	Y coord. of upper left corner of plot
ZUL=	prev. plot	Z coord. of upper left corner of plot
XLR=	prev. plot	X coord. of lower right corner of plot
YLR=	prev. plot	Y coord. of lower right corner of plot
ZLR=	prev. plot	Z coord. of lower right corner of plot

KEYWORD	DEFAULT	DEFINITION
UAX=	prev. plot 0 IF VAX OR WAX is read	X component of direction cosine for the AX axis of the plot (across)
VAX=	prev. plot 0 IF UAX OR WAX is read	Y component of direction cosine for the AX axis of the plot (across)
WAX=	prev. plot 0 IF UAX OR VAX is read	Z component of direction cosine for the AX axis of the plot (across)
UDN=	prev. plot 0 IF VDN OR WDN is read	X component of direction cosine for the DN axis of the plot (down)
VDN=	prev. plot 0 IF UDN OR WDN is read	Y component of direction cosine for the DN axis of the plot (down)
WDN=	prev. plot 0 IF UDN OR VDN is read	Z component of direction cosine for the DN axis of the plot (down)
DLX=		horizontal spacing between points on plot
DLD=		vertical spacing between points on plot
NAX=		no. of intervals to be printed across page
NDN=		no. of intervals to be printed down page
LPI=	8	lines per inch printed down the page
NCH=	CHRS**	delim CHRS delim a one character delimiter signals the beginning and end of the character string.
RUN=	YES	YES allows the problem to execute NO terminates problem after data checking
PLT=	YES	YES allows the plot(s) to be made NO allows reading the plot data without making a plot.

PLOT ORIGIN:

(1) SINGLE UNIT-coincides with origin of geometry description.

(2) BARE ARRAY-at the most negative point of the array. (lower left-hand back corner of the global array).

(3) REFLECTED ARRAY- coincides with the origin of the CORE or ARRAY description of the global array.

**default values of CHRS are given below.

MEDIA	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CHRS		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M
MEDIA	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42			
CHRS	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	#	,	$	-	+	;	|			
MEDIA	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58							
CHRS	&	>	:	'	.	-	%	*	"	=	!	(	@	<	/	0							

Table C4.4.14 (continued)

Summary of starting data

START

Format: READ START enter start data here END START
The default value of start type is zero.
See Section F11.4.8

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
0	none	NST	uniform
		XSM	
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
1	NST	XSM	cosine
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
2	NST	XSM	cosine with
	NXS	XSP	fraction in
	NYS	YSM	specified
	NZS	YSP	unit
	FCT	ZSM	
		ZSP	
		RFL	
		PSP	

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
3	NST	KFS	spike
	TFX	PSP	
	TFY		
	TFZ		
	NXS		
	NYS		
	NZS		
4	NST	KFS	multiple
	TFX	PSP	spikes
	TFY		
	TFZ		
	NBX		
5	NST	PSP	in specified
	NBX		units
6	NST	NXS	arbitrary
	TFX	NYS	points
	TFY	NZS	
	TFZ	KFS	
	LNU*	PS6	
		PSP	

*LNU must be the last entry for each set of start 6 data. The LNU of each successive set of data must be larger than the last.

KEYWORD	DEFAULT	DEFINITION
NST=	0	start type
TFX=	0.0	X coordinate
TFY=	0.0	Y coordinate
TFZ=	0.0	Z coordinate
NXS=	0	X index of unit pos.
NYS=	0	Y index of unit pos.
NZS=	0	Z index of unit pos.
KFS=		fission spectra
LNU=	0	number of last neutron
NBX=	0	source unit number
FCT=	0	fraction
XSM=	-X	-X of source cuboid
XSP=	+X	+X of source cuboid
YSM=	-Y	-Y of source cuboid
YSP=	+Y	+Y of source cuboid
ZSM=	-Z	-Z of source cuboid
ZSP=	+Z	+Z of source cuboid
RFL=	NO	start in reflector
PS6=	NO	print start 6 input
PSP=	NO	print starting points

C4.4.9 SEARCH DATA

Search data are required only for CSAS4 and CSAS4X. They are omitted for all other sequences. Search data are entered in the form:

READ SEARCH search data END SEARCH

The words READ SEARCH inform the code that search data are to be read and the words END SEARCH terminate the search data. The search data options are described in Sects. C4.4.9.1 and C4.4.9.2. Additional information and examples are contained in Sect. C4.5.3.

C4.4.9.1 *Search Type Specification*

The basic search data to define the type of search and various search parameters that set limits for the search are described in this section. These data are listed in Table C4.4.15.

Table C4.4.15 Outline of search type specification

Data Position	Type of Data	Data Entry	Comments
1.	Search Descriptor	OPTIMUM OPT OPTM	Initiates a search for the maximum value of k-effective.
		CRITICAL CRTL	Initiates a search for a specified value of k-effective.
		MINIMUM MINM MIN	Initiates a search for the minimum value of k-effective.
2.	Search Type	PITCH PIT PTCH	Vary the pitch
		DIMENSION DIM DMSN	Vary one or more dimensions in one or more regions.
		CONCENTRATION CON	NOT IMPLEMENTED
3.	Optional search parameters		Optional search parameters allow changing default values. Any or all may be entered in any order.
3a.	No. of search passes	PAS=	Enter the keyword PAS= followed by the desired number of search passes. default=10.
3b.	No. of search parameters	NPM=	Enter the keyword NPM= followed by the number of search parameters. Present capability is limited to 1. default=1.
3c.	Search convergence tolerance	EPS=	Enter the keyword EPS= followed by the desired convergence tolerance. default=0.005
3d.	Desired value of k-effective	KEF=	Enter the keyword KEF= followed by the desired value of k-effective. default =1.000 DO NOT ENTER FOR OPTIMUM OR MINIMUM SEARCHES.
4.	Additional search data	MORE	Enter the delimiter MORE This delimiter ends the optional search parameters and initiates the individual search commands found in Table C4.4.16.

The desired search type must be specified in the search data before the individual search commands can be entered. The desired search type is specified only once in a problem. The data specifications are listed below.

1. SEARCH DESCRIPTOR

   Enter a word (OPTIMUM, CRITICAL, MINIMUM) to define the basic search mode. A default value is not supplied. The three basic search modes are:

   OPTIMUM initiates a search for the maximum value of k-effective.

   CRITICAL initiates a search for a specified value of k-effective.

   MINIMUM initiates a search for the minimum value of k-effective.

2. SEARCH TYPE

   Enter a word (PITCH, DIMENSION, CONCENTRATION) to specify the variable that is changed during the search. A default value is not supplied. The variables are:

   PITCH causes the pitch (center-to-center spacing in the x and y directions) to vary according to the search parameters.

   DIMENSION causes the dimensions in one or more geometry regions to vary according to the search parameters.

   CONCENTRATION causes the concentration to vary. THIS OPTION IS NOT IMPLEMENTED.

3. OPTIONAL SEARCH PARAMETERS

   Optional search parameters are available for changing default values. They are specified by entering a keyword followed by the desired value. The optional search parameters are listed below.

3a. PAS= NUMBER OF SEARCH PASSES

   The number of search passes is the maximum number of times the search will change data and calculate the corresponding k-effective. The default is 10.

3b. NPM= NUMBER OF SEARCH PARAMETERS

   The number of search parameters is the not fully implemented. The default value is 1 and should not be overridden.

3c. EPS= SEARCH CONVERGENCE TOLERANCE

   The search convergence tolerance is the search convergence criteria. OPTIMUM and MINIMUM searches are terminated when a k-effective is calculated that is within EPS of the optimum or minimum respectively. A CRITICAL search is terminated when a calculated k-effective is within EPS of the target or desired k-effective. The default value is 0.005.

3d. KEF= DESIRED VALUE OF K-EFFECTIVE

   This information is entered ONLY for a CRITICAL search. The default value is 1.000.

4. MORE ADDITIONAL SEARCH DATA will follow.
   The word MORE terminates the optional search parameters and initiates the individual search commands. Do not enter MORE unless individual search commands are to be entered.

C4.4.9.2 *Individual Search Commands*

Individual search commands are entered ONLY if MORE, item 4, of the search type specification data were entered (see Table C4.4.15). Individual search commands are used to specify search constraints and to communicate to the search program the units that will be altered, the regions that will be altered within those units, and the faces or surfaces of those regions that will be altered. The data comprising the individual search commands are listed in Table C4.4.16. All data except items 3a, 3b, 3c and 7 are keyworded (i.e., the data are entered by specifying a keyword, followed by a value). An explanation of each individual search command follows.

Table C4.4.16 Outline of individual search commands

Entry Number	Keyword Name	Type of Data	Entry Requirements	Comments
1	MAXPITCH=	Maximum allowed pitch	PITCH	This is used to specify the maximum allowed pitch for the search. The default value is the pitch corresponding to -5*the parameter at the minimum possible pitch.
2	MINPITCH=	Minimum allowed pitch	PITCH	This is used to specify the maximum allowed pitch for the search. The default value is the minimum possible pitch (i.e. the pitch at which the units in the array touch).
3a	ALTER ALTR ALT CHANGE CHNG MODIFY MOD MDFY	Begin a new search command	DIMENSION PITCH	These words are used to specify that modifications will be made to the geometry according to subsequent commands (items 5 through 8 as required to specify the desired changes).
3b	MAINTAIN MNTN	Begin a new search command	DIMENSION PITCH	The spacing (thickness) of the specified geometry regions will be maintained when the interior regions grow or shrink.
3c	KEEP HOLD	Begin a new search command	DIMENSION PITCH	This command resets the specified geometry to the original input specifications.
4	PAR=	Parameter number	DIMENSION PITCH	Enter the parameter number the current command (ALTER, MAINTAIN, KEEP) applies to. The present capability is limited. Default=1 and should not be overridden.
5	UNIT=	Unit the current command applies to	DIMENSION PITCH	Enter the unit in which regions are to be altered. Default=1.
6	REG=	First region to be altered in the unit	DIMENSION PITCH	Enter the first or only region in the unit which the search constants(items 8a, b, c and/or d) apply to.
7	TO	Last region to be altered in the unit	DIMENSION PITCH	Enter the last region in the unit which the search constants (items 8a, b, c and/or d) apply to. Default is the first region. Note--Item 6 must be entered in order to alter a single region in a unit. Both items 6 and 7 must be entered in order to alter more than one region in a unit.

Table C4.4.16 (continued)

8a	ALL=	Search constant for all surfaces (faces) of the region(s).	DIMENSION PITCH	Enter a value for the search constants for the specified regions. This value will be applied to all surfaces of the region(s).
8b	+X=	Search constant for +X face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the +X face of a cuboid.
	-X=	Search constant for -X face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the -X face of a cuboid.
	+Y=	Search constant for +Y face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the +Y face of a cuboid.
	-Y=	Search constant for -Y face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the -Y face of a cuboid.
	+Z=	Search constant for +Z face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the +Z face of a cuboid.
	-Z=	Search constant for -Z face of cuboid	DIMENSION PITCH	Enter a value for the search constant for the -Z face of a cuboid.
8c	+R=	Search constant for radius	DIMENSION PITCH	Enter a value for the search constant for the radius of a sphere or a cylinder.
	+H=	Search constant for + height	DIMENSION PITCH	Enter a value for the search constant for the + height of a cylinder.
	-H=	Search constant for - height	DIMENSION PITCH	Enter a value for the search constant for the - height of a cylinder.
8d	CH=	Search constant for chord	DIMENSION PITCH	Enter a value for the search constant for the chord face of a hemisphere or hemicylinder.
9	+CON=	Maximum constraint	DIMENSION PITCH	Enter the maximum constraint for the current parameter. default=+10E10
0	-CON=	Minimum constraint	DIMENSION PITCH	Enter the minimum constraint for the current parameter. default=-10E10

1. MAXPITCH= MAXIMUM ALLOWED PITCH...The maximum allowed pitch (center-to-center spacing in the x and y directions) is specified using this keyword. It applies to a PITCH search type and causes the search to terminate if the specified pitch is exceeded. The default value is the pitch corresponding to -5 times the parameter at the minimum pitch.

2. MINPITCH= MINIMUM ALLOWED PITCH...The minimum allowed pitch (center-to-center spacing in the x and y directions) is specified using this keyword. It applies to a PITCH search type and causes the search to terminate if the pitch needs to be smaller than MINPITCH. The default value is the pitch at which the pins in the array are touching.

3. A NEW SEARCH COMMAND...A new search command is initiated whenever an item 3a through 3c is encountered. The code will vary the geometry according to subsequent commands.

3a. ALTER, ALTR, ALT, CHANGE, CHNG, MODIFY, MOD, MDFY — ALTER GEOMETRY REGIONS...These words specify that modifications will be made to the geometry according to subsequent commands.

3b. MAINTAIN, MNTN — MAINTAIN THE THICKNESS...The thickness of the specified geometry region(s) will be maintained when the interior regions grow or shrink (i.e., the specified region will grow or shrink in conjunction with the interior region in such a way as to maintain the original distance between the two regions). This means that the original thickness of the region is preserved. For instance, the inner radius of a pipe can be altered and the wall thickness can be preserved by applying the MAINTAIN command to the region defining the outer radius of the pipe.

3c. KEEP, HOLD — KEEP THE ORIGINAL SPECIFICATION...This command causes the specified geometry region(s) to be reset to their original input value for every search pass. Therefore they go through the entire search process unchanged.

4. PAR= PARAMETER NUMBER...The search parameter number is not fully implemented. The default number is 1 and should not be overridden.

5. UNIT= GEOMETRY UNIT NUMBER...This is the geometry unit to which the previously entered search command (item 3a, 3b, or 3c) is applied. Items 6, 7, and 8 specify the region(s) within the unit and the surfaces of the region(s) to be altered.

6. REG= FIRST REGION TO BE ALTERED...This is used to specify the first or only region in the unit (specified by item 5) that is to be altered according to the search command (item 3a, 3b, or 3c). The region(s) are altered according to the search constants (items 8a, 8b, 8c, and/or 8d).

7. TO LAST REGION TO BE ALTERED...This item is entered to specify the last region to be altered, starting with the region specified by REG=. For example, assume unit 3 contains 8 regions and you wish to make changes to regions 4, 5, 6, 7, and 8. These regions are identified by entering the following data. UNIT=3 REG=4 TO 8

8. SEARCH CONSTANTS...A search constant is the proportionality factor utilized to alter a geometry region. A search constant must be entered for each surface of a region that is to be altered. A search constant of 1.0 will cause the region dimension for that surface to be changed. A search constant of 0.0 will cause the region dimension to remain unchanged. The default value of the search constant is 0.0.

8a. ALL= SEARCH CONSTANT FOR ALL SURFACES...All of the surfaces in a region are altered simultaneously by using this search command.

8b. +X= SEARCH CONSTANT FOR +X FACE...This parameter is used to specify the value of the search constant for the +X face of a cuboid.

-X= SEARCH CONSTANT FOR -X FACE...This parameter is used to specify the value of the search constant for the -X face of a cuboid.

+Y= SEARCH CONSTANT FOR +Y FACE...This parameter is used to specify the value of the search constant for the +Y face of a cuboid.

-Y= SEARCH CONSTANT FOR -Y FACE...This parameter is used to specify the value of the search constant for the -Y face of a cuboid.

+Z= SEARCH CONSTANT FOR +Z FACE...This parameter is used to specify the value of the search constant for the +Z face of a cuboid.

-Z= SEARCH CONSTANT FOR -Z FACE...This parameter is used to specify the value of the search constant for the -Z face of a cuboid.

NOTE: If it is desirable to change all the faces of a cuboid except the -z face by some amount (search constant of 1.0), items 8a and 8b can be used together as follows: ALL=1.0 -Z=0.0. This is the same as entering +X=1.0 -X=1.0 +y=1.0 -Y=1.0 +Z=1.0. If both z faces are to remain unchanged, items 8a and 8b can be entered as: ALL=1.0 +Z=0.0 -Z=0.0 or as +X=1.0 -X=1.0 +Y=1.0 -Y=1.0.

8c. +R= SEARCH CONSTANT FOR RADIUS...This parameter is used to specify the value of the search constant for the radius of a sphere, hemisphere, cylinder, or hemicylinder.

+H= SEARCH CONSTANT FOR +HEIGHT...This parameter is used to specify the value of the search constant for the +height of a cylinder or hemicylinder.

-H= SEARCH CONSTANT FOR -HEIGHT...This parameter is used to specify the value of the search constant for the -height of a cylinder or hemicylinder.

8d. CH= SEARCH CONSTANT FOR CHORD...This parameter is used to specify the value of the search constant for the chord of a hemisphere or hemicylinder.

9. +CON= MAXIMUM CONSTRAINT...Enter the maximum value you wish to allow the search parameter to obtain. The maximum constraint must be larger than the mimimum constraint but need not be a positive number. The default value of the maximum constraint is 10^{11}.

   NOTE: For a PITCH search, the maximum constraint is calculated from the data entered for MAXPITCH, item 1. +CON should not be entered if a value was entered for MAXPITCH. If constraints, +CON= and/or -CON= are not entered as data, the code computes the minimum constraint corresponding to the pins touching, and the maximum constraint is then five times the magnitude of the minimum constraint.

10. -CON= MINIMUM CONSTRAINT...Enter the minimum value you wish to allow the search parameter to obtain. The minimum constraint must be smaller than the maximum constraint but need not be a negative number. The default value of the minimum constraint corresponds to the pins in the lattice touching.

NUREG/CR-0200
Volume 1. Section S2
ORNL/NUREG/CSD-2/V1/R3

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

SAS2: A COUPLED ONE-DIMENSIONAL DEPLETION AND SHIELDING ANALYSIS CODE

O. W. Hermann

Manuscript Completed: February 1983
Date Published: June 1983
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

S2.2.6 THE GAMMA SOURCE SPECTRUM

The last subcase of the ORIGEN-S case described in the last subsection invokes the option for computing the gamma source spectrum of the spent-fuel assembly. The photon energy-group-structure of the SCALE cross-section library to be used in the shipping cask shielding analysis is applied.

The code first converts inventories of all nuclides of the cooled fuel assembly to disintegrations per second. Then, applying the ORNL Master Photon Data Base,[2] it sums individual nuclide photon spectra to determine the total gamma source spectrum. The intensity of a line at energy E, from the data base, is normalized to $\bar{E}$, the average energy of the group, using direct multiplication by the factor $E/\bar{E}$, with one exception. In cases where $E-E_1$ or E_2-E is less than 0.03 (E_2-E_1), where E_1 and E_2 are the boundaries of a group, one-half the initial intensity is applied to each of the two groups having the boundary near the line. These procedures maintain the conservation of energy rather than photon intensity, which should give a more correct computation of dose rates in the shielding analysis. As a final correction, the ratio of total nuclide gamma energy, from data in the ORIGEN-S Working Library, to the gamma energy of only those nuclides having line data, is multiplied times the spectrum computed from the data base. Later, the spectrum is converted to a uniform volumetric photon source for the shielding analysis.

The ORNL Master Photon Data Base[2] includes photon intensity and energy data for the following physical processes: x-rays and gamma rays emitted during radioactive decay of 427 nuclides, or more, taken from the Evaluated Nuclear Structure Data File (ENSDF)[3,4]; prompt fission gamma rays from spontaneous fission of heavy nuclides; gamma rays from the fission products produced in spontaneous fission; gamma rays from (α,n) reactions for heavy metal dioxides; and bremsstrahlung production from beta and positron deceleration in UO_2 and water using beta energy data from ENSDF (see Sect. M6).

S2.2.7 THE NEUTRON SOURCE SPECTRUM

An energy-dependent neutron source is required for computing neutron dose rates in the shielding analysis. Since ORIGEN-S did not calculate neutron source spectra at the time SAS2 was written, the procedure for computing neutron spectra described here was developed for SAS2. Also, substantially improved spontaneous fission and (α,n) reaction yield data were applied. The data and method previously used in ORIGEN-S and ORIGEN[5] over-predicted the (α,n) yields by a factor of $\sim$2. The complete model described here has subsequently been incorporated in ORIGEN-S.

The neutron source applied by SAS2 includes that produced from both spontaneous fission and (α,n) reactions of heavy nuclides. The significance of excluding neutrons produced by photofission and photoneutron reactions is discussed later in this subsection.

The major part of the neutron source is produced from spontaneous fission of the heavy nuclides. Data required to compute the neutron production rate from this process include the spontaneous fission half-life, the average neutron yield per spontaneous fission, ν_{sf}, and the concentration for each contributing nuclide. Spontaneous fission half-lives for the more significant nuclides are those from the recently updated[6] ORIGEN-S card-image actinide nuclear data library (Sect. M6). These half-lives are taken from ENSDF[3,4] and from Kocher's recent compilation of decay data,[7] both of which contain evaluated measured data. For several less important nuclides, unmeasured half-lives are taken from Ref. 8. These data were estimated with a correlation between measured data and so-called fissility parameters.[9] The ν_{sf} data are taken from Ref. 8. Measured values are available for 21 nuclides, including the most significant. An equation, derived[8] to compute ν_{sf}, produces values which are within two experimental standard deviations for all except three nuclides. This equation is applied by SAS2 for nuclides that do not have measured data.

A significant neutron source is produced from $^{17}O(\alpha,n)$ and $^{18}O(\alpha,n)$ reactions in the UO_2 and other oxygen compounds of the spent fuel. Thin target cross sections for these reactions and alpha stopping power data may be applied to compute neutron yields of the fuel material. Measurements[10] of thin target cross sections for the $^{17}O(\alpha,n)$ and $^{18}O(\alpha,n)$ reactions produced improvement over earlier data.[11,12] Additionally, thick target energy-dependent (α,n) yields for $^{238}U^{NAT}O_2$ were computed,[10] having estimated accuracies within 10%. SAS2 applies these yield data from Ref. 10 to weighted energy averages of alpha energy-intensity data[7] of all nuclides except for those reported[13] for ^{214}Bi, ^{241}Pu and ^{249}Bk, which have very small alpha branching fractions. Decay constants and alpha decay branching fractions are required to compute the (α,n) source. The decay constants are read directly from the ORIGEN-S Binary Library. However, since alpha decay branching fractions are not supplied directly from an ORIGEN-S Binary Library, the data are provided in the Block Data COMMON/SPECDT/. The data were copied from the current version (updated in 1981-1982) of the ORIGEN-S card image nuclear data library (Sect. M6). Also, since alpha energies are not explicitly included in ORIGEN-S libraries, the weighted averages of the most current (1981) evaluation[7] are also supplied in COMMON/SPECDT/. All of the nuclide yields and associated data are edited during the SAS2 execution.

The isotopes ^{242}Cm and ^{244}Cm characteristically produce all except a few percent of the spontaneous fission and (α,n) neutron source in spent PWR fuel over a 10-year decay time. The next largest contribution is usually from the (α,n) reaction of alphas from ^{238}Pu, which is approximately 1 to 2% of the source. Neutron energy spectra of both the

B. Sets of Interface Data (By Code, Case-Type):

(1) BONAMI-S, depletion unit-cell;
(2) NITAWL-S, depletion unit-cell;
(3) XSDRNPM-S, depletion unit-cell;
(4) COUPLE, depletion library update;
(5) ORIGEN-S, time-dependent depletion densities;
(6) ORIGEN-S, final depletion and decay case;
(7) BONAMI-S, cell-weight shield unit-cell;
(8) NITAWL-S, cell-weight shield unit-cell;
(9) XSDRNPM-S, cell-weight shield unit-cell;
(10) BONAMI-S, region geometry shipping cask;
(11) NITAWL-S, shipping cask library merge;
(12) XSDRNPM-S, region geometry shipping cask;

C. Final Records:

(1) Dose rate detector distances from shipping cask surface.
(2) Shielding library unit number, group sizes, and fuel zone size data.

S2.4.9 UTILITY ROUTINES

Utility routines applied by SAS2, which are not described above or in other SCALE sections, are briefly presented. The labeled common /FUELID/ is also included.

CHARIN - Reads characters for "Halt" and "Restart" features from "=SAS2" card.

COPYNX - Copies code interface data, with minor updating, to MAS for BONAMI-S, NITAWL-S and XSDRNPM-S.

ENDNOW - Prints STOP 444 when "Halt" feature invoked, writing restart data on unit number 55.

ERROR - Calls ABEND and prints STOP 555, when an error occurs.

FINDID - Converts SCALE ID to ORIGEN-S ID and library type no.

/FUELID/ - This is a common that is initialized in a BLOCK DATA routine. It contains the list of nuclides which SAS2 always inputs in trace quantities (1×10^{-20} atoms/b-cm) when not input by user. The nuclide list is: ^{135}Xe, ^{133}Cs, ^{234}U, ^{235}U, ^{236}U, ^{238}U, ^{237}Np, ^{238}Pu, ^{239}Pu, ^{240}Pu, ^{241}Pu, ^{242}Pu, ^{241}Am, ^{242m}Am, ^{242}Am, ^{243}Am, ^{242}Cm, ^{243}Cm, ^{244}Cm, and the 1/v Absorber. The 1/v Absorber is used to calculate the THERM parameter applied in ORIGEN-S (Sects. F6.2.1 and F.6.12).

GEPSIG - Computes $1/L_g$ in Eq. (S2.2.2) for mesh generator.

GETLIB - Converts input library name to unit number in standard SCALE JCL procedure.

IDXCIT - Computes ID for excited state nuclides in ORIGEN-S.

NEWREC - Writes and possibly edits one pair of code interface records.

SHLIB - Reads number of energy groups and nuclide IDs of a SCALE "master" library.

SHXI - Determines scattering power of nuclide for SHRES preprocessing of resonance data.

UPDATE - Copies standard interface formatted data until the array identifier, IA, and data entry number, IBGN, are found. Only reads the record pair requested. Usually, the calling routine updates the record pair with NEWREC, BINRCX or BINRCF. When the output unit number is zero, it simply reads or skips to proper record. It returns, also, after reading identifier "-1", since this signals the end of a code interface. Thus, a complete code interface, excluding the single record of title for XSDRNPM-S, is skipped by requesting an identifier -1 (e.g., -9, as usually sent by SAS2).

WI/WJ - Debug type edit of: mame, array size, and array. WI is for floating point data and WJ is for integers.

WRFLUX - Writes "flux guess" array, of all values equal to 1, and "moments" array.

WRTBY1 - Writes pair of interface records for all nonzero parameters in data list, L, for array identifier, IA. This is similar to specifying all data with an Ai in FIDO (Sect. M10), where i denotes the ith entry. This method allows the use of UPDATE more effectively.

S2.4.10 SUBROUTINE REFERENCE INDEX

The reference indices to subroutines written for SAS2 or directly called in the new SAS2 coding are given below. The functions of other subroutines, e.g., those applied by KNIGHT, are presented in Subsections C4.3.4 through C4.3.7, inclusive. The conventions and applications of Free-Form Reading Routines are given in Sect. M3.

Name	Subsections (Described or Applied)
ALOCAT	C4.3.2, S2.4.1, S2.4.2, S2.4.6
AREAD	M3.3, S2.4.3, S2.4.5
BINRCF	C4.3.3, C1.3, S2.4.2, S2.4.3
BINRCX	C4.3.3, C1.3, S2.4.2, S2.4.3, S2.4.5
CASCYC	S2.4.6
CELLWT	S2.4.5
CHARIN	S2.4.1, S2.4.9
CLPS	S2.4.7
COPYNX	S2.4.3, S2.4.9

Table S2.5.1 (continued)

NAME OR KEYWORD	DEFINITIONS, COMMENTS AND EXAMPLES
	(Enter the following set for each cycle. Entries may be in any order, except for "END".)
POWER**	Average specific power of assembly, megawatts.
BURN**	Fuel irradiation period of reactor, days.
DOWN**	Downtime, days, following BURN; except for last cycle, it is cooling time of spent-fuel for shipment.
BFRAC**	Fraction of first-cycle boron density for this cycle (1).
H2OFRAC**	Fraction of first-cycle H2O density for this cycle. Note, "O" in H2OFRAC is a letter. (1).
END*	Required if any entry omitted and POWER is not next entry. Example: POWER=15.2 BURN=330 DOWN=35.25 END
	(Repeat the following LIGHTEL times, if LIGHTEL>0).
EL	Chemical symbol of element for which gamma source is to be included and (n,γ) Q-value/fission applied.
WTLITE	Effective weight/assembly, kg. See Data Note D. Example: FE 6.2 CO 0.04 ZR 100
ALIB	Name of shipping-cask-case SCALE library: 27N-18COUPLE; 22N-18COUPLE; 18GROUPGAMMA; or those listed for LIB. (The following entries may be in any order.)
TEMPCASK**	Cask temperature, used for all mixtures, K.
NUMZONES=*	Number of material zones in cask, i.e., number of radii, R.
DRYFUEL=*	Enter YES (if dry, i.e., no H2O) or NO to denote removal of fuel zone water derived from the moderator mixture, MX=3. Follow with two or more blanks. (NO).
DETECTORS=*	Number (integer) of dose detector locations. Limit of 30. If zero, locations are 0, 1, 2, and 4 meters from cask surface in midplane. If > 0, or not entered, ANSI standard dose conversion factors are used. If < 0, other factors, possibly, are used (see Data Note F) (0).
END*	Required only if one of the above entries is omitted. Example: 27N-18COUPLE TEMPCASK=300 NUMZONES=6 END
	(|DETECTORS| number of entries. Omit, if DETECTORS=0.)
DTEC	Surface-to-detector distance, cm. Example: 0 34.5 100 500
	(Repeat pair of entries NUMZONES times. See Data Note E.)
MXZ	Zone mixture number; applies all SC where MX=MXZ.
R	Radius of zone, cm. Example: 4 24.5 6 24.8 5 50.2 6 50.4 7 60.8 6 61
	(Repeat pair of entries for each fuel zone; limit of 5.)
ZONE**	Zone number of fuel zone, or position of MXZ-R pair.
FUELBNDL**	Number of spent-fuel assemblies in zone. See Data Note E. Example: ZONE=1 FUELBNDL=1 ZONE=3 FUELBNDL=5
	(Enter, if required or for changing defaults; in any order.)
PRESSURE**	Cask water pressure, psig, for deriving densities. Either this or next entry required (0.0).
DENFM**	Ratio of coolant density to that of the same moderator in reactor-case. Always required in treating sodium. (In entries below, x=C or x=S. The entry is for the cell-weight case if x=C, and for the shipping cask case if x=S.)
SZFCASK**	Mesh size factor: Interval size=default size*SZFCASK (1).
ISNx**	Order of angular quadrature (8).
IOMx**	Maximum number of outer iterations (25).
EPSx**	Overall convergence criteria (0.0001).
PTCx**	Scalar flux point convergence (0.0001).
ITMC**	Maximum computer (CPU) time allowed in XSDRNPM-S, min. (10).
ITMS	Maximum CPU time allowed in XSDRNPM-S, min. (30).
END*	End of SAS2 module data. A new SCALE case may follow.

*Word, words or name (including *), which must be part of data entry.

that in the above methods and ADEN is omitted and then the sum of all WTP input must equal 100%.

An arbitrary application of the material definition data is required in SAS2 in order to apply the total material description to both the unit-cell calculation of the reactor and the shipping cask shielding analysis. The first three mixtures (MX≤3) apply to the assembly pin-cell depletion cases, while larger mixture numbers apply to the multiregion geometry shipping cask case. Specifically, apply MX=1 for the fuel mixture, MX=2 for the pin-clad, and MX=3 for the moderator mixture. (Note that material between the fuel rods, such as spacers, might be represented in the latter mixture by either conserving mass or using some other more appropriate approximation.) Since only one reactor assembly is described in the SAS2 case, the user must decide which type to apply, i.e., an average-type or worst-case-type assembly (also, see Data Note B). Since all nuclides that are generated during the irradiation period and have significant effects upon the problem must be input to reactor analysis cases of SAS2, the code automatically introduces the most important nuclides in trace amounts (10^{-20} atoms/b-cm) when not already input by the user. The nuclides must be on the master library and in the list for /FUELID/ (Sect. S2.4.9). Other nuclides may be input by the user by entering SC, e.g., TH-232, and also MX=1, VF=0, and ADEN=1-20.

All mixtures denoted by MX≥4 pertain to shipping cask materials. Three of the materials are not input as Standard Composition data in the blocks of data described here. They are: (1) the spent-fuel material, which is computed and applied by the code; (2) the clad, which is derived from the code for each fuel zone; and, (3) the shipping cask coolant (or part of the coolant, at least, as explained below), which is derived from other input data. These three materials are part of that contained in a "fuel zone," which is identified in other input data (see Data Note E).

For each fuel zone, a unique mixture number must be available to identify the extra material to be added to the zone for the shielding analysis. For I fuel zones, the extra material mixture numbers must be MX=3+i, i=1, I. Nonfuel zones then have mixture numbers MX>3+I. If no extra material is desired for a fuel zone, the user should supply a dummy mixture with VF=0 and ADEN=1-20.

Extra material density for a fuel zone is determined with the same methods as used for other materials. Its zone-averaged atom density may be entered in ADEN, or, if an element or compound is at full theoretical density, its volume fraction in the zone may be entered in VF. Otherwise, VF=(fraction of the zone volume)*(fraction of theoretical density). Usually, for the wet cask, extra water must be added by using the following rule. Compute the extra water volume from: (the zone volume) - (other extra material volume) - (FUELBNDL*NPIN/ASSM*PITCH * FUELENGTH). Then, the volume fraction, and finally VF, may be derived. The code, as implied by the rule, supplies all data of the cell-weighted unit-cell for the number of pins in the assemblies. (Also, the height of all volumes is assumed to be FUELENGTH.) Note that setting

NUREG/CR-0200
Volume 1, Section H1
ORNL/NUREG/CSD-2/V1/R3

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge Gaseous Diffusion Plant
Post Office Box P
Oak Ridge, Tennessee 37830

HTAS1: A TWO-DIMENSIONAL HEAT TRANSFER ANALYSIS OF FUEL CASKS

W. D. Turner
C. K. Cobb

Manuscript Completed - December 1982
Date Published - June 1983
Date Revised - December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, D. C. 20555
Under Interagency Agreement DOE 40-549-75

NRC FIN No. B0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

TABLE OF CONTENTS

Page

LIST OF FIGURES H1.v
LIST OF TABLES H1.vii
ACKNOWLEDGMENTS H1.ix
ABSTRACT H1.xi
H1.1 INTRODUCTION TO HTAS1 ..... H1.1.1
H1.2 USE OF HTAS1 ..... H1.2.1

H1.2.1 DESCRIPTION OF THE PHYSICAL MODEL H1.2.1
H1.2.2 DISCUSSION OF THE THERMAL PROPERTIES ASSOCIATED WITH EACH ZONE H1.2.8
H1.2.3 DESCRIPTION OF THERMAL ANALYSES AVAILABLE FOR ANALYTICAL SEQUENCES H1.2.12
H1.2.4 DEFINITION OF BOUNDARY CONDITIONS H1.2.14
H1.2.5 MISCELLANEOUS CONSIDERATIONS IN MODELING A PROBLEM H1.2.16

H1.2.5.1 Location of the Neutron Shield, Water Jacket, and Impact Limiter H1.2.16
H1.2.5.2 Modeling of the Neutron Shield and Water Jacket H1.2.16
H1.2.5.3 Modeling of the Impact Limiter H1.2.18
H1.2.5.4 Other Modeling Considerations H1.2.18

H1.3 LOGICAL PROGRAM FLOW AND DESCRIPTION OF SUBROUTINES H1.3.1

H1.3.1 DISCUSSION OF INTERACTION WITH THE DRIVER AND HEATING6 ..... H1.3.1
H1.3.2 DESCRIPTION OF LOGICAL UNITS UTILIZED BY HTAS1 ..... H1.3.3
H1.3.3 RESTARTING HTAS1 ..... H1.3.4
H1.3.4 DESCRIPTION OF SUBROUTINES IN HTAS1 ..... H1.3.4
H1.3.5 DESCRIPTION OF OAK RIDGE SYSTEMS ROUTINES USED BY HTAS1 ..... H1.3.12

H1.4 INPUT DATA DESCRIPTION FOR HTAS1 ..... H1.4.1

H1.4.1 INTRODUCTION H1.4.1
H1.4.2 CARD 1 - TITLE OF PROBLEM H1.4.4
H1.4.3 CARD 2 - PROBLEM CONTROL PARAMETERS H1.4.4

H1.4.3.1 Initial Temperature Input Option (I) H1.4.4
H1.4.3.2 Final Temperature Output Option (I) H1.4.4
H1.4.3.3 Maximum Number of Printout Times (I) H1.4.5
H1.4.3.4 Maximum Number of Ordered Pairs per Tabular Function (I) H1.4.5
H1.4.3.5 Maximum Number of Tabular Functions (I) H1.4.5

H1.4.4 TRANSIENT OUTPUT H1.4.5

H1.4.4.1 Printout Times Data Block H1.4.5
H1.4.4.2 Melting Ratio Output H1.4.6

TABLE OF CONTENTS (cont.)

Page

H1.4 INPUT DATA DESCRIPTION FOR HTAS1 (cont.)

H1.4.5 ZONE DATA BLOCKS H1.4.6

H1.4.5.1 Cavity Zone H1.4.7
H1.4.5.2 Inner Shell Zone H1.4.7
H1.4.5.3 Shielding Zone H1.4.7
H1.4.5.4 Outer Shell Zone H1.4.8
H1.4.5.5 Neutron Shield Zone H1.4.9
H1.4.5.6 Water Jacket Zone H1.4.9
H1.4.5.7 Impact Limiter Zone H1.4.10

H1.4.6 MATERIAL DATA BLOCK H1.4.11

H1.4.6.1 Material Card H1.4.11
H1.4.6.2 Material Name Card H1.4.12
H1.4.6.3 Thermal Conductivity Card H1.4.12
H1.4.6.4 Density Card H1.4.12
H1.4.6.5 Specific Heat Card H1.4.13

H1.4.7 EMISSIVITY DATA H1.4.13
H1.4.8 CHANGE OF PHASE OPTION H1.4.14
H1.4.9 ANALYTICAL SEQUENCE DATA BLOCKS H1.4.15

H1.4.9.1 INITIAL Steady-State Thermal Analysis H1.4.17
H1.4.9.2 PREFIRE Steady-State Thermal Analysis H1.4.19
H1.4.9.3 FIRE Transient Thermal Analysis H1.4.20
H1.4.9.4 POSTFIRE Transient Thermal Analysis H1.4.24
H1.4.9.5 FINAL Steady-State Thermal Analysis H1.4.26

H1.4.10 MODIFYING THE MODEL DURING AN ANALYTICAL SEQUENCE H1.4.27

H1.4.10.1 Neutron Shield H1.4.27
H1.4.10.2 Water Jacket H1.4.28
H1.4.10.3 Impact Limiter H1.4.28

H1.4.11 FINNED SURFACE OPTION H1.4.29
H1.4.12 PLOTTING DATA BLOCKS H1.4.30

H1.4.12.1 Temperature Profile Plots Data Card H1.4.30
H1.4.12.2 Region Plots Data Block H1.4.31

H1.5 SAMPLE PROBLEMS FOR HTAS1 ..... H1.5.1
H1.6 REFERENCES H1.6.1

APPENDICES

H1.A. CONTROL CARDS FOR EXECUTING HTAS1 ON ONE OF THE OAK RIDGE IBM SYSTEMS H1.A.1
H1.B. MICROFICHE OF COMPLETE OUTPUT FROM HTAS1 FOR SAMPLE PROBLEMS H1.B.1

LIST OF FIGURES

		Page
H1.1.1.	Zone Configuration in HTAS1 General Model	H1.1.2
H1.2.1.	Zone Configuration in HTAS1 General Model with Physical Dimension Labels	H1.2.2
H1.2.2.	Location of Boundary Conditions in HTAS1 Basic Model (Cavity, Inner Shell, Shielding, Outer Shell)	H1.2.3
H1.2.3.	Location of Boundary Conditions in HTAS1 Basic Model with Addition of Neutron Shield and Water Jacket	H1.2.4
H1.2.4.	Location of Boundary Conditions in HTAS1 Basic Model with Addition of Neutron Shield and Water Jacket after Neutron Shield Has Been Deleted	H1.2.5
H1.2.5.	Location of Boundary Conditions in HTAS1 General Model	H1.2.6
H1.2.6.	Fin Effectiveness Models, Pictorial	H1.2.17
H1.5.1.	Temperature profile plots generated for the first sample problem	H1.5.26
H1.5.2.	Region plots generated for the second sample problem	H1.5.33
H1.A.1.	Job control language for executing HTAS1 at ORNL	H1.A.2
H1.A.2.	Job control language for executing HTAS1 at ORNL and generating region plots on an 11-inch Versatec plotter	H1.A.2
H1.A.3.	Job control language for executing HTAS1 at ORNL and storing a compressed plot file containing temperature profile plots in the user's PDP-10 area	H1.A.4
H1.A.4.	Job control language for executing HTAS1 at ORGDP	H1.A.4
H1.A.5.	Job control language for executing HTAS1 at ORGDP and generating region plots on a CALCOMP plotter	H1.A.5
H1.A.6.	Job control language for executing HTAS1 at ORGDP and generating temperature profile plots on a CALCOMP plotter	H1.A.5

LIST OF TABLES

		Page
H1.2.1.	Keywords for Zones in HTAS1 Model	H1.2.9
H1.2.2.	HTAS1 Material Property Library	H1.2.10
H1.2.3.	Default Emissivities on Outer Surface of Zones Modeled by HTAS1	H1.2.13
H1.3.1.	Description of Logical Units Utilized by HTAS1	H1.3.3
H1.4.1.	Units for Input Data to HTAS1 and HEATING6	H1.4.2
H1.4.2.	Keywords for Analytical Sequences for HTAS1 Model	H1.4.3
H1.5.1.	Input Data for First Sample Problem for HTAS1	H1.5.3
H1.5.2.	Standard Output from Initial Entry into HTAS1	H1.5.3
H1.5.3.	Temperature Distribution at the Prefire Steady State	H1.5.13
H1.5.4.	Temperature Distribution at the End of the 30-Minute Fire Transient	H1.5.16
H1.5.5.	Temperature Distribution at the End of the Three-Hour, Cool-Down Transient	H1.5.19
H1.5.6.	Temperature Distribution at the Final Steady State	H1.5.22
H1.5.7.	Output Describing the Temperature Profile Plots Generated by HTAS1	H1.5.25
H1.5.8.	Thermal Properties for 1.5% Carbon Steel	H1.5.29
H1.5.9.	Input Data for Second Sample Problem for HTAS1	H1.5.31
H1.A.1.	SPDA Retention Times	H1.A.8

ACKNOWLEDGMENTS

This work was funded by the Transportation Branch, Nuclear Material Safety and Safeguards, U. S. Nuclear Regulatory Commission. HTAS1 was conceived by R. H. Odegaarden of the U. S. Nuclear Regulatory Commission whose ideas and guidance proved invaluable in the design of the control module. Appreciation is expressed to C. V. Parks who served as project leader of this work and to L. M. Petrie who provided valuable information concerning the use of the SCALE system. Special thanks are extended to A. I. Cole for patiently typing this document, to B. R. Ferguson and L. F. Norris for the layout of the text, and to M. H. Cuthbert for editorial review.

ABSTRACT

The Heat Transfer Analysis Sequence Number One (HTAS1) control module was developed for the United States Nuclear Regulatory Commission to perform standardized thermal analyses on a class of nuclear fuel shipping containers. HTAS1, a control module in the SCALE system where SCALE is an acronym for Standardized Computer Analyses for Licensing Evaluation, uses the HEATING6 functional module to perform the prescribed thermal analysis. The nuclear fuel shipping container must be composed of acceptable combinations of zones consisting of a cavity, an inner shell, shielding, an outer shell, a neutron shield, a water jacket, and impact limiters. Thermal properties for the material in each zone may be extracted from a material property library or may be supplied by the user. The desired thermal analysis of the nuclear fuel shipping container is defined by an analytical sequence composed of one or more of the following calculations chosen in an acceptable order: an initial steady state; a prefire steady state; a fire transient; a postfire transient; and a final steady state. Each calculation in an analytical sequence specifies the type (steady state or transient), the ambient temperature, the duration of any transient, and the heat transfer mechanisms at the surface of the model. Selected portions of the model may be deleted at certain points in an analytical sequence to simulate changes in the nuclear fuel shipping container during an analysis. The model and analytical sequence are described to HTAS1 using free-form, keyword-oriented data making extensive use of defaults. HTAS1 generates the HEATING6 input data necessary to model each calculation in an analytical sequence and transfers control to the SCALE driver which accesses the HEATING6 functional module to perform the desired analysis.

The HTAS1 control module has recently been revised since its initial release to allow various geometry and output plots to be generated and to update default values to match those specified by 10CFR71.

where

σ is the Stefan-Boltzmann constant or 0.173×10^{-8} Btu/(hr-ft^2-°R^4),
ε_s is the emissivity of the material on the surface of the model.

The emissivity for each material will be extracted from HTAS1's material property library (see Table H1.2.2) unless it is provided by the user when a particular zone is defined. The emissivity is defined independently for each thermal analysis to be performed on the model. During the fire, the effective radiative heat transfer coefficient used by HTAS1 has the form (ref. 9)

$$h_r = \sigma\left(\frac{1}{\frac{1}{\varepsilon_s} + \frac{1}{\varepsilon_f} - 1}\right)(T_s^2 + T_f^2)(T_s + T_f) \quad , \qquad \text{(H1.2.5.)}$$

where

ε_f is the emissivity of the fire,
T_f^f is the temperature of the fire.

In keeping with the specifications in Part 71 of Title 10, Chapter 1, Code of Federal Regulations - Energy, United States Nuclear Regulatory Commission, the default values for the emissivity of the fire is 0.9 and for the emissivity on the surfaces of the package during the fire is 0.8. Also, the default temperature of the fire is 1475°F. However, the default value of each of these parameters may be overridden.

The default value for the solar heat load is 122.92 Btu/(hr-ft^2) (ref. 8) except the solar heat load is not applied to the surface of the package during the fire analysis. The user may override these defaults and specify a heat flux across the surface of the package for any thermal analysis.

When the material in the neutron shield is deleted or replaced, HTAS1 generates a boundary condition to model one-dimensional radiative heat transfer between the opposing radial surfaces of the neutron shield. The effective heat transfer coefficient is calculated as

$$h_r = \sigma\left(\frac{1}{\frac{1}{\varepsilon_{is}} + \frac{r_{is}}{r_{os}}\left(\frac{1}{\varepsilon_{os}} - 1\right)}\right)(T_{is}^2 + T_{os}^2)(T_{is} + T_{os}) \quad , \qquad \text{(H1.2.6.)}$$

where

ε_{is} is the emissivity of the inner surface (default is 0.5),
ε_{os} is the emissivity of the outer surface (default is 0.5),
r_{is} is the radius of the inner surface,
r_{os} is the radius of the outer surface,
T_{is} is the temperature of the node in question on the inner radial surface,
T_{os} is the temperature of the opposing node on the outer radial surface.

Under the default conditions, HTAS1 assumes all surfaces are unfirned. However, this default can be overridden by defining one of HEATING6's eight fin effectiveness techniques on the outer radial and/or axial surfaces of any zone that is part of the outer surface of the model (see Fig. H1.2.6).

H1.2.5 MISCELLANEOUS CONSIDERATIONS IN MODELING A PROBLEM

H1.2.5.1 Location of the Neutron Shield, Water Jacket, and Impact Limiter

HTAS1 generates the fine grid lines for the model based on the grid spacing information supplied with the zone data. The axial fine grid lines are based on the axial grid spacing information provided for the cavity, inner shell, shielding, outer shell, and impact limiters. If one of the axial boundaries of the neutron shield and water jacket does not lie on one of these zone boundaries or on one of the fine grid lines generated by HTAS1, then HTAS1 will move the axial boundary of the neutron shield and water jacket to the nearest axial fine grid line. It will also generate a warning message indicating this action. If one of the impact limiters wraps around the package, HTAS1 will locate its axial extent in a similar manner. The radial fine grid lines are based on the radial grid spacing information provided for the cavity, inner shell, shielding, outer shell, neutron shield, water jacket and impact limiter if the impact limiter extends radially beyond the water jacket. If the radius of the impact limiter does not extend beyond the radius of the water jacket, it must lie on one of the boundaries of the neutron shield or water jacket or on one of the fine grid lines within those zones. If this is not the case, HTAS1 will move the radius of the impact limiter to the nearest radial fine grid line and will generate a warning message indicating this action. HTAS1 follows this procedure to minimize the occurrence of an unfavorable distribution of nodes in the model. As an example, large differences in the size of coefficients in the system of equations describing the temperature distribution can lead to numerical problems when an attempt is made to solve the system. This can happen when large differences exist between node sizes throughout the model, especially if they are within the same zone. It should be noted that when the boundaries of one of the zones is moved, the resulting model will be slightly different from the one which was input. For instance, the volume and heat flow paths will be slightly changed. This error will decrease as the grid spacing decreases.

H1.2.5.2 Modeling of the Neutron Shield and Water Jacket

The water jacket actually wraps around the neutron shield and extends to the surface of the outer shell. The top and bottom of the water jacket tends to serve as a heat flow path between the water jacket and outer shell since its thermal conductivity will generally be higher than that of the neutron shield. HTAS1 assumes this heat flow path is small, and thus it is not modeled in order to minimize the occurrence of an unfavorable nodal configuration. The effects of this will be small unless the axial thickness of the water jacket at one of the ends of the neutron shield is large. The effects of radiative heat transfer on the ends of the water

of the water jacket during the fire would be the temperature of the fire or 1475°F. Thus, the temperature difference across the air-filled neutron shield will range from approximately 6°F immediately following the rupture of the water jacket to -1225°F during the fire. Since the temperature on the surface of the outer shell will increase during the fire transient and since the temperature on the inner surface of the water jacket will be less than the fire temperature, the temperature difference will not really be that large. It will also vary along the axial length of the water jacket. During the postfire transient, the temperature difference will increase to zero and will gradually increase to a maximum of several hundred degrees Fahrenheit. Thus, the absolute value of the temperature difference across the air-filled annulus will range from 0°F to around 1000°F.

If the average air temperatur is assumed to be the average of the temperatures of the opposing surfaces, then it will range from 275°F to around 1000°F. To estimate the effects of natural convection through the annulus, a number of hand calculations were performed using the correlation of Lui, Mueller, and Landis (ref. 10) to obtain the effective thermal conductivity of the air-filled zone. The calculations were performed with the average air temperature ranging from 250°F to 1000°F and the temperature difference across the annulus ranging from 10°F to 1000°F. The effective thermal conductivity ranged from 0.06 to 0.15 Btu/hr-ft-°F. The thermal conductivity of air ranges from 0.01944 Btu/hr-ft-°F at 263°F to 0.03483 Btu/hr-ft-°F at 1073°F (ref. 3). Based on this analysis, the effective thermal conductivity of the air in the annulus would range from a minimum of 0.01944 Btu/hr-ft-°F with pure conduction to around 0.15 Btu/hr-ft-°F with natural convection. Since the effective thermal conducti.ity of the air will be a function of both the air temperature and the temperature difference across the annulus, since it will be a function of both axial location and time, and since a conservative value during the fire will not necessarily be conservative following the fire and vice versa, a constant value of 0.1 Btu/hr-ft-°F was used for the effective thermal conductivity of material 18, AIRCONV.

An analysis was performed to compare the amount of heat transferred by radiation with that transferred by conduction and natural convection. The emissivities on the opposing surfaces of the annulus were assumed to be the default of 0.5. One set of calculations held the temperature of the outer shell at 280°F while increasing the temperature of the water jacket from 275°F to 1475°F. The ratio of heat transferred by radiation to that transferred by conduction and natural convection with an effective thermal conductivity of 0.1 Btu/hr-ft-°F ranged from 4.3 to 29. The calculations were repeated by allowing the temperature of the outer shell to increase. As the outer shell temperature approached that of the water jacket, the above ratios increased for each water jacket temperature. Thus, the heat transferred across the annulus was largely controlled by radiation. As a final check on the effective thermal conductivity of the air in the annulus, a parametric study was run with HTAS1 on sample problem number 1 assuming the water jacket was ruptured following the prefire steady state. The emissivities on the opposing surfaces of the annulus were assumed to be the default of 0.5. Calculations were run with the effective thermal conductivity of the air ranging from 0 (or a void) to 0.2 Btu/hr-ft-°F.

The variations in the temperatures along the outer shell were at most a few percent during this parametric study. Thus, 0.1 Btu/hr-ft- F is a reasonable effective thermal conductivity to simulate the natural convective heat flow across a 4.5 in. annulus for the shipping container similar to the one modeled by sample problem number 1. This value should be reasonable for shipping packages having similar dimensions and material properties. However, the user should analyze the natural convective heat flow across an annulus for shipping packages basically different from sample problem number 1 prior to using the HTAS1 default properties for air with natural convection or material 18, AIRCONV.

Any support structure or internal fins that may be inside the neutron shield are not modeled. For a specific design, the user may approximate any such effects by developing an effective density and an effective thermal conductivity to simulate any change in the density and heat flow paths due to such a structure inside the neutron shield.

Since the internal heat load is applied as a heat flux across the surface of the inner shell, the actual fuel in the nuclear fuel transport package is not physically modeled, and thus its heat capacity effects are not included in the transient calculations.

H1.2.6 DESCRIPTION OF HTAS1 PLOTTING OPTIONS

Subroutines from REGPLOT6 (see Sect. F14) are included in HTAS1 so that the HEATING6 input generated by HTAS1 to describe the heat transfer model at each calculational phase can be graphically verified if the user selects the region plotting option. Maps of the regions, the materials, the heat generation function numbers, the initial temperature function numbers, and the boundary condition function numbers are generated by plotting the model regions and labeling each region appropriately. By default, plots of the regions, the materials, and the boundary condition function numbers are generated for each calculational phase. HEATING6 is not executed when region plots are generated. Therefore, a minimum of execution time can be used to verify that the input supplied to HTAS1 has produced the desired HEATING6 input.

Temperature profile plots may also be generated if the user desires. If the temperature profile plotting option is selected, one temperature distribution plot data set is written by HEATING6 during the fire transient calculations and another temperature distribution plot data set is written by HEATING6 during the postfire transient calculations. These data sets are written to logical units 18 and 19, respectively.

Because these temperature distribution plot data sets are written by HEATING6, they are written using a format that can be read by HEATPLOT-S (see Sect. F13). Therefore, the user may save the temperature distribution plot data sets and later execute HEATPLOT-S to generate plots using these data sets as input. HEATPLOT-S has the ability to generate temperature contours, temperature-time profiles, and temperature-distance profiles.

Three default plots are generated by HTAS1. The first plot contains temperature vs time curves for the nodes in the center of the inner shell, the shielding, and the outer shell. If a node is not located in exactly the center of a particular zone, the node closest to the center of that zone is used. If the shielding is not modeled, the node that represents the inner shell is located on the radial fine grid line that is the border between the cavity and the inner shell, the node that represents the shielding is located on the radial fine grid line that is the border between the inner shell and the outer shell, and the node that represents the outer shell is located on the radial fine grid line that denotes the outer boundary of the outer shell. The second plot contains a curve that shows maximum temperature at any problem time vs the radial distance from the origin for all nodes on the axial center line. The third plot contains a curve that shows the difference in volume-averaged temperatures between the inner shell and the outer shell as a function of time. The temperature at each node in a zone is multiplied by the volume of that node. These products are summed and the sum is divided by the volume of the entire zone to compute the volume-averaged temperature for a zone.

If both the fire transient and postfire transient calculations are performed by HEATING6, the time axis on plots includes problem times from both the fire transient and postfire transient calculational phases. If only the fire transient calculation is performed by HEATING6, then the time axis only includes problem times from the fire transient calculational phase. In a similar manner, if only the postfire transient calculation is performed by HEATING6, then the time axis only includes problem times from the postfire transient calculational phase. If neither the fire transient calculation nor the postfire transient calculation are performed by HEATING6, then temperature profile plots are not generated.

The user may choose from several options that allow plots to be generated in a manner other than the default plots which include data at all problem times for which temperature distributions exist. The plot of maximum temperature at any problem time vs the radial distance from the origin for all nodes on the axial center line may be omitted, as may the plot of the difference in volume-averaged temperature between the inner shell and the outer shell vs time. The temperature vs time curves for the centermost nodes of the inner shell, the shielding, and the outer shell may be drawn on individual plots. These same curves may be generated for temperature change from the initial temperature distribution instead of for the actual temperatures. Finally, a minimum and maximum time may be specified so that the time axis only includes a specified time interval. Combinations of these options may be specified. For example, it is possible to generate curves of temperature change for the inner shell, the shielding, and the outer shell on individual plots between specified minimum and maximum problem times.

Plots in HTAS1 are generated using the DISSPLA (Display Integrated Software System and Plotting Language) graphics package (ref. 12). A compressed plot data set is generated that can be stored on disk or routed to various plotting devices.

H1.3.2 DESCRIPTION OF LOGICAL UNITS UTILIZED BY HTAS1

The logical units used during an execution of HTAS1 are identified in Table H1.3.1. Data is transmitted between HTAS1 and HEATING6 using the files created on the logical units identified by KTPIN, KTPOUT, ICONT and IBIN. If HTAS1 is restarted, the initial temperature distribution must be supplied by the user on the logical unit identified by ITPIN. If the user directs HTAS1 to save the final temperature distribution, it will be written on the logical unit identified by ITPOUT. The user may choose these files to be binary or formatted. If these options are not selected, these units are not used. If the user references any materials in the Lawrence Livermore National Laboratory material property library (ref. 2), HEATING6 will select their values from the library on the logical unit identified by IMATLB.

Table H1.3.1 Description of Logical Units Utilized by HTAS1.

Unit Name	Unit Number	Type of Data	Creating Module	User Module
KTPIN	1	Binary Scratch File with HEATING6 Initial Temperatures	HTAS1	HEATING6
KTPOUT	2	Binary Scratch File with HEATING6 Final Temperatures	HEATING6	HTAS1
IN	4	Formatted Scratch File with HTAS1 Input Data	HTAS1	DRIVER
IECHO	5	Standard HTAS1 Input Data, Formatted	User	HTAS1
IO	6	Standard Output (Default, Printer)	HTAS1 HEATING6	User
IMATLB	78	Material Property Library (Default, LLNL, Formatted)		HEATING6
ICONT	91	Binary Scratch File	HTAS1	HTAS1
IBIN	95	HEATING6 Binary Input Data	HTAS1	DRIVER
ITPIN	Input	Initial Temperatures for Restart of HTAS1 (Binary or Formatted as Specified by the User)	User	HTAS1
ITPOUT	Input	Final Temperatures for Restart of HTAS1 (Binary or Formatted as Specified by the User)	HTAS1	User

H1.3.3 RESTARTING HTAS1

HEATING6 has provisions for restarting a thermal analysis when an earlier calculation was terminated prematurely. If this option is invoked, HEATING6 will generate a data set containing the final temperature distribution calculated for steady-state problems and the temperature distribution at the last time level that had satisfied the convergence criteria for transient problems. This temperature distribution may then be used to restart the HEATING6 calculations at a later date. This feature is used in the option for restarting a HTAS1 analytical sequence. The user may indicate to HTAS1 to save the HEATING6 final temperature distribution by entering the logical unit number on which the temperature distribution will be written (see Sect. H1.4.3.3). After the final HEATING6 calculation has been completed, HTAS1 will write the final HEATING6 temperature distribution on an unformatted data set on the specified unit. Later, the user may restart the analytical sequence by identifying the unit number from which the unformatted temperature distribution will be read. The HTAS1 input data deck must also be modified. Any analytical sequence data blocks that have been successfully completed must be removed. Any zones that were modified during the previous HTAS1 run must be modified in a similar manner in the input data for the restart run. Since HTAS1 does not allow radiative heat transfer across the neutron shield for the initial model in an analytical sequence, HTAS1 should not be restarted after the neutron shield has been deleted or changed to air. The nodal configuration for the restart model must be identical to the nodal configuration corresponding to the final temperature distribution to be used in the restart run. The logical unit numbers ITPIN and ITPOUT may not be equal to 1, 2, 4, 5, and 6, nor greater than 75. In addition, the appropriate data or file definition parameters must be supplied to describe the unformatted data sets or files on units ITPIN and ITPOUT.

H1.3.4 DESCRIPTION OF SUBROUTINES IN HTAS1

MAIN - Allocates buffer space for the necessary input and output units prior to the call to ALOCAT (see Sect. M2.2.1). Enters calculational portion of control module HTAS1 by instructing ALOCAT to call HTAS1A. Upon completion of each set of HTAS1 calculations, MAIN sets up the return to the SCALE driver instructing it to either call HEATING6 (alias Q#$010) or to terminate the calculations. Called from HTAS1. Calls HTAS1A and H1PALC. Calls SCALE routine ALOCAT.

BLOCK DATA - Initializes data in labeled common blocks ARYLN1, ARYLNG, FLGTYP, TSTOUT. IOUNIT, MAPFMT, DIVIDR, and WIDTH. Labeled common ARYLN1 contains default dimensions for some of the HTAS1 variably dimensioned arrays. The value of MXPRS, MXPRT, and MXTBL can be changed by entering the desired values on card 2. Labeled commons ARYLNG, FLGTYP, TSTOUT, IOUNIT, MAPFMT, DIVIDR, and WIDTH are from HEATING6. Although some of

these variables are used in HTAS1, they are mainly used to generate NAMELIST /OPTION/ for input to HEATING6. Values for the variables in these common blocks are initialized in this block data. Values for some of these variables are later computed by HTAS1 based on the model being generated.

DOUBLE - Copies the double word stored in the second argument into two consecutive elements of the single word array identified by the first argument. Called from HIP, OUT123, OUTGRD, OUTITP, OUTPRT, OUTSS, OUTTAB, OUTTT, WBC, WMAT and WREG.

ECHO - Reads input data card images from unit IECHO, lists input data card images on unit IO, puts ' END' after each data string, and writes the resulting input data card images on unit IN. Assumes card image with '@' in column 1 is a continuation of the previous card image. Replaces '@' in column 1 with a ' '. Assumes card image with '*' in column 1 is a comment card. Called from H6TAS1. Calls Oak Ridge Systems routine ICOMPA.

FGRID - Generates fine grid lines along the appropriate axis based on gross grid lines RG(IGT0) through RG(IGT1) and the spacings between the respective gross grid lines as stored in the NRG array. Stores generated fine grid lines in the array R and the total number generated in IT. Called from NODMAP.

FILL - Stores the character string 'END ' in the D-array immediately after the data string which extends through D(ID+NREC). Called from HIP, OUT123, OUTBC, OUTEND, OUTGRD, OUTITP, OUTMAT, OUTPRT, OUTREG, OUTSS, OUTTAB, OUTTT, WBC, WMAT, and WREG.

FINDLN - Checks the axial boundaries of the neutron shield, water jacket, and impact limiters. Generates a warning message and moves the respective axial boundary to the nearest axial fine grid line if one of these boundaries does not match a fine grid line generated by HTAS1 for the cavity, inner shell, shielding, or outer shell. Called from GENREG.

GENBC - Generates parameters for the HEATING6 B2 input data card. In particular, generates the coefficient for radiative heat transfer to include that for surface-to-surface heat transfer across the neutron shield, when appropriate, and the coefficient and exponent for the term simulating the effects of natural convective heat transfer. Generates the terms for each zone and for each calculational phase of the analytical sequence. Called from HTAS1.

GENDAT - Scans the input data of each zone for obvious errors. In particular, determines if required zones are present, if zones appear in proper combinations, if dimensions of each zone satisfy all constraints, if grid spacing information is

defined for each zone, if fin types are properly defined, and if the removal of a zone during calculations is properly defined. When an error is encountered, writes error message, continues to scan remaining input data, if possible, and sets flag to terminate calculations when error scanning is completed. Called from READER.

GENDEL - Determines the gross grid lines bounding the model during each calculational phase of the analytical sequence when the impact limiters or neutron shield and water jacket are deleted following a calculational phase. Called from H6TAS1.

GENREG - Examines each zone and determines the radial and axial gross grid lines and the number of divisions separating consecutive gross grid lines. This data will be used later to generate the HEATING6 XGRID and ZGRID data blocks. For cases where the radius of the impact limiters is not the outer radius of the model, also matches the radius of the impact limiters with radial fine grid lines generated by HTAS1 for the outer shell, neutron shield, or water jacket. If a match is not found, issues a warning message and moves the radius of the impact limiter to the nearest radial fine grid line. Checks model structure and, if any inconsistencies are located, issues a warning message and attempts to continue. Determines the maximum number of axial and radial gross grid lines, the maximum number of axial and radial fine grid lines, the maximum number of points, and the maximum bandwidth for steady-state models using the direct solution technique. Called from H6TAS1. Calls FINDLN.

H6TAS1 - Controls flow of the calculations in the HTAS1 control module. Initializes certain parameters for the calculations, calculates starting indices for the variably dimensioned arrays, calculates the starting point for the storage of the HEATING6 input data in the D-array, calculates the storage requirements for these arrays, notifies the user if the amount of core requested can be reduced, or writes an error message and terminates the calculations if not enough core was requested. If errors are detected in the input data or if the calculations cannot be continued, arranges for HTAS1 control module to tell the SCALE driver to terminate the calculations. On the initial entrance into HTAS1 control module, writes the generated HEATING6 input data on binary file on unit IBIN. On subsequent entrances, reads HEATING6 input data for the previous calculation in the analytical sequence, arranges for HIP to modify it for the next calculation, and writes the modified HEATING6 input data on unit IBIN. Called from HTAS1A. Calls ECHO, GENBC, GENDEL, GENREG, HIP, INITIL, INPUT, MATLIB, NODMAP, OUTDAT, READER, REGPLT, RGPINP, TABDAT and WTITLE. Calls SCALE routines IONUMS, RCRDLN and SCANON (see Sect. M2.2.61). Calls Oak Ridge systems routines MODEL and JSTIME.

WREG - Stores region data for region number NREG for the HEATING6 input data in the D-array. Counts the number of regions for HEATING6. Called from OUTREG. Calls DOUBLE and FILL.

WTITLE - Generates the descriptive heading for the HTAS1 output. Heading contains the code name and version date, the date and time the job was executed, the computer on which the job was executed and the jobname under which the job was executed. Includes the title for the job and the author of the code. Date of the current version of the code appears in the data initialization statement for V(2). Called from H6TAS1. Calls Oak Ridge systems routines IDAY, ITIME, JOBNUM and MODEL.

BCDRAW - Draws the boundary condition symbols when the plot of the model boundary condition function numbers is generated. Called from REGPLT.

H1PALC - Determines the dimension of the master array that contains all variably dimensioned arrays in the HTAS1 temperature plotting subroutines. Called from MAIN. Calls H1PINT.

H1PINT - Calculates starting locations for the variably dimensioned arrays. Writes messages describing which temperature plots will be generated. Called from H1PALC. Calls H1PLOT. Calls Oak Ridge systems routine MODEL.

H1PLOT - Generates curves of temperature vs time for the centermost nodes of the inner shell, the shielding, and the outer shell if the curves are drawn on the same plot. Generates the curve of maximum temperature at any problem time vs the radial distance from the origin for all nodes on the axial center line. Called from H1PINT. Calls MINMAX, NODNUM, ROUND, SEPRAT, SETUP, and VOLAVG. Calls Oak Ridge systems routine ICOMPA.

MINMAX - Determines the minimum and maximum values of the array received in the argument list. Called from H1PLOT, SEPRAT, and VOLAVG.

NODNUM - Determines the node number of the node that is nearest the center of either the inner shell, the shielding, or the outer shell. Called from H1PLOT.

REGPLT - Generates plots of the regions, materials, heat generation function numbers, initial temperature function numbers, and boundary conditions of the model. Called from H6TAS1. Calls BCDRAW. Calls Oak Ridge systems routine ICOMPA.

RGPINP - Stores data in common blocks that are used to generate REGPLOT6-type plots for the current phase of the calculation. Called from H6TAS1.

ROUND - Determines rounded axis numbers based on the minimum and maximum values to be plotted in a certain direction. Called from from H1PLOT, SEPRAT, and VOLAVG.

SEPRAT - Generates plots of temperature vs time for the centermost nodes of the inner shell, the shielding, and the outer shell if the curves are drawn on individual plots. Called from H1PLOT. Calls MINMAX, ROUND, and SETUP.

SETUP - Draws the axes and writes the heading for each plot. Called from H1PLOT, SEPRAT, and VOLAVG.

VOLAVG - Generates the plot of a curve showing the difference in volume-averaged temperature between the inner shell and the outer shell vs time. Called from H1PLOT. Calls MINMAX, ROUND, and SETUP.

H1.3.5 DESCRIPTION OF OAK RIDGE SYSTEMS ROUTINES USED BY HTAS1

This section contains a description of the Oak Ridge systems routines used by the HTAS1 control module. This information was extracted from ref. 11 with some minor revisions.

Function ICOMPA

This function is used in the following manner:

I = ICOMPA (ARG1, ARG2, N)

The result of the function, stored in the integer*4 variable I, is -1, 0, or 1. The function compares ARG1 and ARG2, and the number of bytes to be compared is specified by the integer*4 value N where N may be 1 through 256. If, for example, N is 1, then the first byte of ARG1 is compared with the first byte of ARG2. If N is 4, then the first four bytes of ARG1 are compared with the first four bytes of ARG2.

This function is particularly useful in comparing alphanumeric characters and avoids any overflow or underflow interrupts which might occur in the arithmetic subtraction of two variables containing alphanumeric data. Also, this function allows only the characters of interest to be compared thus eliminating the need for blank filling of alphanumeric data in some instances.

Comparison starts at the locations specified by ARG1 and ARG2 and proceeds from left to right, bit by bit. The result returned by the function will be 0 if the specified number of bytes of ARG1 are the same as those of ARG2; a 1 will be returned if, going from left to right, a bit in ARG1 is 1 and the corresponding bit in ARG2 is 0; a -1 will be returned if, going from left to right, a bit in ARG1 is 0 and the corresponding bit in ARG2 is 1.

If N is less than or equal to 0, the value returned by the function is 0. If N is greater than 256, the result will be meaningless. The function ICOMPA is also available under the name ICOMPARE, thus I = ICOMPARE (ARG1, ARG2, N).

Subroutine IDAY

This subroutine is called in the following manner:

```
CALL IDAY (WHEN)
```

The subroutine returns the date as eight EBCDIC characters consisting of the month, day, and year separated by minus signs, for example, 05-20-66. The eight-byte date will be placed in the location designated by the argument. The argument may be one word of type real*8 or a dimensioned variable of at least two words of type real*4 or integer*4. The date may then be printed using an A8 format (with a real*8 argument) or a 2A4 format (with two real*4 or integer*4 words).

Subroutine JOBNUM

The purpose of this routine is to return the jobname of the user's job. The jobname is the identification code on the user's JOB card which can consist of from three to eight characters.

The routine is used as follows:

```
CALL JOBNUM (Q)
```

where Q is a real*8 variable name. Eight characters are placed in Q with blank fill on the right if the name is less than eight characters.

Function JSTIME

This function returns the remaining job step time (CPU time) before cancellation (completion code 322) in hundredths of seconds. JSTIME may be called in two ways:

```
CALL JSTIME (I)  or
J=JSTIME (I)
```

where J and I are full-word binary integers containing the renaming job step time.

Function MODEL

This integer*4 function is used in the following manner:

```
M = MODEL (0)
```

The function returns the integer value 155, 195, or 3033 as the result depending on which computer the program is running. The argument of the function is not used, so 0 is suggested for use.

Subroutine TIME

This subroutine is called in the following manner:

CALL TIME (TIMNOW)

The subroutine returns the time of day as eight EBCDIC characters consisting of the hour, minutes, and seconds separated by periods, for example, 13.15.45. The eight-byte time will be placed in the location designated by the argument. The argument may be one word of type real*8 or a dimensioned variable of at least two words of type real*4 or integer*4. The date may then be printed using an A8 format (with real*8 argument) or a 2A4 format (with two real*4 or integer*4 words).

Seven different data blocks, denoted as zone data blocks, may be used to describe the zones to be modeled by HTAS1. (See Table H1.2.1.) A MATERIAL data block, an EMISSIVITY card, a LATENT HEAT card, and/or a DELETE card may appear in a zone data block to override default values assigned to the zone. The zone data blocks may appear in any order following the parameter card. Every data deck must have a CAVITY, an INNER SHELL, and an OUTER SHELL data block. Also, NEUTRON SHIELD and WATER JACKET data blocks must appear in pairs. Five different data blocks may be used to describe the different calculations that may be performed on a model. (See Table H1.4.2.) These are denoted as Analytical Sequence data blocks, and they may appear in any order in any acceptable combination following the zone data blocks. A PRINTOUT TIMES data block may be entered to define the times at which the transient temperature distribution is to be printed by HEATING6. A MELTING RATIOS data card may be entered to generate a map depicting the nodal melting ratios at each printout time for transient calculations involving change of phase. A PLOT data card may be entered to generate temperature profile plots. A REGP data block may be entered to generate region plots. Each keyword associated with a data block must consist of at least the first four characters of the respective character string. Each keyword must be terminated by two or more blanks.

Table H1.4.2. Keywords for Analytical Sequences for HTAS1 Model.

INITIAL	- initial steady state
PREFIRE	- steady state immediately before fire
FIRE	- fire transient
POSTFIRE	- transient immediately following fire
FINAL	- final steady state

The final data card in a data deck for a case is an end-of-data (EOD) card and must contain a percent (%) in column 1 and a blank in column 2. This card signifies the end of data for a problem. As HTAS1 reads the input data, it sets a flag when an error is encountered and continues to read additional data unless the error was fatal. When the end-of-data indicator (%) is encountered, HTAS1 terminates reading and commences processing the input data. After processing is completed, the code performs the specified calculations unless an error was encountered, in which case it terminates the calculations.

H1.4.2 CARD 1 - TITLE OF PROBLEM

This card, which can contain alphanumeric characters in the first 56 columns, contains a descriptive title for the problem and must be the first card in the data deck for each case. The card itself cannot be omitted although it may be left blank. This title serves to identify the output for a case executed on HTAS1 since it appears with a number of tables and messages generated by the code. It is also used as part of the title for each thermal analysis performed by HEATING6. The appropriate entry from the following list is added to columns 57-72 of the HTAS1 title to generate the title for each thermal analysis being performed by HEATING6.

- INITIAL SS
- PREFIRE SS
- FIRE TR
- POSTFIRE TR
- FINAL SS

H1.4.3 CARD 2 - PROBLEM CONTROL PARAMETERS

H1.4.3.1 Initial Temperature Input Option (I)

The first entry, JTPIN, on this card specifies whether explicitly specified, lattice-point initial temperatures will be read by HEATING6 from an existing data set or file. If JTPIN is nonzero, HEATING6 will read the initial temperatures in unformatted form from logical unit number /JTPIN/. If there are no initial temperatures explicitly specified for HEATING6, this entry must be zero. The user must ensure that the appropriate data or file definition parameters have been supplied to describe the data set or file. See Sect. F10.3.7.5 for more details on using this option with HEATING6.

This entry must be nonzero for HTAS1 runs that begin with either the fire transient or the postfire transient. If it is zero for one of these cases, HTAS1 will generate an error message. It must also be nonzero for HTAS1 runs that restart a HEATING6 calculation. This entry is usually zero or blank for HTAS1 calculations that begin by performing the STEADY STATE, PREFIRE, or FINAL steady-state calculations.

H1.4.3.2 Final Temperature Output Option (I)

In addition to the normal HEATING6 output, the user may wish to have the final temperature distribution saved in some manner to facilitate the restarting of the problem. As an example, one may wish to perform a PREFIRE steady-state and FIRE transient analysis and save the temperature distribution at the end of the fire. Later, one can perform the POSTFIRE transient analysis by restarting from the saved temperature distribution.

If this option is desired, a nonzero integer must be entered as the second entry, JTPOUT, on this card. If JTPOUT is nonzero, then HEATING6 will write the final temperature distribution in unformatted form on a data

H1.4.9 ANALYTICAL SEQUENCE DATA BLOCKS

HTAS1 is designed to prepare HEATING6 input data to perform five different thermal analyses on a model of a radioactive material transport package. Each thermal analysis is identified by a keyword:

INITIAL - to obtain the steady-state normal operating thermal conditions for the transport package for operating conditions as specified in Part 71 of Title 10, Chapter 1, Code of Federal Regulations - Energy, United States Nuclear Regulatory Commission,

PREFIRE - to obtain the steady-state thermal conditions for the transport package prior to the fire transient (the only difference in INITIAL and PREFIRE is that due to the specified accident conditions of 10CFR71.73, the PREFIRE has a zero solar heat load as default),

FIRE - to obtain the transient thermal conditions during a specified accident in which the transport package is subjected to a fire at a stated temperature for a stated period of time,

POSTFIRE - to obtain the transient thermal conditions of the transport package for a specified period of time following exposure to the fire, and

FINAL - to obtain the steady-state operating thermal conditions following the accident and exposure to the fire.

By using acceptable combinations of the above thermal analyses, 15 different analytical sequences may be performed:

i) all five thermal analyses may be performed in a sequence or INITIAL, PREFIRE, FIRE, POSTFIRE, and FINAL,

ii) the INITIAL, PREFIRE, FIRE, and POSTFIRE thermal analyses may be performed in a sequence,

iii) the INITIAL, PREFIRE, and FIRE thermal analyses may be performed in a sequence,

iv) the INITIAL and PREFIRE thermal analyses may be performed in a sequence,

v) the PREFIRE, FIRE, POSTFIRE, and FINAL thermal analyses may be performed in a sequence,

vi) the PREFIRE, FIRE, and POSTFIRE thermal analyses may be performed in a sequence,

vii) the PREFIRE and FIRE thermal analyses may be performed in a sequence,

viii) the FIRE, POSTFIRE, and FINAL thermal analyses may be performed in a sequence,

ix) the FIRE and POSTFIRE thermal analyses may be performed in a sequence,

x) the POSTFIRE and FINAL thermal analyses may be performed in a sequence,

xi) the INITIAL thermal analysis may be performed separately,

xii) the PREFIRE thermal analysis may be performed separately,

xiii) the FIRE thermal analysis may be performed separately,

xiv) the POSTFIRE thermal analysis may be performed separately, and

xv) the FINAL thermal analysis may be performed separately.

The initial temperature distribution for any analytical sequence that begins with a FIRE thermal analysis must be taken from the final temperature distribution of a PREFIRE thermal analysis that was executed previously. Likewise, the initial temperature distribution for any analytical sequence that begins with a POSTFIRE thermal analysis must be taken from the final temperature distribution of a FIRE thermal analysis that was executed previously.

HTAS1 will define the appropriate calculations to HEATING6 based on the combination of keywords it encounters in the input data. If an unacceptable sequence of keywords is encountered, HTAS1 will write an error message and terminate calculations. Basically, all five Analytical Sequence Data Blocks are similar. Each is identified by the appropriate keyword followed by at least two blanks. Data unique to each analytical sequence may be entered after the two or more blanks following the keyword. If the input value of any data item on the INITIAL, PREFIRE, FIRE, POSTFIRE, or FINAL data cards is negative, then the value of the appropriate variable is set to zero and that option is not used for the model. If the input value of any data item is zero or blank, then the value of the appropriate variable is set to the default value. Otherwise, the input value of any data item is used for the value of the appropriate variable for the model. If a data card contains only one of the Analytical Sequence Data Block keywords, INITIAL, PREFIRE, FIRE, POSTFIRE, or FINAL, the value of each variable associated with the keyword is set to the default value. If none of the Analytical Sequence Data Blocks are present in the input data, then HTAS1 will generate a model for HEATING6 to perform the complete analytical sequence, INITIAL, PREFIRE, FIRE, POSTFIRE, and FINAL, using all of the default values. Any data card containing one of the Analytical Sequence Data Block keywords may be continued by beginning the continuation card with an "@" in column 1.

In order to override the default numerical technique for a particular thermal analysis, an additional data card or cards may be entered following the Analytical Sequence Data Block keyword card. For the INITIAL, PREFIRE, and FINAL thermal analyses, the additional card is the SS input data card for HEATING6 (Sect. F10.5.21) which contains the parameter values defining the numerical technique to be used to solve for the steady-state temperature distribution. For the FIRE and POSTFIRE thermal analysis a group of data cards must be entered. The first such data card must contain the keyword IMPLICIT or EXPLICIT to select either an implicit or explicit numerical technique. The IMPLICIT data card must be followed by two additional data cards or the IP and TP input data cards for HEATING6 (Sect. F10.5.22) which contain the parameter values defining the numerical technique to be used to solve for the transient temperature distribution. HEATING6 will not allow a transient implicit technique for models allowing materials with change of phase (i.e., when a LATENT HEAT card is present in the HTAS1 input data). The EXPLICIT data card must be followed by a data card containing the initial time step size and Levy's time step factor. The variables which may be defined in each of the Analytical Sequence Data Blocks are described below.

H1.4.9.1 INITIAL Steady-State Thermal Analysis

a) INITIAL Data Card

Each entry on the INITIAL data card is discussed below:

DBDTP - ambient temperature of the transport package, default 100°F,

DCONCR - coefficient for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal cylinders) to still air at a temperature of DBDTP (first data entry), default 0.18 Btu/(hr-ft^2-°F),

DCONER - exponent for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal cylinders) to still air at a temperature of DBDTP (first data entry), default 1/3,

DCONCA - coefficient for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (first data entry), default 0.19 Btu/(hr-ft^2-°F),

DCONEA - exponent for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (first data entry), default 1/3, and

DFLUX - solar heat load to be applied to all external surfaces of the transport package, default 122.92 Btu/(hr-ft^2).

b) Default Numerical Technique

As the default, HTAS1 will generate input data for HEATING6 to perform the INITIAL thermal analysis using the direct solution technique. For linear problems this technique requires only one pass through the numerical algorithm to obtain the steady-state temperature distribution. Nonlinear problems must be solved iteratively using the direct solution technique, and the HTAS1 default values allow a maximum of 20 iterations and a heat residual convergence criterion of 1.0×10^{-5}.

c) Overriding the Default Numerical Technique

If the user wishes to override the default values or if he wishes to specify that HEATING6 use the SOR technique to obtain the steady-state temperature distribution, a HEATING6 SS data card (Sect. F10.5.21) may be added to the HTAS1 input data immediately following the data card containing the keyword INITIAL. HTAS1 will use the values on the SS data card for the steady-state technique parameters instead of the default values.

The parameters that are defined on the HEATING6 SS data card are summarized below:

Maximum Number of Steady-State Iterations Allowed (I)

HEATING6 uses the SOR technique to solve the system of equations arising from a steady-state problem if this entry is 0 or a positive integer. If this entry is a negative integer, HEATING6 will use the direct solution technique to solve the system of equations arising from a one- or two-dimensional steady-state problem. In either case, this entry defines the maximum number of iterations allowed before the steady-state calculations are terminated. Normally 200 to 500 iterations are sufficient to converge to the solution using the SOR technique. For linear problems, the direct solution technique requires only one iteration to obtain the solution. Experience has indicated that 20 iterations are usually sufficient to obtain a converged solution for nonlinear steady-state problems using the direct solution technique. Thus, it is recommended to use -20 for this entry to indicate the direct solution technique. If this entry is zero, the code will use the SOR technique with a default of 500 iterations. If the maximum number of steady-state iterations is reached, and the convergence criterion is not satisfied, the HEATING6 program will write "END STEADY-STATE CALCULATIONS, CONVERGENCE NOT SATISFIED" on the printer and will terminate the calculations and attempt to read the next problem. In the event that the convergence criterion is not satisfied, one may wish to save the final temperature distribution in order to restart the problem as discussed in Sect. F10.5.5.4.

Steady-State Convergence Criterion (R)

For the SOR technique, this entry contains the value of ε in Eq. (F10.2.14) which defines the steady-state convergence criterion. For the direct solution technique, this entry contains the convergence criterion as discussed in Sect. F10.2.2.2. The steady-state calculation will continue until the convergence criterion is met. Since the criterion which ensures convergence varies from case to case, the user must rely on his own judgment and experience in determining the correct value for his particular problem. If this entry is zero, the code uses a default of 10^{-5}.

Steady-State Overrelaxation Factor (R)

The value of [see Eq. (F10.2.13)] is the third entry on this card and its value must be in the range $1 \leq \beta < 2$. If a zero is entered for steady-state problems using the SOR solution technique, the code assumes that the initial value of β is 1.9. This entry is not used for the direct solution technique.

Updating Temperature-Dependent Properties (I)

The fourth entry on this card specifies the number of iterations which are allowed before the temperature-dependent thermal properties are reevaluated for steady-state problems using the SOR solution technique. Once the convergence criterion has been satisfied, the code continues to iterate. However, the temperature-dependent thermal properties are now reevaluated after every iteration until the convergence criterion is satisfied a second time. Some nonlinear problems will converge in fewer iterations if the thermal properties are not evaluated at each iteration, and certainly, the computing time per iteration will be less. It is recommended that this parameter be on the order of 10 or 20. If it is zero, then the default value is unity for nonlinear problems. This entry is not used for the direct solution technique.

H1.4.9.2 PREFIRE Steady-State Thermal Analysis

a) PREFIRE Data Card

Each entry on the PREFIRE data card is discussed below:

DBDTP - ambient temperature of the transport package, default 100°F,

DCONCR - coefficient for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal cylinders) to still air at a temperature of DBDTP (first data entry), default 0.18 Btu/(hr-ft^2-°F),

DCONER - exponent for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal

cylinders) to still air at a temperature of DBDTP (first data entry), default 1/3,

DCONCA - coefficient for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (first data entry), default 0.19 Btu/(hr-ft^2-°F),

DCONEA - exponent for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (first data entry), default 1/3, and

DFLUX - solar heat load to be applied to all external surfaces of the transport package, default 0 Btu/(hr-ft^2).

b) Default Numerical Technique

As the default, HTAS1 will generate input data for HEATING6 to perform the PREFIRE steady-state thermal analysis using the direct solution technique. As with the INITIAL thermal analysis, the HTAS1 default values allow a maximum of 20 iterations and a heat residual convergence criterion of 1.0×10^{-5}.

c) Overriding the Default Numerical Technique

If the user wishes to override the default values or if he wishes to specify that HEATING6 use the SOR technique to obtain the PREFIRE steady-state temperature distribution, then a HEATING6 SS data card may be added to the HTAS1 input data immediately following the data card containing the keyword PREFIRE. HTAS1 will use the values on the SS data card for the steady-state technique parameters instead of the default values. See the section on the INITIAL thermal analysis for a description of the SS data card (Sect. H1.4.9.1b).

H1.4.9.3 FIRE Transient Thermal Analysis

a) FIRE Data Card

Each entry on the FIRE data card is discussed below:

DTIMEF - duration of the fire transient, default 30 min,

DEMISF - emissivity coefficient of the fire, default 0.9,

DBDTP - temperature of the fire, default 1475 °F,

DCONCR - coefficient for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal

DCONER - exponent for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal cylinders) to still air at a temperature of DBDTP (second data entry), default 1/3,

DCONCA - coefficient for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (second data entry), default 0.19 Btu/(hr-ft^2-°F),

DCONEA - exponent for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (second data entry), default

DFLUX - solar heat load to be applied to all external surfaces of the transport package, default 0 Btu/(hr-ft^2).

b) Default Numerical Technique

Normally, HTAS1 will generate input data for HEATING6 to perform the POSTFIRE thermal analysis using the implicit transient technique. As with the FIRE thermal analysis, HTAS1 default conditions will use an initial time step size of zero which causes HEATING6 to choose an initial time step size equal to the stability criterion as calculated for the explicit procedure. Other HTAS1 default values give a maximum time step size of 5 minutes and allow HEATING6 to adjust the time step size to keep the maximum percent of relative temperature change allowed at a node from one time level to the next one to 2%. HTAS1 allows HEATING6 to use its own default values for the remaining implicit transient technique parameters.

c) Overriding the Default Numerical Technique

If the user wishes to override the default option and choose the numerical technique to solve for the POSTFIRE transient temperature distribution, he may do so by supplying a group of data cards immediately following the data card containing the keyword POSTFIRE. The first card in the group must contain one of the keywords IMPLICIT or EXPLICIT to choose either an implicit or explicit numerical technique. HEATING6 IP and TP cards must immediately follow the data card containing the keyword IMPLICIT. Only one HEATING6 TP data card may be supplied. HTAS1 will use the values on the IP and TP cards for the implicit transient technique parameters instead of the default values.

See the FIRE thermal analysis section (Sect. H1.4.9.3) for a summary of the HEATING6 IP and TP data cards.

If the data card immediately following the POSTFIRE analytical sequence keyword contains the keyword EXPLICIT, an additional data card as

described in Sect. H1.4.9.3 must be supplied to define the EXPLICIT transient technique.

H1.4.9.5 FINAL Steady-State Thermal Analysis

a) FINAL Data Card

Each entry on the FINAL data card is discussed below:

DBDTP - ambient temperature of the transport package, default 100°F,

DCONCR - coefficient for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal cylinders) to still air at a temperature of DBDTP (first data entry), default 0.18 Btu/(hr-ft^2-°F),

DCONER - exponent for the term simulating the effects of natural convective heat transfer from the smooth radial surfaces of the transport package (assumed to be long, horizontal cylinders) to still air at a temperature of DBDTP (first data entry), default 1/3,

DCONCA - coefficient for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (first data entry), default 0.19 Btu/(hr-ft^2-°F),

DCONEA - exponent for the term simulating the effects of natural convective heat transfer from the smooth axial surfaces of the transport package (assumed to be vertical plates) to still air at a temperature of DBDTP (first data entry), default 1/3, and

DFLUX - solar heat load to be applied to all external surfaces of the transport package, default 0 Btu/(hr-ft^2).

b) Default Numerical Technique

As the default, HTAS1 will generate input data for HEATING6 to perform the FINAL steady-state thermal analysis using the direct solution technique. As with the INITIAL thermal analysis, the HTAS1 default values allow a maximum of 20 iterations and a heat residual convergence criterion of 1.0×10^{-5}.

c) Overriding the Default Numerical Technique

If the user wishes to override the default values or if he wishes to specify that HEATING6 use the SOR technique to obtain the FINAL steady-state temperature distribution, a HEATING6 SS data card may be added to the HTAS1 input data immediately following the data card containing the keyword

H1.4.11 FINNED SURFACE OPTION

One of HEATING6's eight fin effectiveness techniques may be applied along the outer surfaces of the model of the shipping container. If fins are defined on the outer radial surface of the outer shell and if the water jacket and neutron shield and/or one or both impact limiters are removed during the analytical sequence, the resulting uncovered surfaces of the outer shell are assumed to be smooth. However, the fins on the original uncovered portion of the outer shell surface are assumed to remain intact. This option is invoked along an outer surface or surfaces by entering a FINS data block in the data block for the zone or zones containing the particular surface or surfaces. The FINS data block may be entered in any order with other data blocks or cards defining the particular zone. A FINS data block may not be entered in the CAVITY, INNER SHELL, or SHIELDING data blocks. The format of the FINS data block for each zone follows:

KEYWORD - FINS

IFINR - Radial fin flag. This entry, a nonnegative integer, contains the HEATING6 fin type as defined in Fig. H1.2.6 that is to be applied along the outer radial surface of the zone. If this entry is zero or blank, then the fin effectiveness option is not invoked for the outer radial surfaces of the zone.

IFINZ - Axial fin flag. This entry, a nonnegative integer, contains the HEATING6 fin type as defined in Fig. H1.2.6 that is to be applied along the outer axial surface of the zone. If this entry is zero or blank, then the fin effectiveness option is not invoked for the outer axial surfaces of the zone.

HEATING6 B5 (Sect. F10.5.10.6) cards must follow the FINS data card to define the fin parameters for the radial and axial outer surfaces, respectively. The B5 card for the radial outer surface must precede the B5 card for the axial outer surface. If IFINR is zero or blank, the corresponding B5 card is omitted. Likewise, if IFINZ is zero or blank, the corresponding B5 card is omitted. The HEATING6 fin descriptor card, B5, containing six floating point numbers is described below:

a. FINLEN - height of the fin from the base surface in inches.

b. DELBAS - half-width of fin at base in inches (for rectangular spine, DELBAS is the half-width in the first direction).

c. DELTIP - half-width of fin at tip in inches (for rectangular spine, DELTIP is the half-width perpendicular to DELBAS).

d. S1 - fin spacing in inches (centerline to centerline).

e. S2 - spine spacing in inches (centerline to centerline) perpendicular to S1. Enter zero if not spine.

f. HCFAC - fin heat capacity multiplier (default 1.0 if HCFAC input as 0.0 or blank). If HCFAC < 0.0, then the fin heat capacity will not be added to the node's heat capacity [HCFAC will be 0.0 in Eq.(F10.2.62)].

For spines, the actual direction of DELBAS, DELTIP or S1, S2 is not important, but DELBAS must be perpendicular to DELTIP and S1 must be perpendicular to S2.

H1.4.12 PLOTTING DATA BLOCKS

Either region plots or temperature profile plots may be generated for the model that is defined by the HEATING6 input generated by HTAS1. If region plots are desired, the region plots data block must be the final data block listed in the HTAS1 input. If temperature profile plots are desired, the temperature profile plots data card may appear anywhere in the input after the title card and the problem control parameters card and before the end of data card. If both the temperature profile plots data card and the region plot data block are included in the input, only the region plots are generated because HEATING6 is not executed when region plots are generated. Therefore, no temperature distribution plot data sets exist from which temperature profile plots can be generated. The temperature profile plots data card must precede the region plots data block in this case.

H1.4.12.1 Temperature Profile Plots Data Card

KEYWORD - PLOT

MAXZVR - 0, generates a plot containing a curve that shows maximum temperature at any problem time vs the radial distance from the origin for all nodes on the axial center line (default).
1, above plot is not generated.

IVA - 0, generates a plot containing a curve that shows the difference in volume-averaged temperatures between the inner shell and the outer shell vs time (default).
1, above plot is not generated.

ISEP - 0, the temperature vs time curves for the centermost nodes of the inner shell, the shielding, and the outer shell are generated on the same plot (default).
1, each of the above curves is generated on an individual plot.

IDELT - 0, temperature vs time curves are generated for the actual temperatures (default).
1, temperature vs time curves are generated for temperature change from the initial temperature distribution.

TIMMIN - The problem time below which temperature distributions are not considered in the temperature vs time plots and the volume-averaged temperature plot. If TIMMIN is zero, the initial temperature distribution written by HEATING6 is the first one used

when generating these plots. The entry must be a non-negative floating-point number. The default is zero.

TIMMAX - The problem time above which temperature distributions are not considered in the temperature vs time plots and the volume-averaged temperature plot. If TIMMAX is zero, the final temperature distribution written by HEATING6 is the last one used when generating these plots. The entry must be a non-negative floating-point number. The default is zero.

If the only entry on this card is the keyword PLOT, then the three default temperature profile plots described in Sect. H1.2.6 are generated.

H1.4.12.2 Region Plots Data Block

The first card in the region plots data block contains only the keyword REGP. If no other cards are included in this data block, then by default plots of the regions, the materials, and the boundary condition function numbers are generated for each thermal analysis specified in the analytical sequence data blocks (Sect. H1.4.9).

Up to five plot description cards, one for each type of plot, may follow the keyword card if plots other than the default region plots are desired. The first entry on a plot description is an integer. All other entries are floating-point numbers. The entries are described below.

ITYPE - The number corresponding to the type of plot generated.

- =1 Plots are generated with region numbers inside the region boundaries for each calculational phase.
- =2 Plots are generated with material names inside the region boundaries for each calculational phase.
- =3 Plots are generated with heat generation function numbers inside the region boundaries for each calculational phase.
- =4 Plots are generated with initial temperature function numbers inside the region boundaries for each calculational phase.
- =5 Plots are generated with boundary condition function numbers inside the region boundaries for each calculational phase.

XSCALE - The scaling factor in units per inch for the horizontal axis of of region plots for every calculational phase. If XSCALE is zero, the scaling factor is computed by the program for each calculational phase. The default is zero.

YSCALE - The scaling factor in units per inch for the vertical axis of region plots for every calculational phase. If YSCALE is zero, the scaling factor is computed by the program for each calculational phase. The default is zero.

XREG1 - The minimum r value of the area for which region plots are generated for every calculational phase. If XREG1 is zero, the minimum r value of the model at each calculational phase is used. The default is zero.

XREG2 - The maximum r value of the area for which region plots are generated for every calculational phase. If XREG2 is zero, the maximum r value of the model at each calculational phase is used. The default is zero.

YREG1 - The minimum z value of the area for which region plots are generated for every calculational phase. If YREG1 is zero, the minimum z value of the model at each calculational phase is used. The default is zero.

YREG2 - The maximum z value of the area for which region plots are generated for every calculational phase. If YREG2 is zero, the maximum z value of the model at each calculational phase is used. The default is zero.

H1.5 SAMPLE PROBLEMS FOR HTAS1

This section contains two sample problems to demonstrate the use of some of the options available in HTAS1. The first sample problem makes extensive use of HTAS1 defaults to indicate the ease of performing a detailed thermal analysis of a shipping container using HTAS1. The second sample problem invokes a number of options available in HTAS1 to include the use of alternate methods of defining material thermal parameters.

The first sample problem is described below. The diameter of the cavity is 13.5 in. (radius, 6.75 in.) and its total height is 178 in. (half height, 89 in.). The total heat load within the cavity is 11.5 Kw (39247 Btu/hr). The radial thickness of the inner shell is 0.31 in., and its axial thickness at both top and bottom is 0.25 in. The shielding has a radial thickness of 6.63 in., and its axial extent is equal to that of the inner shell. The outer shell has a radial thickness of 1.25 in. The axial thickness of the top and bottom of the outer shell is 7.5 and 7.75 in., respectively. The diameter and total thickness of each impact limiter is 78 in. (radius 39 in.) and 38.5 in., respectively. The shipping container extends into each impact limiter a distance of 11 in. Thus, the axial thickness of each impact limiter is 27.5 in. and each impact limiter has a wraparound of 11 in. The radial thickness of the neutron shield and water jacket is 4.5 and 0.16 in., respectively. The neutron shield and water jacket extend axially to each impact limiter so their axial distance above and below the horizontal midplane is 85.5 and 86 in., respectively. The inner shell, outer shell, and water jacket are composed of 304 stainless steel. The shielding is lead, and the impact limiters are balsa wood. The neutron shield is water, and it is desired to model the natural convective heat transfer effects across the neutron shield with an effective thermal conductivity. Thus, one can use the HTAS1 default material associated with each zone. The emissivity associated with the outer surfaces of the water jacket and impact limiters is 0.5.

The shipping container is in a horizontal position and, for purposes of defining the steady-state conditions for the accident scenario of 10CFR71, the solar heat load is assumed to be zero. The loaded shipping container is allowed to reach steady-state operating conditions with an ambient temperature of 100°F. The shipping container is cooled by natural convection and radiation to the surrounding environment. It is then subjected to a fire at a temperature of 1475°F for 30 minutes. The emissivity of the fire is 0.9, and the absorptivity of the surfaces of the shipping container during the fire is 0.8. After the fire, the shipping container is allowed to cool by natural convection and radiation to the surrounding environment at an ambient temperature of 100 F. The emissivity on the surfaces of the shipping container remains at 0.8 following the fire.

It is desired to obtain the prefire steady-state operating temperature distribution, the transient temperature distribution during the fire, the transient temperature distribution for three hours following the fire, and the final steady state temperature distribution after the fire. The default numerical technique will be used to perform each set of

calculations. The input data necessary for HTAS1 to perform this analytical sequence and generate the default temperature profile plots for the first sample problem is shown in Table H1.5.1. The radial and axial grid divisions indicated for each zone are necessary to yield accurate solutions. Their values were determined by performing a series of analytical sequences using grid divisions that were both more coarse and more fine than the ones indicated in Table H1.5.1. This sample problem required 5 minutes 5 seconds of CPU time on the IBM 3033, 644K bytes of core, 2940 units of IO, and 13,800 lines of output.

The standard output from the initial entry to HTAS1 is presented in Table H1.5.2. The temperature distributions at the prefire steady state, at the end of the 30-minute fire transient, at the end of the three-hour cool down period, and at the final steady-state are presented in Tables H1.5.3, H1.5.4, H1.5.5, and H1.5.6, respectively. The HTAS1 output that describes the temperature profile plots generated for the first sample problem is presented in Table H1.5.7. The plots are presented in Fig. H1.5.1. A microfiche copy of the entire standard output from the analytical sequence for the first sample problem is included in Appendix H1.B.

The second sample problem is designed to demonstrate the use of some of the more useful options in HTAS1. It is the same as the first sample problem with the following changes. The inner shell is composed of carbon steel (1.5%) with thermal properties as presented in Table H1.5.8. The outer shell is composed of stainless steel 347. Its material properties are for material number 3135 from the LLNL material property library (ref. 2). The water jacket is composed of carbon steel (1%) with thermal properties from material number 14 in the HTAS1 material property library. The water in the neutron shield will be lost after the prefire steady state. The water will be replaced by a void so that heat will be transferred across the resulting gap by radiation only. The emissivity for each radial surface (outer surface of the outer shell and inner surface of the water jacket) of the resulting gap is 0.6 (the default emissivity on each surface is 0.5). The bottom impact limiter will be deleted following the fire transient. The emissivity on the outer surface of the outer shell not including the portion covered by the water jacket is 0.9 during the postfire transient and final steady state. This represents the portion of the outer surface previously covered by the bottom impact limiter. The default emissivity for the outer shell during this part of the calculations is 0.8. The ambient temperature during the prefire steady state, the postfire transient and the final steady state is 130 °F instead of the default of 100 °F. It is desired to compute the transient for the first four hours of the cool down period following the fire. The transient temperature distribution is needed every 5 minutes during the 30-minute fire, every 10 minutes for the first hour of the cool down period following the fire, and every 20 minutes for the remaining cool down period. After running several cases by varying the restrictions on the time step size, it was determined that an accurate solution could be obtained by adjusting the time step size while keeping the maximum relative change in temperature over a time step to 5%. The time step size should not exceed 5 minutes. The transient temperature distributions were made using the Crank-Nicolson implicit procedure subject to the above restrictions on the time step size.

Table H1.5.1. Input Data for First Sample Problem for HTAS1.

```
HTAS1 SAMPLE PROBLEM 1

CAVITY  6.75 5 89 5 39247
INNER SHELL  0.31 1 0.25 1
SHIELDING  6.63 5
OUTER SHELL  1.25 2 7.25 4 7.75 4
NEUTRON SHIELD  4.5 4 85.5 86
WATER JACKET  0.16 1
IMPACT LIMITER  39 5 27.5 5 0 0 11
PREFIRE
FIRE
POSTFIRE
FINAL
PLOT
%
```

Table H1.5.2. Standard Output from Initial Entry into HTAS1.

```
************************************************
* CONTROL MODULE HTAS1    HAS BEEN ACCESSED. *
************************************************



PRINT INPUT CARD IMAGES W/CARD COLUMNS INDICATED EVERY 10TH CARD--

CARD
NO./COL. 1.......10........20........30........40........50........60........70........80
    1    HTAS1 SAMPLE PROBLEM 1
    2
    3    CAVITY  6.75 5 89 5 39247
    4    INNER SHELL  0.31 1 0.25 1
    5    SHIELDING  6.63 5
    6    OUTER SHELL  1.25 2 7.25 4 7.75 4
    7    NEUTRON SHIELD  4.5 4 85.5 86
    8    WATER JACKET  0.16 1
    9    IMPACT LIMITER  39 5 27.5 5 0 0 11
   10    PREFIRE
NO./COL. 1.......10........20........30........40........50........60........70........80
   11    FIRE
   12    POSTFIRE
   13    FINAL
   14    PLOT
   15    %
```

Table H1.5.2. (cont'd)

```
CODE    : HTAS1          DATE: 10-05-84          COMPUTER: IBM3033/3033
VERSION: 02/12/83        TIME: 08.20.32          JOBNAME : CLIP1H1P

HTAS1: HEAT TRANSFER ANALYSIS SEQUENCE NUMBER 1 FOR THE SCALE SYSTEM.

HTAS1 IS A CONTROL MODULE FOR GENERATING INPUT DATA FOR HEATING6 FOR
TWO-DIMENSIONAL RZ MODELS ANALYZING A CLASS OF SHIPPING CONTAINERS.

HTAS1 WAS WRITTEN BY
     W.D. TURNER
     COMPUTER SCIENCES
     BUILDING K-1007, MAIL STOP NUMBER 53
     UNION CARBIDE CORPORATION, NUCLEAR DIVISION
     POST OFFICE BOX P
     OAK RIDGE, TENNESSEE 37830

     PHONE - 615-574-8677
             FTS 624-8677

TABULATION OF INPUT DATA

JOB DESCRIPTION -- HTAS1 SAMPLE PROBLEM 1

CAVITY PARAMETERS

     RADIUS                                  6.7500000D+00
     RADIAL DIVISIONS                                    5
     TOP HEIGHT                              8.9000000D+01
     AXIAL DIVISIONS FOR TOP                             5
     BOTTOM HEIGHT                           8.9000000D+01
     AXIAL DIVISIONS FOR BOTTOM                          5
     HEAT LOAD                               3.9247000D+04

INNER SHELL PARAMETERS

     RADIAL THICKNESS                        3.1000000D-01
     RADIAL DIVISIONS                                    1
     AXIAL THICKNESS AT TOP                  2.5000000D-01
     AXIAL DIVISIONS                                     1
     AXIAL THICKNESS AT BOTTOM               2.5000000D-01
     AXIAL DIVISIONS                                     1
     MATERIAL NUMBER                                    13
     MATERIAL NAME                                  SST304

        NUMBER     TEMPERATURE      CONDUCTIVITY
          1      3.2000000D+01    8.1000000D+00
          2      2.1200000D+02    8.7000000D+00
          3      3.9200000D+02    8.7000000D+00
          4      5.7200000D+02    9.4000000D+00
          5      7.5200000D+02    1.0000000D+01
          6      1.1120000D+03    1.1000000D+01
          7      1.4720000D+03    1.3000000D+01

        NUMBER     TEMPERATURE         DENSITY
                                    4.9100000D+02

        NUMBER     TEMPERATURE      SPECIFIC HEAT
          1      3.2000000D+01    1.2000000D-01
          2      7.5200000D+02    1.3500000D-01
```

Revised 12/84

Table H1.5.2. (cont'd)

SHIELDING PARAMETERS

RADIAL THICKNESS	6.6300000D+00
RADIAL DIVISIONS	5
AXIAL THICKNESS AT TOP	0.0
AXIAL DIVISIONS	0
AXIAL THICKNESS AT BOTTOM	0.0
AXIAL DIVISIONS	0
MATERIAL NUMBER	11
MATERIAL NAME	LEAD

NUMBER	TEMPERATURE	CONDUCTIVITY
1	3.2000000D+01	2.0300000D+01
2	2.1200000D+02	1.9300000D+01
3	3.9200000D+02	1.8200000D+01
4	5.7200000D+02	1.7200000D+01
5	6.3000000D+02	1.2100000D+01
6	7.1700000D+02	9.7000000D+00
7	8.0000000D+02	9.0000000D+00
8	9.8000000D+02	8.7000000D+00
9	1.2760000D+03	8.6600000D+00

NUMBER	TEMPERATURE	DENSITY
		7.1000000D+02

NUMBER	TEMPERATURE	SPECIFIC HEAT
1	3.2000000D+01	3.0500000D-02
2	2.1200000D+02	3.1500000D-02
3	6.2150000D+02	3.3800000D-02
4	6.2170000D+02	3.4000000D-02
5	1.8320000D+03	3.2800000D-02

OUTER SHELL PARAMETERS

RADIAL THICKNESS	1.2500000D+00
RADIAL DIVISIONS	2
AXIAL THICKNESS AT TOP	7.2500000D+00
AXIAL DIVISIONS	4
AXIAL THICKNESS AT BOTTOM	7.7500000D+00
AXIAL DIVISIONS	4
EMISSIVITY	
INITIAL STEADY STATE	5.0000000D-01
PREFIRE STEADY STATE	5.0000000D-01
FIRE TRANSIENT	8.0000000D-01
POSTFIRE TRANSIENT	8.0000000D-01
FINAL STEADY STATE	8.0000000D-01
MATERIAL NUMBER	13
MATERIAL NAME	SST304

NUMBER	TEMPERATURE	CONDUCTIVITY
1	3.2000000D+01	8.1000000D+00
2	2.1200000D+02	8.7000000D+00
3	3.9200000D+02	8.7000000D+00
4	5.7200000D+02	9.4000000D+00
5	7.5200000D+02	1.0000000D+01
6	1.1120000D+03	1.1000000D+01
7	1.4720000D+03	1.3000000D+01

NUMBER	TEMPERATURE	DENSITY
		4.9100000D+02

NUMBER	TEMPERATURE	SPECIFIC HEAT
1	3.2000000D+01	1.2000000D-01
2	7.5200000D+02	1.3500000D-01

Table H1.5.2. (cont'd)

Revised 12/84

NEUTRON SHIELD PARAMETERS

RADIAL THICKNESS	4.5000000D+00
RADIAL DIVISIONS	4
TOP HEIGHT	8.5500000D+01
BOTTOM HEIGHT	8.6000000D+01
EMISSIVITY	
INITIAL STEADY STATE	5.0000000D-01
PREFIRE STEADY STATE	5.0000000D-01
FIRE TRANSIENT	8.0000000D-01
POSTFIRE TRANSIENT	8.0000000D-01
FINAL STEADY STATE	8.0000000D-01
MATERIAL NUMBER	17
MATERIAL NAME	H2OCONV

NUMBER	TEMPERATURE	CONDUCTIVITY
1	1.5000000D+02	1.4750000D+01
2	2.0000000D+02	1.7330000D+01
3	2.5000000D+02	1.9150000D+01
4	3.0000000D+02	2.0830000D+01
5	4.0000000D+02	2.3350000D+01

NUMBER	TEMPERATURE	DENSITY
		6.0000000D+01

NUMBER	TEMPERATURE	SPECIFIC HEAT
1	3.2000000D+01	1.0072000D+00
2	6.8000000D+01	9.9861000D-01
3	1.0400000D+02	9.9780000D-01
4	1.4000000D+02	9.9921000D-01
5	1.7600000D+02	1.0021000D+00
6	2.1200000D+02	1.0068000D+00
7	2.4800000D+02	1.0149000D+00
8	2.8400000D+02	1.0228000D+00
9	3.2000000D+02	1.0369000D+00
10	3.5600000D+02	1.0548000D+00
11	3.9200000D+02	1.0758000D+00
12	4.2800000D+02	1.1009000D+00
13	4.6400000D+02	1.1357000D+00
14	5.0000000D+02	1.1818000D+00
15	5.3710000D+02	1.2437000D+00
16	5.7200000D+02	1.3678000D+00

Revised 12/84

Table H1.5.2. (cont'd)

WATER JACKET PARAMETERS

RADIAL THICKNESS	1.6000000D-01
RADIAL DIVISIONS	1
TOP HEIGHT	8.5500000D+01
BOTTOM HEIGHT	8.6000000D+01
EMISSIVITY	
INITIAL STEADY STATE	5.0000000D-01
PREFIRE STEADY STATE	5.0000000D-01
FIRE TRANSIENT	8.0000000D-01
POSTFIRE TRANSIENT	8.0000000D-01
FINAL STEADY STATE	8.0000000D-01
MATERIAL NUMBER	13
MATERIAL NAME	SST304

NUMBER	TEMPERATURE	CONDUCTIVITY
1	3.2000000D+01	8.1000000D+00
2	2.1200000D+02	8.7000000D+00
3	3.9200000D+02	8.7000000D+00
4	5.7200000D+02	9.4000000D+00
5	7.5200000D+02	1.0000000D+01
6	1.1120000D+03	1.1000000D+01
7	1.4720000D+03	1.3000000D+01

NUMBER	TEMPERATURE	DENSITY
		4.9100000D+02

NUMBER	TEMPERATURE	SPECIFIC HEAT
1	3.2000000D+01	1.2000000D-01
2	7.5200000D+02	1.3500000D-01

IMPACT LIMITER

RADIUS	3.9000000D+01
RADIAL DIMENSIONS FROM 1.9600000D+01	5
AXIAL THICKNESS OF TOP IMPACT LIMITER	2.7500000D+01
AXIAL DIVISIONS	5
AXIAL THICKNESS OF TOP IMPACT LIMITER ALONG OUTER SURFACE OF CONTAINER	1.1000000D+01
AXIAL THICKNESS OF BOTTOM IMPACT LIMITER	2.7500000D+01
AXIAL DIVISIONS	5
AXIAL THICKNESS OF BOTTOM IMPACT LIMITER ALONG OUTER SURFACE OF CONTAINER	1.1000000D+01
EMISSIVITY	
INITIAL STEADY STATE	5.0000000D-01
PREFIRE STEADY STATE	5.0000000D-01
FIRE TRANSIENT	8.0000000D-01
POSTFIRE TRANSIENT	8.0000000D-01
FINAL STEADY STATE	8.0000000D-01
MATERIAL NUMBER	19
MATERIAL NAME	BALSA

NUMBER	TEMPERATURE	CONDUCTIVITY
		4.8400000D-02

NUMBER	TEMPERATURE	DENSITY
		2.0000000D+01

NUMBER	TEMPERATURE	SPECIFIC HEAT
		5.5000000D-01

Revised 12/84

Table H1.5.2. (cont'd)

```
CODE    : HTAS1          DATE: 10-05-84          COMPUTER: IBM3033/3033
VERSION: 02/12/83        TIME: 08.20.32          JOBNAME : CLIP1H1P

JOB DESCRIPTION -- HTAS1 SAMPLE PROBLEM 1

*********************************************************************************************
HEATING6 INPUT DATA WILL BE GENERATED FOR THE PREFIRE STEADY STATE THERMAL ANALYSIS.
*********************************************************************************************

THE STEADY STATE TEMPERATURE DISTRIBUTION FOR THE SHIPPING CONTAINER MODEL
WILL BE OBTAINED SUBJECT TO THE FOLLOWING BOUNDARY CONDITIONS:

     AMBIENT TEMPERATURE                                1.0000000D+02
     CONVECTION AT RADIAL BOUNDARY
          CONVECTIVE COEFFICIENT                        1.8000000D-01
          CONVECTIVE EXPONENT                           3.3333333D-01
     CONVECTION AT AXIAL BOUNDARY
          CONVECTIVE COEFFICIENT                        1.9000000D-01
          CONVECTIVE EXPONENT                           3.3333333D-01
     RADIATION TO SURROUNDING MEDIUM AT AMBIENT
          TEMPERATURE WITH EMISSIVITY COEFFICIENTS
          AS SPECIFIED ALONG EACH SURFACE
     SOLAR HEAT FLUX                                    0.0

THE STEADY STATE TEMPERATURE DISTRIBUTION WILL BE
SOLVED USING THE DIRECT SOLUTION TECHNIQUE.
     MAXIMUM NUMBER OF ITERATIONS WILL BE                      20
     CONVERGENCE CRITERION WILL BE                      1.0000000D-05
```

Table H1.5.2. (cont'd)

```
CODE    : HTAS1          DATE: 10-05-84          COMPUTER: IBM3033/3033
VERSION: 02/12/83        TIME: 08.20.32          JOBNAME : CLIP1H1P

JOB DESCRIPTION -- HTAS1 SAMPLE PROBLEM 1

********************************************************************************
HEATING6 INPUT DATA WILL BE GENERATED FOR THE FIRE TRANSIENT THERMAL ANALYSIS.
********************************************************************************

THE TRANSIENT TEMPERATURE DISTRIBUTION FOR THE SHIPPING CONTAINER MODEL WILL BE
OBTAINED DURING EXPOSURE TO A FIRE SUBJECT TO THE FOLLOWING BOUNDARY CONDITIONS:

     DURATION OF FIRE                                  3.0000000D+01
     FIRE TEMPERATURE                                  1.4750000D+03
     CONVECTION AT RADIAL BOUNDARY
          CONVECTIVE COEFFICIENT                       0.0
          CONVECTIVE EXPONENT                          0.0
     CONVECTION AT AXIAL BOUNDARY
          CONVECTIVE COEFFICIENT                       0.0
          CONVECTIVE EXPONENT                          0.0
     RADIATION FROM FIRE
          FIRE EMISSIVITY COEFFICIENT                        0.9000
          SURFACE ABSORPTION COEFFICIENTS ARE
          AS SPECIFIED FOR THE MATERIALS
          ALONG EACH SURFACE.
     SOLAR HEAT FLUX                                   0.0

THE IMPLICIT PROCEDURE WILL BE USED TO CALCULATE THE TRANSIENT TEMPERATURE DISTRIBUTION.
MAXIMUM NORMALIZED HEAT RESIDUAL CONVERGENCE CRITERION = 0.0
(CORRESPONDS TO EPSILON SUB 1, DEFAULT=1.0D-5)
NUMBER-OF-ITERATIONS BETWEEN TESTS FOR CONVERGENCE, LINEAR LOOP (DEFAULT = 1) =  0
AVERAGE L1 NORM OF RELATIVE TEMPERATURE DIFFERENCE CONVERGENCE CRITERION
FOR TEMPERATURE DEPENDENT PROPERTIES = 0.0
(CORRESPONDS TO EPSILON SUB 3, DEFAULT=1.0D-5)
THETA (0.5 FOR CRANK-NICOLSON,1.0 FOR CLASSICAL IMPLICIT) = 0.0
(DEFAULT = 0.5)
THE SOR ACCELERATION PARAMETER (BETA) WILL BE OPTIMIZED EMPIRICALLY.
BETA = 0.0
A BETA UPDATE WILL BE ATTEMPTED EVERY    0 TIME STEPS (DEFAULT=1).
NUMBER-OF-ITERATIONS TOLERANCE FOR BETA UPDATE CALCULATIONS, OUTER LOOP (DEFAULT=5) =    0
NUMBER-OF-ITERATIONS TOLERANCE FOR BETA UPDATE CALCULATIONS, INNER LOOP (DEFAULT=2) =    0

THE FOLLOWING TIME STEP PARAMETERS WILL BE USED TO COMPUTE THE FIRE TRANSIENT.
     TIME STEP  0.0
     THE MAXIMUM TIME STEP ALLOWED IS  5.0000000D+00
     NEW TIME STEP PARAMETERS WILL BE READ IN AT TIME =  3.0000000D+01
     IF THE MAXIMUM TEMPERATURE CHANGE AT A NODE EXCEEDS  2.0000000D+00 PER CENT OVER A
     TIME STEP, THEN THE TIME STEP SIZE WILL BE DECREASED ACCORDINGLY.
```

Table H1.5.2. (cont'd)

```
CODE    : HTAS1          DATE: 10-05-84          COMPUTER: IBM3033/3033
VERSION: 02/12/83        TIME: 08.20.32          JOBNAME : CLIP1H1P

JOB DESCRIPTION -- HTAS1 SAMPLE PROBLEM 1

**********************************************************************************
HEATING6 INPUT DATA WILL BE GENERATED FOR THE POSTFIRE TRANSIENT THERMAL ANALYSIS.
**********************************************************************************

THE TRANSIENT TEMPERATURE DISTRIBUTION FOR THE SHIPPING CONTAINER MODEL WILL BE
OBTAINED FOLLOWING EXPOSURE TO A FIRE SUBJECT TO THE FOLLOWING BOUNDARY CONDITIONS:

     DURATION OF POST-FIRE TRANSIENT                  1.8000000D+02
     AMBIENT TEMPERATURE                              1.0000000D+02
     CONVECTION AT RADIAL BOUNDARY
          CONVECTIVE COEFFICIENT                      1.8000000D-01
          CONVECTIVE EXPONENT                         3.3333333D-01
     CONVECTION AT AXIAL BOUNDARY
          CONVECTIVE COEFFICIENT                      1.9000000D-01
          CONVECTIVE EXPONENT                         3.3333333D-01
     RADIATION TO SURROUNDING MEDIUM AT AMBIENT
          TEMPERATURE WITH EMISSIVITY COEFFICIENTS
          AS SPECIFIED ALONG EACH SURFACE
     SOLAR HEAT FLUX                                  0.0

THE IMPLICIT PROCEDURE WILL BE USED TO CALCULATE THE TRANSIENT TEMPERATURE DISTRIBUTION.
MAXIMUM NORMALIZED HEAT RESIDUAL CONVERGENCE CRITERION = 0.0
(CORRESPONDS TO EPSILON SUB 1, DEFAULT=1.0D-5)
NUMBER-OF-ITERATIONS BETWEEN TESTS FOR CONVERGENCE, LINEAR LOOP (DEFAULT = 1) =  0
AVERAGE L1 NORM OF RELATIVE TEMPERATURE DIFFERENCE CONVERGENCE CRITERION
FOR TEMPERATURE DEPENDENT PROPERTIES = 0.0
(CORRESPONDS TO EPSILON SUB 3, DEFAULT=1.0D-5)
THETA (0.5 FOR CRANK-NICOLSON,1.0 FOR CLASSICAL IMPLICIT) = 0.0
(DEFAULT = 0.5)
THE SOR ACCELERATION PARAMETER (BETA) WILL BE OPTIMIZED EMPIRICALLY.
BETA = 0.0
A BETA UPDATE WILL BE ATTEMPTED EVERY    0 TIME STEPS (DEFAULT=1).
NUMBER-OF-ITERATIONS TOLERANCE FOR BETA UPDATE CALCULATIONS, OUTER LOOP (DEFAULT=5) =    0
NUMBER-OF-ITERATIONS TOLERANCE FOR BETA UPDATE CALCULATIONS, INNER LOOP (DEFAULT=2) =    0

THE FOLLOWING TIME STEP PARAMETERS WILL BE USED TO COMPUTE THE POST-FIRE TRANSIENT.
     TIME STEP  0.0
     THE MAXIMUM TIME STEP ALLOWED IS  5.0000000D+00
     NEW TIME STEP PARAMETERS WILL BE READ IN AT TIME =  2.1000000D+02
     IF THE MAXIMUM TEMPERATURE CHANGE AT A NODE EXCEEDS  2.0000000D+00 PER CENT OVER A
     TIME STEP, THEN THE TIME STEP SIZE WILL BE DECREASED ACCORDINGLY.
```

Table H1.5.2. (cont'd)

```
CODE    : HTAS1          DATE: 10-05-84          COMPUTER: IBM3033/3033
VERSION: 02/12/83        TIME: 08.20.32          JOBNAME : CLIP1H1P

JOB DESCRIPTION -- HTAS1 SAMPLE PROBLEM 1

*******************************************************************************
HEATING6 INPUT DATA WILL BE GENERATED FOR THE FINAL STEADY STATE THERMAL ANALYSIS.
*******************************************************************************

THE STEADY STATE TEMPERATURE DISTRIBUTION FOR THE SHIPPING CONTAINER MODEL
WILL BE OBTAINED SUBJECT TO THE FOLLOWING BOUNDARY CONDITIONS:

    AMBIENT TEMPERATURE                                  1.0000000D+02
    CONVECTION AT RADIAL BOUNDARY
         CONVECTIVE COEFFICIENT                          1.8000000D-01
         CONVECTIVE EXPONENT                             3.3333333D-01
    CONVECTION AT AXIAL BOUNDARY
         CONVECTIVE COEFFICIENT                          1.9000000D-01
         CONVECTIVE EXPONENT                             3.3333333D-01
    RADIATION TO SURROUNDING MEDIUM AT AMBIENT
         TEMPERATURE WITH EMISSIVITY COEFFICIENTS
         AS SPECIFIED ALONG EACH SURFACE
    SOLAR HEAT FLUX                                      0.0

THE STEADY STATE TEMPERATURE DISTRIBUTION WILL BE
SOLVED USING THE DIRECT SOLUTION TECHNIQUE.
     MAXIMUM NUMBER OF ITERATIONS WILL BE                      20
     CONVERGENCE CRITERION WILL BE                    1.0000000D-05
```

Table H1.5.2. (cont'd)

```
CODE    : HTAS1          DATE: 10-05-84          COMPUTER: IBM3033/3033
VERSION: 02/12/83        TIME: 08.20.32          JOBNAME : CLIP1H1P

JOB DESCRIPTION -- HTAS1 SAMPLE PROBLEM 1

TRANSIENT PRINTOUT TIMES
     OUTPUT        OUTPUT
     NUMBER         TIME
        1       5.00000D+00
        2       1.00000D+01
        3       1.50000D+01
        4       2.00000D+01
        5       2.50000D+01
        6       3.00000D+01
        7       3.50000D+01
        8       4.00000D+01
        9       4.50000D+01
       10       5.00000D+01
       11       5.50000D+01
       12       6.00000D+01
       13       6.50000D+01
       14       7.00000D+01
       15       7.50000D+01
       16       8.00000D+01
       17       8.50000D+01
       18       9.00000D+01
       19       9.50000D+01
       20       1.00000D+02
       21       1.05000D+02
       22       1.10000D+02
       23       1.15000D+02
       24       1.20000D+02
       25       1.25000D+02
       26       1.30000D+02
       27       1.35000D+02
       28       1.40000D+02
       29       1.45000D+02
       30       1.50000D+02
       31       1.55000D+02
       32       1.60000D+02
       33       1.65000D+02
       34       1.70000D+02
       35       1.75000D+02
       36       1.80000D+02
       37       1.85000D+02
       38       1.90000D+02
       39       1.95000D+02
       40       2.00000D+02
       41       2.05000D+02
       42       2.10000D+02

IN HTAS1, LIMIT (DIMENSIONS FOR THE D-ARRAY) IS    142848 WHICH USES  559K BYTES OF CORE.
(LIMIT IS COMPUTED BY ALOCAT AND IS A FUNCTION OF THE COMPUTER CORE REQUEST.)
THE VARIABLY DIMENSIONED ARRAYS REQUIRE     14880 ELEMENTS OR   59K BYTES OF CORE.
THERE ARE       500K BYTES OF CORE LEFT FOR STORAGE OF HEATING6 INPUT DATA IN BINARY FORM.

HTAS1 USED A TOTAL OF     13865 ELEMENTS IN THE D-ARRAY WHICH REQUIRED A TOTAL
OF   55K BYTES OF CORE.  THE CORE REQUIREMENTS FOR THE HTAS1 PORTION OF THE
ANALYSIS COULD BE REDUCED BY  502K BYTES.

*********************************************************************************
* EXIT FROM CONTROL MODULE HTAS1    AND DIRECT SCALE TO EXECUTE HEATING6 (Q#$010  ). *
*********************************************************************************
```

Table H1.5.3. Temperature Distribution at the Prefire Steady State.

HEATING6 02/12/83 HTAS1 SAMPLE PROBLEM 1 -- PREFIRE SS IBM3033/3033
CLIP1H1P 08.21.57 10-05-8

STEADY STATE TEMPERATURE DISTRIBUTION AFTER 14 ITERATIONS, TIME = 0.0

GROSS GRID			1					2	3				4	
	FINE GRID		1	2	3	4	5	6	7	8	9	10	11	12	13
		DISTANCE	-1.35	1.35	2.70	4.05	5.40	6.75	7.06	8.39	9.71	11.04	12.36	13.69	14.3
1	1	0.0	0I0	102.43	102.42	102.40	102.36	102.32	102.31	102.25	102.19	102.12	102.05	101.96	101.9
	2	5.50	0I0	122.56	122.42	122.17	121.83	121.40	121.29	120.76	120.14	119.45	118.68	117.84	117.4
	3	11.00	0I0	145.92	145.63	145.14	144.46	143.58	143.35	142.26	140.99	139.54	137.93	136.16	135.2
	4	16.50	0I0	175.78	175.33	174.58	173.53	172.17	171.81	170.09	168.05	165.67	162.96	159.90	158.3
	5	22.00	0I0	215.52	214.97	214.05	212.77	211.11	210.67	208.57	206.02	202.94	199.23	194.68	192.1
2	6	27.50	0I0	267.45	266.76	265.69	264.32	262.80	262.45	260.95	259.56	258.35	257.37	256.61	256.3
	7	29.44	0I0	268.51	267.73	266.51	264.96	263.24	262.85	261.18	259.66	258.36	257.33	256.62	256.4
	8	31.38	0I0	271.69	270.65	268.97	266.80	264.38	263.85	261.65	259.71	258.08	256.80	255.93	255.7
	9	33.31	0I0	277.74	276.26	273.77	270.33	266.26	265.41	262.24	259.62	257.44	255.65	254.29	253.8
3	10	35.25	0I0	287.86	285.97	282.57	277.12	268.48	266I73	262.61	259.31	256.46	253.92	251I55	249.9
4	11	35.50	0I0	289.55	287.65	284.24	278.74	268I76	266I55	262.62	259.29	256.40	253.82	251I46	249.1
	12	53.30	0I0	0.0	0.0	0.0	0.0	265I95	263I85	260.18	257.04	254.30	251.86	249I66	247.5
	13	71.10	0I0	0.0	0.0	0.0	0.0	265I16	263I07	259.40	256.27	253.54	251.12	248I95	246.8
	14	88.90	0I0	0.0	0.0	0.0	0.0	264I78	262I69	259.02	255.89	253.17	250.75	248I58	246.5
	15	106.70	0I0	0.0	0.0	0.0	0.0	264I60	262I51	258.84	255.72	252.99	250.58	248I41	246.3
5	16	124.50	0I0	0.0	0.0	0.0	0.0	264I55	262I46	258.79	255.67	252.94	250.53	248I36	246.2
	17	142.30	0I0	0.0	0.0	0.0	0.0	264I61	262I51	258.85	255.72	253.00	250.58	248I41	246.3
	18	160.10	0I0	0.0	0.0	0.0	0.0	264I79	262I70	259.03	255.90	253.18	250.76	248I59	246.5
	19	177.90	0I0	0.0	0.0	0.0	0.0	265I17	263I08	259.41	256.28	253.55	251.13	248I96	246.8
	20	195.70	0I0	0.0	0.0	0.0	0.0	265I97	263I88	260.21	257.07	254.32	251.88	249I69	247.5
6	21	213.50	0I0	289.98	288.06	284.61	279.03	268I84	266I63	262.69	259.36	256.46	253.87	251I50	249.1
7	22	213.75	0I0	288.30	286.38	282.95	277.41	268.58	266I81	262.69	259.38	256.52	253.97	251I60	250.0
	23	215.56	0I0	278.71	277.16	274.56	270.93	266.60	265.71	262.41	259.70	257.44	255.58	254.16	253.7
	24	217.37	0I0	272.84	271.70	269.88	267.51	264.86	264.28	261.89	259.81	258.07	256.69	255.75	255.5
	25	219.19	0I0	269.69	268.82	267.45	265.71	263.77	263.33	261.48	259.79	258.35	257.22	256.44	256.2
8	26	221.00	0I0	268.64	267.86	266.63	265.08	263.35	262.95	261.26	259.71	258.36	257.27	256.46	256.1
	27	226.50	0I0	215.85	215.28	214.32	212.98	211.27	210.81	208.64	206.04	202.91	199.14	194.55	192.0
	28	232.00	0I0	175.84	175.39	174.63	173.56	172.18	171.81	170.07	168.00	165.60	162.86	159.78	158.2
	29	237.50	0I0	145.91	145.61	145.12	144.43	143.54	143.31	142.21	140.93	139.47	137.85	136.07	135.1
	30	243.00	0I0	122.54	122.39	122.15	121.81	121.37	121.26	120.72	120.10	119.40	118.63	117.79	117.3
9	31	248.50	0I0	102.43	102.42	102.39	102.36	102.32	102.30	102.25	102.19	102.12	102.94	101.96	101.9

Table H1.5.3. (cont'd)

HEATING6 02/12/83 HTAS1 SAMPLE PROBLEM 1 -- PREFIRE_SS IBM3033/3033
CLIP1HIP 08.21.57 10-05-8

STEADY STATE TEMPERATURE DISTRIBUTION AFTER 14 ITERATIONS, TIME = 0.0

	GROSS GRID		5				6	7					8
	FINE GRID		14	15	16	17	18	19	20	21	22	23	24
		DISTANCE	14.94	16.06	17.19	18.31	19.44	19.60	23.48	27.36	31.24	35.12	39.00
1	1	0.0	101.88	101.80	101.71	101.63	101.54	101.52	101.20	100.89	100.59	100.31	100105
	2	5.50	117.00	116.21	115.38	114.54	113.67	113.55	110.49	107.56	104.90	102.52	100141
	3	11.00	134.37	132.67	130.91	129.09	127.25	126.98	120.51	114.51	109.25	104.70	100173
	4	16.50	156.71	153.62	150.38	147.05	143.67	143.19	131.64	121.61	113.37	106.66	101101
	5	22.00	189.36	183.54	177.24	170.82	164.47	163.59	143.31	127.70	116.29	107.86	101118
2	6	27.50	256105	234.69	217.18	202.22	189.04	187.27	151.55	129.18	115.72	107.20	101108
	7	29.44	256131	238.62	222.69	208.12	194.49	192.60	150.36	126.65	113.82	106.21	100194
	8	31.38	255159	240.84	227.13	213.83	200.31	198.33	145.91	122.06	110.92	104.82	100173
	9	33.31	253169	242.92	232.96	222.65	210.56	208.53	135.91	114.91	107.03	103.06	100147
3	10	35.25	248189	244.78	241.80	238.79	234.30	232.85	114.97	105.00	102.36	101.05	100117
4	11	35.50	246181	244.83	242.98	241.25	239162	239114	110.50	103.61	101.73	100.79	100113
	12	53.30	245153	243.95	242.47	241.09	239178	239139	0.0	0.0	0.0	0.0	010
	13	71.10	244186	243.29	241.83	240.46	239116	238178	0.0	0.0	0.0	0.0	010
	14	88.90	244150	242.94	241.48	240.11	238182	238143	0.0	0.0	0.0	0.0	010
	15	106.70	244133	242.77	241.31	239.94	238165	238127	0.0	0.0	0.0	0.0	010
5	16	124.50	244128	242.72	241.27	239.90	238161	238122	0.0	0.0	0.0	0.0	010
	17	142.30	244133	242.78	241.32	239.95	238166	238127	0.0	0.0	0.0	0.0	010
	18	160.10	244151	242.95	241.49	240.12	238182	238144	0.0	0.0	0.0	0.0	010
	19	177.90	244187	243.31	241.84	240.47	239117	238179	0.0	0.0	0.0	0.0	010
	20	195.70	245155	243.97	242.50	241.11	239180	239142	0.0	0.0	0.0	0.0	010
6	21	213.50	246185	244.86	243.01	241.28	239165	239117	110.39	103.57	101.71	100.77	100113
7	22	213.75	248189	244.79	241.79	238.79	234.35	232.91	114.81	104.95	102.32	101.03	100117
	23	215.56	253151	242.95	233.22	223.14	211.20	209.17	134.70	114.16	106.64	102.88	100145
	24	217.37	255138	240.79	227.26	214.09	200.57	198.58	144.53	120.97	110.28	104.51	100169
	25	219.19	256111	238.46	222.60	208.06	194.38	192.48	149.11	125.50	113.07	105.84	100188
8	26	221.00	255186	234.54	217.06	202.09	188.84	187.06	150.59	128.16	114.99	106.82	101102
	27	226.50	189.20	183.34	177.02	170.57	164.18	163.29	142.83	127.20	115.90	107.64	101115
	28	232.00	156.57	153.46	150.20	146.85	143.46	142.98	131.38	121.35	113.16	106.54	101100
	29	237.50	134.27	132.56	130.79	128.97	127.12	126.85	120.37	114.38	109.14	104.64	100172
	30	243.00	116.95	116.15	115.32	114.48	113.61	113.49	110.43	107.50	104.85	102.49	100140
9	31	248.50	101.87	101.79	101.71	101.62	101.53	101.52	101.20	100.88	100.59	100.31	100105

Table H1.5.3. (cont'd)

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	0.0
2	100.000000
3	100.000000
4	100.000000
5	100.000000
6	100.000000
7	100.000000
8	100.000000

ELAPSED CPU TIME IS 8.95 SECONDS

THE MAXIMUM TEMPERATURE IS - 2.89979D+02 (+-0.1 PERCENT)

MAX. TEMP. APPEARS AT NODES - 380

THE MINIMUM TEMPERATURE IS - 1.00054D+02 (+-0.1 PERCENT)

MIN. TEMP. APPEARS AT NODES - 23 253 402 632

Table H1.5.4. Temperature Distribution at the End of the 30-Minute Fire Transient.

HEATING6 02/12/83 HTASI SAMPLE PROBLEM 1 -- FIRE TR IBM3033/3033
CLIP1H1P 08.25.57 10-05-8

TRANSIENT TEMPERATURE DISTRIBUTION AFTER 246 TIME STEPS, TIME = 3.00000D+01

GROSS GRID			1 I				2 I	3 I					4 I	
FINE GRID			1	2	3	4	5	6	7	8	9	10	11	12	13
DISTANCE			-1.35	1.35	2.70	4.05	5.40	6.75	7.06	8.39	9.71	11.04	12.36	13.69	14.3
1	1	0.0	0I0	1465.67	1465.67	1465.67	1465.67	1465.67	1465.67	1465.67	1465.66	1465.66	1465.65	1465.65	1465.6
	2	5.50	0I0	133.26	133.11	132.87	132.53	132.10	131.99	131.45	130.84	130.14	129.37	128.54	128.1
	3	11.00	0I0	145.97	145.67	145.19	144.50	143.63	143.40	142.31	141.04	139.59	137.97	136.21	135.3
	4	16.50	0I0	175.78	175.33	174.58	173.53	172.17	171.81	170.09	168.05	165.67	162.96	159.90	158.3
	5	22.00	0I0	215.52	214.97	214.05	212.77	211.11	210.68	208.57	206.02	202.95	199.23	194.69	192.1
2	6	27.50	0I0	267.58	266.97	266.02	264.87	263.64	263.37	262.24	261.30	260.59	260.07	259.57	259.2
	7	29.44	0I0	268.82	268.22	267.32	266.28	265.25	265.03	264.18	263.63	263.42	263.46	263.51	263.4
	8	31.38	0I0	272.50	271.95	271.25	270.68	270.45	270.43	270.51	271.20	272.68	274.72	276.63	277.0
	9	33.31	0I0	279.45	279.18	279.30	280.63	283.73	284.37	286.72	290.39	296.07	303.64	311.93	314.8
3	10	35.25	0I0	290.43	290.69	292.49	298.88	316.43	319122	323.92	332.48	346.18	365.51	391109	411.8
4	11	35.50	0I0	292.12	292.40	294.21	300.77	324150	323183	327.27	336.12	350.40	370.58	397153	431.2
	12	53.30	0I0	0.0	0.0	0.0	0.0	346145	345121	347.17	354.08	365.58	381.54	402100	427.7
	13	71.10	0I0	0.0	0.0	0.0	0.0	346104	344180	346.74	353.56	364.91	380.69	400192	426.4
	14	88.90	0I0	0.0	0.0	0.0	0.0	345168	344143	346.38	353.20	364.55	380.33	400157	426.1
	15	106.70	0I0	0.0	0.0	0.0	0.0	345151	344127	346.21	353.03	364.39	380.17	400141	425.9
5	16	124.50	0I0	0.0	0.0	0.0	0.0	345146	344122	346.16	352.98	364.34	380.12	400136	425.9
	17	142.30	0I0	0.0	0.0	0.0	0.0	345151	344127	346.21	353.04	364.39	380.17	400141	425.9
	18	160.10	0I0	0.0	0.0	0.0	0.0	345169	344144	346.38	353.21	364.56	380.33	400157	426.1
	19	177.90	0I0	0.0	0.0	0.0	0.0	346106	344181	346.75	353.57	364.93	380.70	400193	426.4
	20	195.70	0I0	0.0	0.0	0.0	0.0	346149	345124	347.21	354.12	365.61	381.57	402103	427.7
6	21	213.50	0I0	292.52	292.75	294.49	300.98	324182	324114	327.56	336.39	350.65	370.82	397176	431.5
7	22	213.75	0I0	290.82	291.04	292.76	299.08	316.80	319159	324.23	332.76	346.43	365.74	391127	412.3
	23	215.56	0I0	280.47	280.21	280.36	281.84	285.30	285.99	288.50	292.40	298.44	306.53	315.50	318.7
	24	217.37	0I0	273.74	273.17	272.46	271.94	271.83	271.84	272.04	272.93	274.70	277.14	279.44	279.9
	25	219.19	0I0	270.08	269.42	268.45	267.35	266.28	266.05	265.22	264.73	264.64	264.86	265.04	264.9
8	26	221.00	0I0	268.81	268.13	267.08	265.82	264.48	264.19	263.08	262.04	261.37	260.92	260.45	260.0
	27	226.50	0I0	215.85	215.28	214.32	212.98	211.27	210.82	208.65	206.04	202.92	199.15	194.56	192.0
	28	232.00	0I0	175.84	175.39	174.63	173.56	172.18	171.81	170.07	168.00	165.60	162.86	159.78	158.2
	29	237.50	0I0	145.96	145.66	145.17	144.47	143.59	143.36	142.26	140.98	139.52	137.89	136.12	135.2
	30	243.00	0I0	133.24	133.09	132.84	132.50	132.07	131.96	131.42	130.80	130.10	129.33	128.49	128.0
9	31	248.50	0I0	1465.67	1465.67	1465.67	1465.67	1465.67	1465.67	1465.67	1465.66	1465.66	1465.66	1465.65	1465.6

Table H1.5.4. (cont'd)

HEATING6 02/12/83 HTAS1 SAMPLE PROBLEM 1 -- FIRE TR I8M3033/3033
CLIP1HIP 08.25.57 10-05-8

TRANSIENT TEMPERATURE DISTRIBUTION AFTER 246 TIME STEPS, TIME = 3.000000D+01

GROSS GRID			5				6	7					8
			I				I	I					I
FINE GRID			14	15	16	17	18	19	20	21	22	23	24
		DISTANCE	14.94	16.06	17.19	18.31	19.44	19.60	23.48	27.36	31.24	35.12	39.00
1	1	0.0	1465.65	1465.65	1465.64	1465.64	1465.64	1465.64	1465.62	1465.61	1465.60	1465.78	1474196
	2	5.50	127.70	126.90	126.08	125.24	124.37	124.25	121.19	118.27	115.83	137.81	1469153
	3	11.00	134.42	132.72	130.95	129.14	127.29	127.03	120.56	114.56	109.52	129.51	1469149
	4	16.50	156.71	153.62	150.38	147.05	143.67	143.19	131.64	121.61	113.60	131.42	1469150
	5	22.00	189.37	183.54	177.24	170.82	164.47	163.59	143.31	127.70	116.53	132.61	1469151
2	6	27.50	258174	234.80	217.18	202.22	189.04	187.27	151.55	129.18	115.96	131.96	1469150
	7	29.44	263123	238.96	222.71	208.13	194.49	192.60	150.39	126.68	114.08	131.00	1469150
	8	31.38	276187	242.16	227.28	213.97	200.53	198.56	147.08	123.24	112.34	130.75	1469150
	9	33.31	315178	250.29	237.41	228.53	219.09	217.69	175.44	155.01	147.41	167.29	1469165
3	10	35.25	429107	359.20	377.87	411.56	460.19	472.50	957.79	956.84	955.45	964.08	1472191
4	11	35.50	473177	512.54	560.51	618.81	687183	713105	1443.03	1443.05	1442.96	1443.50	1474187
	12	53.30	460119	489.41	527.78	575.37	633107	654174	0.0	0.0	0.0	0.0	010
	13	71.10	458171	487.79	526.04	573.56	631123	652194	0.0	0.0	0.0	0.0	010
	14	88.90	458138	487.47	525.72	573.25	630193	652164	0.0	0.0	0.0	0.0	010
	15	106.70	458123	487.32	525.59	573.12	630180	652152	0.0	0.0	0.0	0.0	010
5	16	124.50	458119	487.28	525.55	573.08	630176	652148	0.0	0.0	0.0	0.0	010
	17	142.30	458124	487.33	525.59	573.12	630180	652152	0.0	0.0	0.0	0.0	010
	18	160.10	458138	487.47	525.73	573.26	630193	652165	0.0	0.0	0.0	0.0	010
	19	177.90	458172	487.80	526.05	573.57	631124	652195	0.0	0.0	0.0	0.0	010
	20	195.70	460122	489.44	527.80	575.40	633109	654177	0.0	0.0	0.0	0.0	010
6	21	213.50	474101	512.75	560.69	619.08	688102	713124	1444.40	1444.43	1444.35	1444.86	1474188
7	22	213.75	429193	362.90	382.60	417.28	467.19	479.88	979.70	978.97	977.62	985.88	1473100
	23	215.56	319184	251.20	238.47	230.08	221.25	219.96	181.14	161.27	154.03	174.01	1469167
	24	217.37	279179	242.35	227.45	214.28	200.87	198.90	146.10	122.57	112.11	130.85	1469150
	25	219.19	264175	238.90	222.62	208.07	194.39	192.48	149.15	125.55	113.34	130.64	1469150
8	26	221.00	259151	234.69	217.06	202.09	188.84	187.06	150.59	128.16	115.22	131.58	1469150
	27	226.50	189.21	183.35	177.02	170.57	164.18	163.29	142.83	127.20	116.14	132.40	1469150
	28	232.00	156.57	153.46	150.20	146.85	143.46	142.98	131.38	121.35	113.40	131.30	1469150
	29	237.50	134.32	132.61	130.83	129.01	127.16	126.90	120.41	114.42	109.42	129.45	1469149
	30	243.00	127.65	126.85	126.02	125.17	124.31	124.19	121.13	118.21	115.79	137.78	1469153
9	31	248.50	1465.65	1465.65	1465.64	1465.64	1465.64	1465.64	1465.62	1465.61	1465.60	1465.78	1474196

Table H1.5.4. (cont'd)

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	0.0
2	1475.000000
3	1475.000000
4	1475.000000
5	1475.000000
6	1475.000000
7	1475.000000
8	1475.000000

THE CURRENT TIME STEP (DELTAT) = 6.75961994D-01

ELAPSED CPU TIME IS 122.23 SECONDS

THE MAXIMUM TEMPERATURE IS - 1.47496D+03 (+-0.1 PERCENT)

MAX. TEMP. APPEARS AT NODES - 23 253 402 632

THE MINIMUM TEMPERATURE IS - 1.09418D+02 (+-0.1 PERCENT)

MIN. TEMP. APPEARS AT NODES - 67 584

Table H1.5.5. Temperature Distribution at the End of the Three-Hour, Cool-Down Transient.

HEATING6 02/12/83 HTAS1 SAMPLE PROBLEM 1 -- POSTFIRE TR IBM3033/3033
CLIP1HIP 08.33.50 10-05-8

TRANSIENT TEMPERATURE DISTRIBUTION AFTER 227 TIME STEPS, TIME = 2.10000D+02

GROSS GRID			1					2	3				4	
	FINE GRID		I					I	I				I	
			1	2	3	4	5	6	7	8	9	10	11	12	13
		DISTANCE	-1.35	1.35	2.70	4.05	5.40	6.75	7.06	8.39	9.71	11.04	12.36	13.69	14.3
1	1	0.0	OI0	~~137.91~~	~~137.90~~	~~137.89~~	~~137.88~~	~~137.86~~	~~137.86~~	~~137.84~~	~~137.81~~	~~137.79~~	~~137.76~~	~~137.72~~	~~137.7~~
	2	5.50	OI0	144.23	144.08	143.84	143.51	143.08	142.96	142.43	141.82	141.13	140.36	139.53	139..
	3	11.00	OI0	147.13	146.84	146.35	145.67	144.79	144.56	143.47	142.20	140.75	139.14	137.37	136.4
	4	16.50	OI0	175.89	175.44	174.70	173.65	172.30	171.94	170.22	168.19	165.81	163.10	160.03	158.4
	5	22.00	OI0	219.30	218.83	218.03	216.90	215.40	215.00	213.03	210.61	207.62	203.91	199.19	196.3
2	6	27.50	OI0	~~383.72~~	~~383.59~~	~~383.34~~	~~382.96~~	~~382.43~~	~~382.28~~	~~381.57~~	~~380.71~~	~~379.71~~	~~378.55~~	~~377.15~~	~~376.2~~
	7	29.44	OI0	386.27	386.08	385.73	385.22	384.52	384.34	383.46	382.46	381.38	380.27	379.21	378.7
	8	31.38	OI0	391.81	391.49	390.92	390.06	388.91	388.62	387.28	385.86	384.44	383.10	381.9[illegible]	381.5
	9	33.31	OI0	400.24	399.73	398.79	397.29	395.19	394.67	392.39	390.14	387.98	385.96	384.20	383.5
3	10	35.25	OI0	411.57	410.90	409.57	407.17	402.88	401I67	~~397.78~~	~~394.29~~	~~391.01~~	~~387.82~~	~~384I49~~	382.1
4	11	35.50	OI0	~~413.25~~	~~412.58~~	~~411.25~~	~~408.84~~	~~404I16~~	402I05	398.08	394.51	391.17	387.90	384I62	381.3
	12	53.30	OI0	0.0	0.0	0.0	0.0	429I51	427I40	423.20	419.37	415.77	412.33	408I98	405.6
	13	71.10	OI0	0.0	0.0	0.0	0.0	438I50	436I40	432.16	428.25	424.54	420.95	417I44	413.9
	14	88.90	OI0	0.0	0.0	0.0	0.0	440I19	438I09	433.84	429.91	426.16	422.52	418I94	415.3
	15	106.70	OI0	0.0	0.0	0.0	0.0	440I34	438I24	434.00	430.06	426.30	422.65	419I06	415.4
5	16	124.50	OI0	0.0	0.0	0.0	0.0	440I33	438I23	433.99	430.05	426.29	422.64	419I04	415.4
	17	142.30	OI0	0.0	0.0	0.0	0.0	440I35	438I24	434.00	430.06	426.30	422.65	419I06	415.4
	18	160.10	OI0	0.0	0.0	0.0	0.0	440I20	438I10	433.85	429.92	426.17	422.53	418I95	415.3
	19	177.90	OI0	0.0	0.0	0.0	0.0	438I57	436I46	432.23	428.31	424.60	421.01	417I49	413.9
	20	195.70	OI0	0.0	0.0	0.0	0.0	429I90	427I79	423.59	419.75	416.14	412.68	409I31	405.9
6	21	213.50	OI0	~~417.68~~	~~416.84~~	~~415.21~~	~~412.29~~	~~406I56~~	404I43	400.37	396.71	393.25	389.86	386I44	382.9
7	22	213.75	OI0	416.00	415.17	413.54	410.64	405.50	404I16	~~400.15~~	~~396.56~~	~~393.17~~	~~389.84~~	~~386I40~~	383.9
	23	215.56	OI0	405.59	404.94	403.75	401.90	399.35	398.75	396.20	393.74	391.39	389.20	387.28	386.6
	24	217.37	OI0	398.02	397.59	396.83	395.72	394.29	393.94	392.34	390.70	389.08	387.57	386.31	385.8
	25	219.19	OI0	393.12	392.84	392.36	391.65	390.74	390.50	389.41	388.[illegible]	386.96	385.71	384.54	384.0
8	26	221.00	OI0	~~390.88~~	~~390.67~~	~~390.30~~	~~389.75~~	~~389.02~~	~~388.83~~	~~387.93~~	~~386.88~~	~~385.73~~	~~384.45~~	~~382.96~~	~~382.0~~
	27	226.50	OI0	219.94	219.45	218.63	217.46	215.93	215.51	213.51	211.04	208.01	204.26	199.47	196.6
	28	232.00	OI0	175.96	175.51	174.75	173.69	172.31	171.95	170.21	168.15	165.75	163.01	159.92	158.3
	29	237.50	OI0	147.12	146.82	146.33	145.64	144.76	144.52	143.42	142.14	140.69	139.06	137.28	136.4
	30	243.00	OI0	144.21	144.06	143.82	143.48	143.05	142.94	142.40	141.78	141.09	140.32	139.48	139.0
9	31	248.50	OI0	~~137.91~~	~~137.90~~	~~137.89~~	~~137.88~~	~~137.86~~	~~137.86~~	~~137.84~~	~~137.81~~	~~137.78~~	~~137.75~~	~~137.72~~	~~137.7~~

Table H1.5.5. (cont'd)

HEATING6 02/12/83 HTAS1 SAMPLE PROBLEM 1 -- POSTFIRE TR IBM3033/3033
CLIP1H1P 08.33.50 10-05-8

TRANSIENT TEMPERATURE DISTRIBUTION AFTER 227 TIME STEPS, TIME = 2.10000D+02

		GROSS GRID	5				6	7					8
			I				I	I					I
		FINE GRID	14	15	16	17	18	19	20	21	22	23	24
		DISTANCE	14.94	16.06	17.19	18.31	19.44	19.60	23.48	27.36	31.24	35.12	39.00
1	1	0.0	137.69	137.66	137.63	137.59	137.56	137.56	137.44	137.33	137.27	136.14	106175
	2	5.50	138.70	137.91	137.09	136.25	135.39	135.27	132.23	129.59	130.84	156.61	118145
	3	11.00	135.58	133.88	132.12	130.30	128.46	128.19	121.74	116.03	115.03	142.39	117185
	4	16.50	156.83	153.70	150.43	147.09	143.71	143.23	131.69	121.95	118.00	143.34	117192
	5	22.00	193.08	185.47	177.98	171.05	164.52	163.62	143.33	128.01	120.89	144.50	118100
2	6	27.50	375123	286.05	235.00	207.42	190.24	188.08	152.09	130.03	120.86	144.32	117198
	7	29.44	378143	295.83	245.17	216.75	198.68	196.39	155.84	132.76	124.18	147.80	118118
	8	31.38	381127	309.71	262.83	235.66	218.16	215.88	171.17	149.12	142.05	163.67	118197
	9	33.31	383125	338.83	306.18	286.09	271.86	269.70	202.03	184.67	180.12	193.26	119117
3	10	35.25	380149	375.71	369.59	363.77	355.64	352.97	146.04	137.22	136.06	137.96	107148
4	11	35.50	377170	374.78	371.53	367.93	363199	362148	127.85	122.74	122.08	123.13	104189
	12	53.30	401196	399.27	396.27	392.99	389144	388104	0.0	0.0	0.0	0.0	010
	13	71.10	410112	407.31	404.20	400.80	397113	395167	0.0	0.0	0.0	0.0	010
	14	88.90	411152	408.68	405.54	402.11	398141	396195	0.0	0.0	0.0	0.0	010
	15	106.70	411162	408.77	405.63	402.20	398150	397103	0.0	0.0	0.0	0.0	010
5	16	124.50	411160	408.76	405.62	402.19	398148	397102	0.0	0.0	0.0	0.0	010
	17	142.30	411162	408.78	405.63	402.20	398150	397103	0.0	0.0	0.0	0.0	010
	18	160.10	411153	408.69	405.55	402.12	398142	396195	0.0	0.0	0.0	0.0	010
	19	177.90	410117	407.36	404.25	400.85	397117	395172	0.0	0.0	0.0	0.0	010
	20	195.70	402126	399.55	396.55	393.27	389170	388130	0.0	0.0	0.0	0.0	010
6	21	213.50	379119	376.17	372.84	369.18	365121	363169	127.00	121.90	121.25	122.42	104181
7	22	213.75	382118	376.65	370.21	364.28	356.20	353.54	144.60	135.82	134.68	136.80	107135
	23	215.56	386125	342.75	310.64	290.91	276.80	274.61	202.83	185.93	181.64	194.07	118198
	24	217.37	385158	314.03	266.62	239.23	221.72	219.44	174.05	152.52	145.85	166.99	119111
	25	219.19	383171	299.79	247.48	218.17	199.75	197.41	156.39	133.53	125.33	149.04	118125
8	26	221.00	380196	289.82	236.68	207.97	190.28	188.07	151.37	129.27	120.38	144.16	117197
	27	226.50	193.25	185.48	177.85	170.83	164.24	163.32	142.86	127.52	120.51	144.29	117198
	28	232.00	156.69	153.55	150.26	146.89	143.50	143.02	131.43	121.69	117.79	143.23	117191
	29	237.50	135.48	133.77	132.00	130.18	128.33	128.06	121.59	115.89	114.93	142.33	117185
	30	243.00	138.64	137.85	137.03	136.18	135.33	135.20	132.17	129.53	130.79	156.59	118145
9	31	248.50	137.69	137.66	137.62	137.59	137.56	137.55	137.43	137.33	137.27	136.14	106175

Table H1.5.5. (cont'd)

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	0.0
2	100.000000
3	100.000000
4	100.000000
5	100.000000
6	100.000000
7	100.000000
8	100.000000

THE CURRENT TIME STEP (DELTAT) = 5.00000000D+00

ELAPSED CPU TIME IS 141.24 SECONDS

THE MAXIMUM TEMPERATURE IS - 4.40346D+02 (+-0.1 PERCENT)

MAX. TEMP. APPEARS AT NODES - 282 296 310 324 338

THE MINIMUM TEMPERATURE IS - 1.04808D+02 (+-0.1 PERCENT)

MIN. TEMP. APPEARS AT NODES - 253 402

Table H1.5.6. Temperature Distribution at the Final Steady State.

HEATING6 02/12/83 HTAS1 SAMPLE PROBLEM 1 -- FINAL SS IBM3033/3033
CLIP1H1P 08.35.04 10-05-8

STEADY STATE TEMPERATURE DISTRIBUTION AFTER 12 ITERATIONS, TIME = 2.10000D+02

GROSS GRID			1 I				2 I	3 I				4 I	
	FINE GRID		1	2	3	4	5	6	7	8	9	10	11	12	13
	DISTANCE		-1.35	1.35	2.70	4.05	5.40	6.75	7.06	8.39	9.71	11.04	12.36	13.69	14.3
1	1	0.0	0I0	101.52	101.51	101.49	101.47	101.45	101.44	101.40	101.36	101.32	101.27	101.21	101.1
	2	5.50	0I0	118.80	118.68	118.47	118.18	117.82	117.72	117.27	116.75	116.16	115.52	114.81	114.4
	3	11.00	0I0	138.82	138.57	138.15	137.56	136.81	136.61	135.68	134.59	133.35	131.97	130.46	129.7
	4	16.50	0I0	164.42	164.03	163.38	162.47	161.28	160.97	159.48	157.72	155.67	153.33	150.70	149.3
	5	22.00	0I0	198.65	198.15	197.32	196.17	194.69	194.31	192.44	190.20	187.52	184.29	180.36	178.1
2	6	27.50	0I0	243.89	243.20	242.13	240.77	239.26	238.91	237.42	236.05	234.86	233.90	233.19	232.9
	7	29.44	0I0	244.93	244.16	242.94	241.40	239.68	239.29	237.63	236.13	234.84	233.83	233.15	232.9
	8	31.38	0I0	248.09	247.05	245.38	243.21	240.79	240.26	238.06	236.14	234.53	233.26	232.41	232.1
	9	33.31	0I0	254.13	252.64	250.15	246.71	242.62	241.78	238.61	236.01	233.84	232.06	230.72	230.3
3	10	35.25	0I0	264.24	262.34	258.93	253.46	244.78	243I03	238.93	235.65	232.81	230.29	227I93	226.3
4	11	35.50	0I0	265.92	264.02	260.60	255.07	245I05	242I84	238.93	235.63	232.75	230.19	227I84	225.5
	12	53.30	0I0	0.0	0.0	0.0	0.0	242I04	239I95	236.31	233.19	230.46	228.04	225I86	223.7
	13	71.10	0I0	0.0	0.0	0.0	0.0	241I19	239I09	235.45	232.35	229.64	227.24	225I08	222.9
	14	88.90	0I0	0.0	0.0	0.0	0.0	240I79	238I70	235.06	231.96	229.25	226.85	224I70	222.6
	15	106.70	0I0	0.0	0.0	0.0	0.0	240I61	238I52	234.88	231.78	229.08	226.68	224I53	222.4
5	16	124.50	0I0	0.0	0.0	0.0	0.0	240I56	238I47	234.83	231.73	229.03	226.63	224I48	222.3
	17	142.30	0I0	0.0	0.0	0.0	0.0	240I62	238I52	234.88	231.78	229.08	226.68	224I53	222.4
	18	160.10	0I0	0.0	0.0	0.0	0.0	240I80	238I70	235.06	231.96	229.26	226.86	224I70	222.6
	19	177.90	0I0	0.0	0.0	0.0	0.0	241I[illegible]	239I10	235.46	232.36	229.65	227.25	225I09	223.0
	20	195.70	0I0	0.0	0.0	0.0	0.0	242I07	239I97	236.33	233.21	230.48	228.06	225I88	223.7
6	21	213.50	0I0	266.34	264.42	260.96	255.35	245I12	242I91	238.99	235.68	232.80	230.23	227I88	225.5
7	22	213.75	0I0	264.66	262.74	259.29	253.74	244.88	243I10	239.00	235.71	232.86	230.33	227I97	226.3
	23	215.56	0I0	255.08	253.53	250.92	247.29	242.95	242.06	238.77	236.07	233.82	231.98	230.58	230.1
	24	217.37	0I0	249.22	248.09	246.26	243.89	241.25	240.67	238.29	236.23	234.49	233.13	232.21	231.9
	25	219.19	0I0	246.10	245.23	243.86	242.12	240.19	239.75	237.91	236.24	234.82	233.70	232.95	232.7
8	26	221.00	0I0	245.06	244.28	243.05	241.50	239.78	239.39	237.71	236.17	234.85	233.79	233.01	232.7
	27	226.50	0I0	198.97	198.45	197.58	196.38	194.84	194.44	192.51	190.21	187.47	184.20	180.23	178.0
	28	232.00	0I0	164.48	164.09	163.42	162.49	161.29	160.97	159.46	157.67	155.60	153.24	150.59	149.2
	29	237.50	0I0	138.81	138.55	138.13	137.53	136.77	136.57	135.63	134.53	133.28	131.89	130.37	129.6
	30	243.00	0I0	118.78	118.66	118.45	118.16	117.79	117.70	117.24	116.72	116.13	115.47	114.77	114.4
9	31	248.50	0I0	101.52	101.51	101.49	101.47	101.44	101.44	101.40	101.36	101.31	101.26	101.21	101.1

Table H1.5.6. (cont'd)

HEATING6 02/12/83 HTAS1 SAMPLE PROBLEM 1 -- FINAL SS I8M3033/3033
CLIP1HIP 08.35.04 10-05-8

STEADY STATE TEMPERATURE DISTRIBUTION AFTER 12 ITERATIONS, TIME = 2.10000D+02

		GROSS GRID	5				6	7					8
			I				I	I					I
		FINE GRID	14	15	16	17	18	19	20	21	22	23	24
		DISTANCE	14.94	16.06	17.19	18.31	19.44	19.60	23.48	27.36	31.24	35.12	39.00
1	1	0.0	101.16	101.11	101.05	101.00	100.94	100.93	100.73	100.53	100.35	100.18	100I02
	2	5.50	114.10	113.43	112.74	112.03	111.31	111.20	108.64	106.19	103.97	101.99	100I23
	3	11.00	128.93	127.48	125.97	124.43	122.86	122.64	117.14	112.06	107.61	103.78	100I43
	4	16.50	147.97	145.32	142.55	139.69	136.81	136.40	126.56	118.04	111.06	105.38	100I60
	5	22.00	175.78	170.77	165.37	159.86	154.43	153.67	136.37	123.12	113.47	106.35	100I70
2	6	27.50	232I68	214.30	199.23	186.38	175.07	173.55	143.11	124.21	112.91	105.78	100I64
	7	29.44	232I85	217.51	203.72	191.16	179.44	177.82	141.91	121.98	111.27	104.95	100I55
	8	31.38	232I08	219.13	207.16	195.63	183.98	182.29	137.87	117.99	108.80	103.79	100I42
	9	33.31	230I12	220.43	211.57	202.53	192.08	190.34	129.04	111.85	105.50	102.33	100I26
3	10	35.25	225I28	221.21	218.28	215.39	211.28	210.01	110.82	103.42	101.57	100.67	100I08
4	11	35.50	223I19	221.11	219.17	217.36	215I65	215I16	106.97	102.24	101.04	100.45	100I05
	12	53.30	221I72	220.06	218.51	217.05	215I68	215I29	0.0	0.0	0.0	0.0	0I0
	13	71.10	220I99	219.35	217.81	216.37	215I01	214I62	0.0	0.0	0.0	0.0	0I0
	14	88.90	220I62	218.98	217.45	216.01	214I65	214I27	0.0	0.0	0.0	0.0	0I0
	15	106.70	220I45	218.81	217.28	215.85	214I49	214I11	0.0	0.0	0.0	0.0	0I0
5	16	124.50	220I40	218.77	217.24	215.80	214I45	214I06	0.0	0.0	0.0	0.0	0I0
	17	142.30	220I45	218.82	217.29	215.85	214I49	214I11	0.0	0.0	0.0	0.0	0I0
	18	160.10	220I62	218.99	217.45	216.02	214I66	214I27	0.0	0.0	0.0	0.0	0I0
	19	177.90	221I00	219.36	217.82	216.38	215I02	214I63	0.0	0.0	0.0	0.0	0I0
	20	195.70	221I74	220.08	218.53	217.07	215I70	215I31	0.0	0.0	0.0	0.0	0I0
6	21	213.50	223I22	221.14	219.29	217.38	215I67	215I15	106.89	102.22	101.03	100.45	100I05
7	22	213.75	225I27	221.21	218.27	215.38	211.31	210.06	110.70	103.38	101.54	100.66	100I08
	23	215.56	229I94	220.41	211.75	202.89	192.57	190.83	127.99	111.21	105.17	102.18	100I25
	24	217.37	231I86	219.03	207.20	195.77	184.14	182.43	136.66	117.06	108.26	103.54	100I40
	25	219.19	232I63	217.31	203.57	191.03	179.27	177.64	140.80	120.99	110.63	104.64	100I52
8	26	221.00	232I47	214.12	199.07	186.20	174.83	173.30	142.24	123.31	112.28	105.46	100I60
	27	226.50	175.61	170.58	165.15	159.61	154.14	153.38	135.94	122.68	113.14	106.17	100I68
	28	232.00	147.83	145.17	142.38	139.51	136.62	136.21	126.32	117.81	110.88	105.28	100I59
	29	237.50	128.84	127.38	125.87	124.32	122.74	122.52	117.01	111.94	107.52	103.73	100I42
	30	243.00	114.06	113.38	112.69	111.97	111.25	111.14	108.58	106.14	103.92	101.97	100I23
9	31	248.50	101.15	101.10	101.05	100.99	100.94	100.93	100.72	100.53	100.34	100.18	100I02

Table H1.5.6. (cont'd)

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	0.0
2	100.000000
3	100.000000
4	100.000000
5	100.000000
6	100.000000
7	100.000000
8	100.000000

ELAPSED CPU TIME IS 7.74 SECONDS

THE MAXIMUM TEMPERATURE IS - 2.66342D+02 (+-0.1 PERCENT)

MAX. TEMP. APPEARS AT NODES - 380

THE MINIMUM TEMPERATURE IS - 1.00021D+02 (+-0.1 PERCENT)

MIN. TEMP. APPEARS AT NODES - 23 230 253 402 425
632

Table H1.5.7. Output Describing the Temperature Profile Plots Generated by HTAS1

```
*****************************************************
* THE TEMPERATURE PLOTTING OPTION HAS BEEN INVOKED *
*****************************************************

PLOTS WILL INCLUDE TEMPERATURE DISTRIBUTIONS FROM THE FIRE TRANSIENT AND POSTFIRE TRANSIENT HEATING6 CALCULATIONS

TEMPERATURE VS. TIME CURVES FOR THE INNER SHELL, SHIELDING, AND OUTER SHELL WILL BE GENERATED ON ONE PLOT

TEMPERATURE DISTRIBUTIONS AT EVERY PRINTOUT TIME WILL BE USED

A PLOT WILL BE GENERATED SHOWING A CURVE OF MAXIMUM TEMPERATURE AT ANY TIME
AT EACH CENTER AXIAL NODE VS. THE R DISTANCE FROM THE ORIGIN

A PLOT WILL BE GENERATED SHOWING A CURVE OF THE DIFFERENCE IN THE VOLUME AVERAGED
TEMPERATURES OF THE INNER SHELL AND THE OUTER SHELL
```

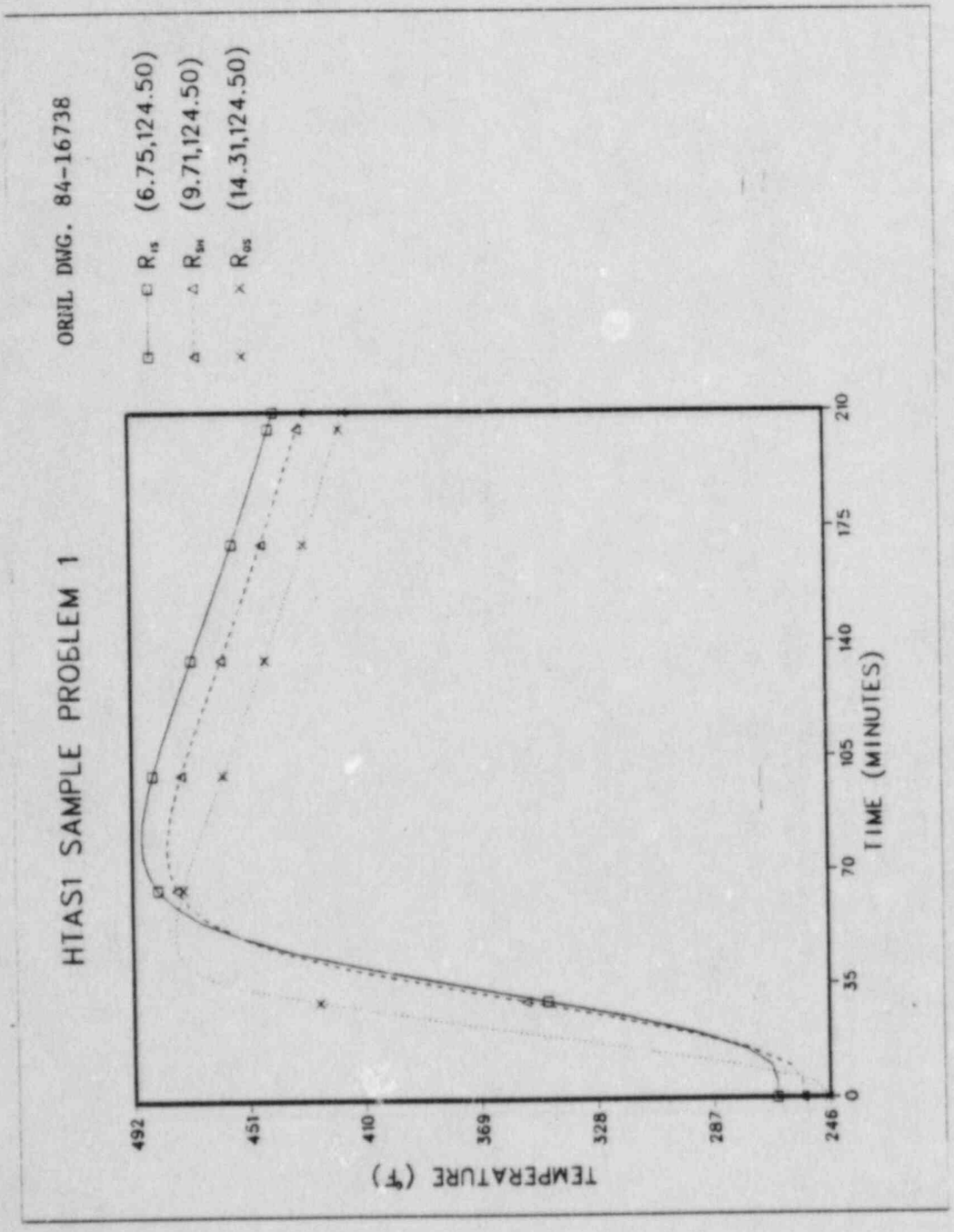


Fig. H1.5.1. Temperature profile plots generated for the first sample problem.

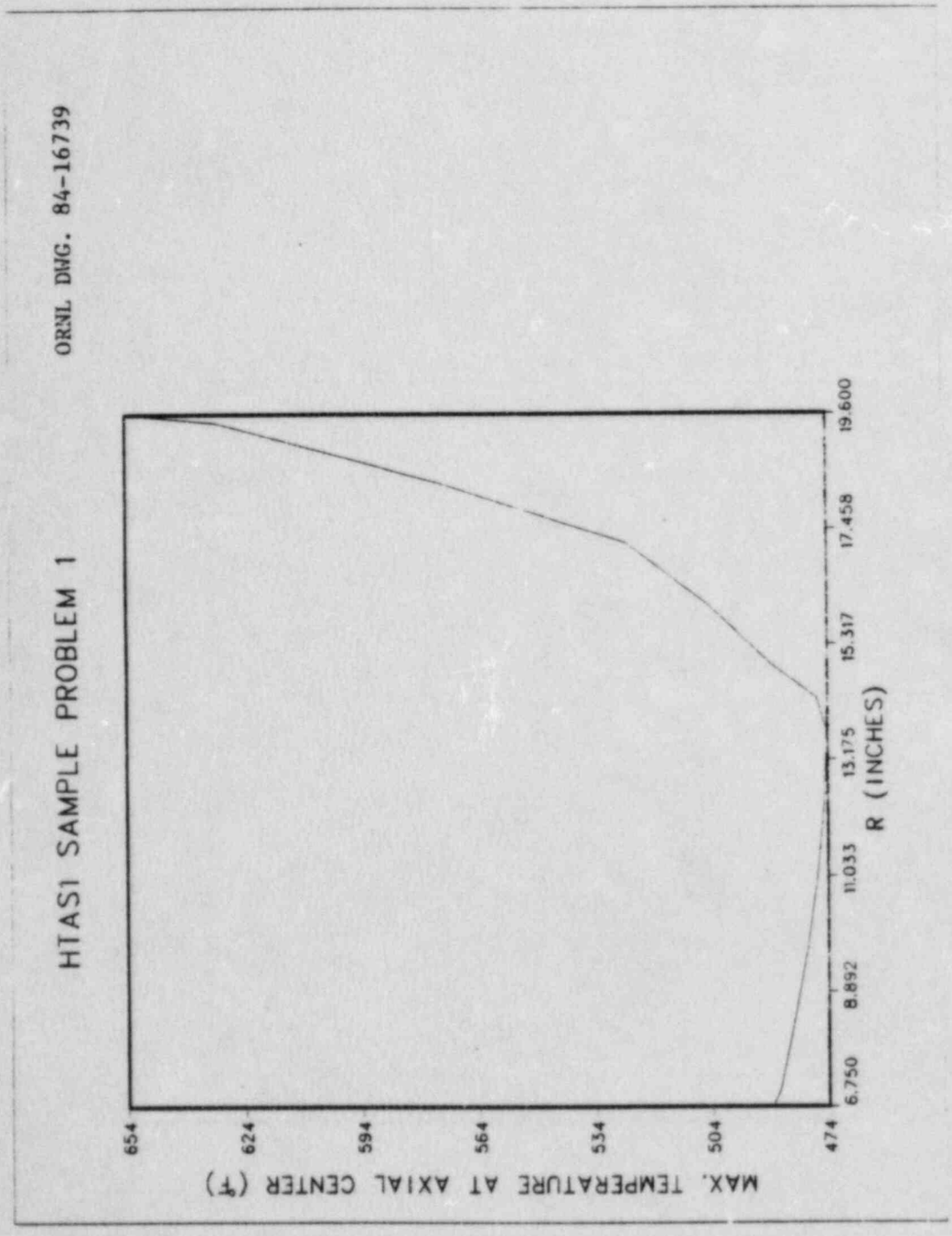


Fig. H1.5.1 (continued)

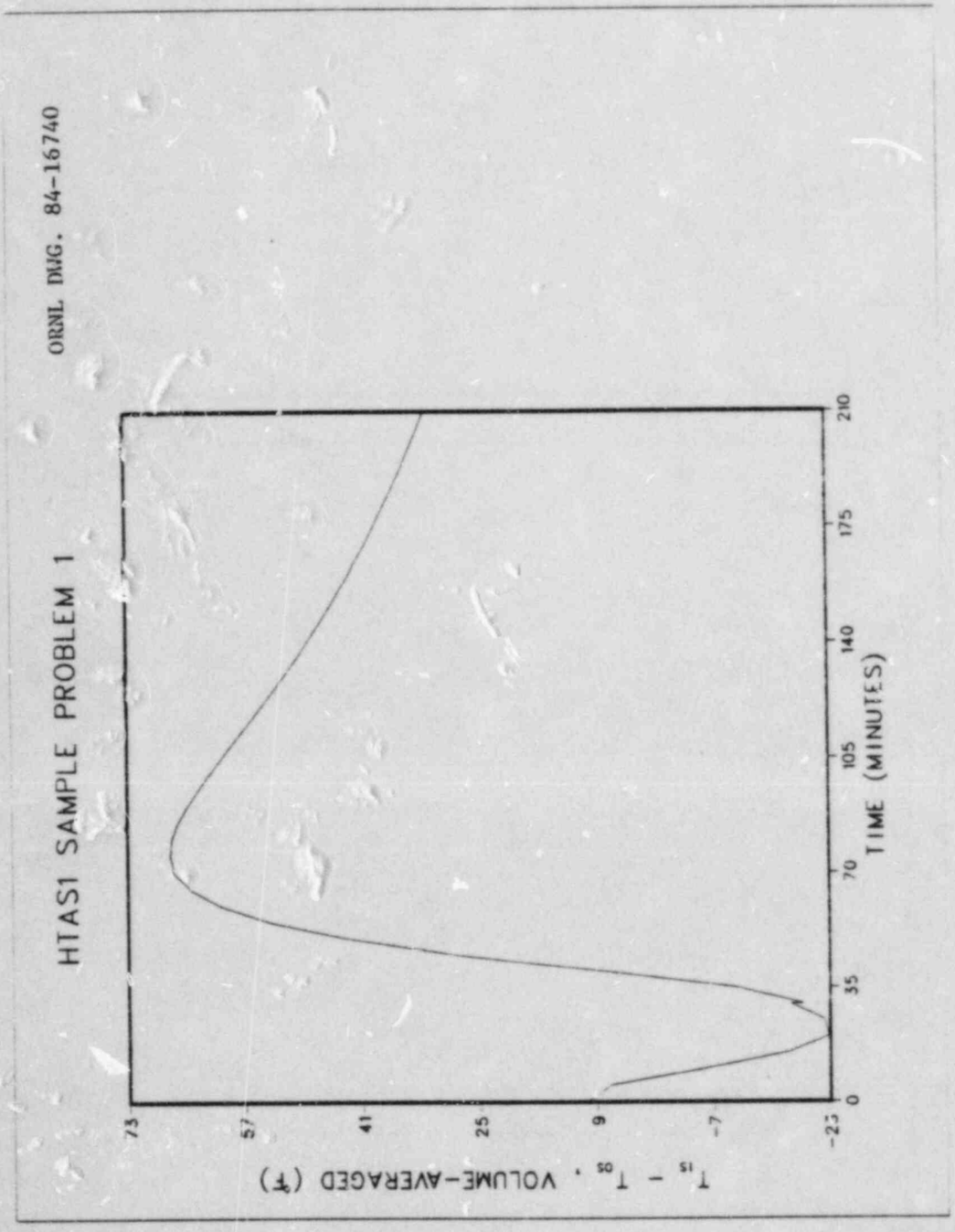


Fig. H1.5.1 (continued)

Table H1.5.8 Thermal Properties for 1.5% Carbon Steel.

Temperature (°F)	Thermal Conductivity[a] Btu/(hr-ft-°F)	Density[a] (lb/ft^3)	Temperature (°F)	Specific Heat[b] Btu/(lb-°F)
32	21	484	32	0.105
212	21		167	0.120
392	21		392	0.135
572	20		752	0.15
752	19		1112	0.17
1112	18		1292	0.20
1472	16			

[a] Ref. 1
[b] Ref. 2

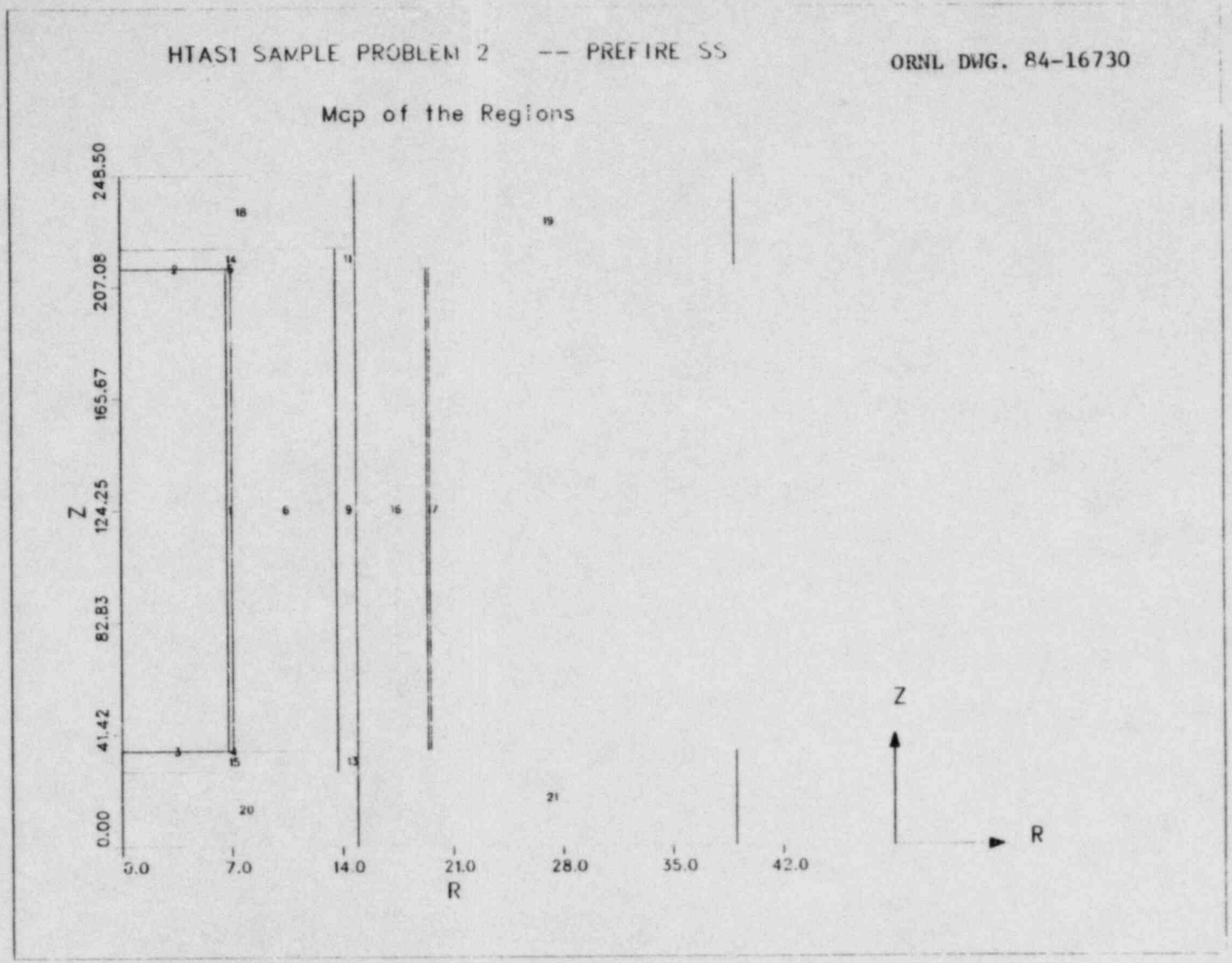
The input data necessary for HTAS1 to generate region plots for the second sample problem is shown in Table H1.5.9. Several of the default region plots generated by HTAS1 are shown in Fig. H1.5.2. The map of the regions for the fire calculation and all the plots for the final steady calculation are omitted because they are identical to that provided for the previous calculational phase. If the REGP data card is removed from the input data, HTAS1 will perform the analytical sequence described above for the second sample problem. Microfiche copies of the output for this problem when region plots are generated and when the analytical sequence is performed are presented in Appendix H1.B. Generating the region plots for the second sample problem required 27 seconds of CPU time on the IBM 3033, 636K bytes of core, 2902 units of IO, and 1927 lines of output. Performing the analytical sequence required 3 minutes, 35 seconds of CPU time on the IBM 3033, 636K bytes of core, 3217 units of IO, and 9100 lines of output. Factors contributing to the reduced CPU time for the second sample problem include the loss of the neutron shield and the bottom impact limiter during the calculations and the fact that the second sample problem uses larger time step sizes than the first sample problem.

Table H1.5.9 Input Data for Second Sample Problem for HTAS1.

```
HTAS1 SAMPLE PROBLEM 2

CAVITY  6.75 5 89 5 39247
*
* EXAMPLE OF OVERRIDING THE DEFAULT MATERIAL FOR THE INNER SHELL BY
* EXPLICITLY SPECIFYING THE MATERIAL AND ITS THERMAL PROPERTIES.
INNER SHELL  0.31 1 0.25 1
MATERIAL
CARBONST
   32 21
@ 212 21
@ 392 21
@ 572 20
@ 752 19
@1112 18
@1472 16
484
   32 0.105
@ 167 0.12
@ 392 0.135
@ 752 0.15
@1112 0.17
@1292 0.2
SHIELDING  6.63 5
OUTER SHELL  1.25 2 7.25 4 7.75 4
*
* EXAMPLE OF OVERRIDING DEFAULT EMISSIVITIES OF THE OUTER SHELL FOR
* THE POSTFIRE TRANSIENT AND FINAL STEADY STATE CALCULATIONS.
EMISSIVITIES  0 0 0 0.9 0.9
*
* EXAMPLE OF OVERRIDING THE DEFAULT MATERIAL FOR THE OUTER SHELL BY
* REPLACING IT WITH A MATERIAL FROM THE LLNL MATERIAL PROPERTY LIBRARY
* (MATERIAL NUMBER 3135 -- STAINLESS STEEL 347).
MATERIAL
 *3135
NEUTRON SHIELD  4.5 4 85.5 86
*
* EXAMPLE OF REPLACING THE NEUTRON SHIELD WITH A VOID AFTER THE PREFIRE
* STEADY STATE CALCULATIONS.  ALSO, OVERRIDE THE DEFAULT EMISSIVITIES
* FOR THE RADIATIVE HEAT TRANSFER ACROSS THE RESULTING GAP.
DELETE  PREFIRE  -1 0.6 0.6
WATER JACKET  0.16 1
*
* EXAMPLE OF OVERRIDING THE DEFAULT MATERIAL IN THE WATER JACKET BY
* REPLACING IT WITH MATERIAL NUMBER 14 FROM THE HTAS1 MATERIAL PROPERTY
* LIBRARY.
MATERIAL  14
```

Table H1.5.9 (cont'd)

```
IMPACT LIMITER  39 5 27.5 5 0 0 11
*
* EXAMPLE OF REMOVING THE BOTTOM IMPACT LIMITER AFTER THE FIRE
* TRANSIENT CALCULATIONS.
DELETE  FIRE  BOTTOM
*
* EXAMPLE OF REQUESTING PRINTED OUTPUT OF THE TRANSIENT TEMPERATURE
* DISTRIBUTIONS AT SELECTED TIMES.
PRINTOUT TIMES  5 10 15 20 25 30 40 50 60 70 80 90 110 130 150 170 190
@ 210 230 250 270
*
* EXAMPLE OF OVERRIDING THE DEFAULT AMBIENT TEMPERATURE FOR THE PREFIRE
* STEADY STATE CALCULATIONS.
PREFIRE  130
FIRE
*
* EXAMPLE OF OVERRIDING THE DEFAULT NUMERICAL TECHNIQUE FOR THE FIRE
* TRANSIENT CALCULATIONS.  USE CRANK-NICOLSON WITH A MAXIMUM TIME STEP
* SIZE OF 5 MINUTES AND ADJUST THE TIME STEP SIZE TO KEEP THE MAXIMUM
* RELATIVE CHANGE IN TEMPERATURE NEAR 5 PER CENT OVER A TIME STEP.
IMPLICIT

0 0 5 0 0 5
*
* EXAMPLE OF OVERRIDING THE DEFAULT DURATION TIME AND AMBIENT
* TEMPERATURE FOR THE POSTFIRE TRANSIENT CALCULATIONS.
POSTFIRE  240 130
*
* EXAMPLE OF OVERRIDING THE DEFAULT NUMERICAL TECHNIQUE FOR THE POSTFIRE
* TRANSIENT CALCULATIONS.  USE CRANK-NICOLSON WITH A MAXIMUM TIME STEP
* SIZE OF 5 MINUTES AND ADJUST THE TIME STEP SIZE TO KEEP THE MAXIMUM
* RELATIVE CHANGE IN TEMPERATURE NEAR 5 PER CENT OVER A TIME STEP.
IMPLICIT

0 0 5 0 0 5
*
* EXAMPLE OF OVERRIDING THE DEFAULT AMBIENT TEMPERATURE FOR THE FINAL
* STEADY STATE CALCULATIONS.
FINAL  130
*
* APPENDIX H1.B CONTAINS A MICROFICHE COPY OF THE OUTPUT FOR THIS
* PROBLEM WHEN THE REGP DATA CARD IS INCLUDED AND THE DEFAULT REGION
* PLOTS ARE GENERATED.  APPENDIX H1.B ALSO CONTAINS A MICROFICHE COPY
* OF THE OUTPUT FOR THIS PROBLEM WHEN THE REGP DATA CARD IS NOT
* INCLUDED AND THE ANALYTICAL SEQUENCE IS PERFORMED.
REGP
%
```



H1.5.33

Revised 12/84

Fig. H1.5.2. Region plots generated for the second sample problem.

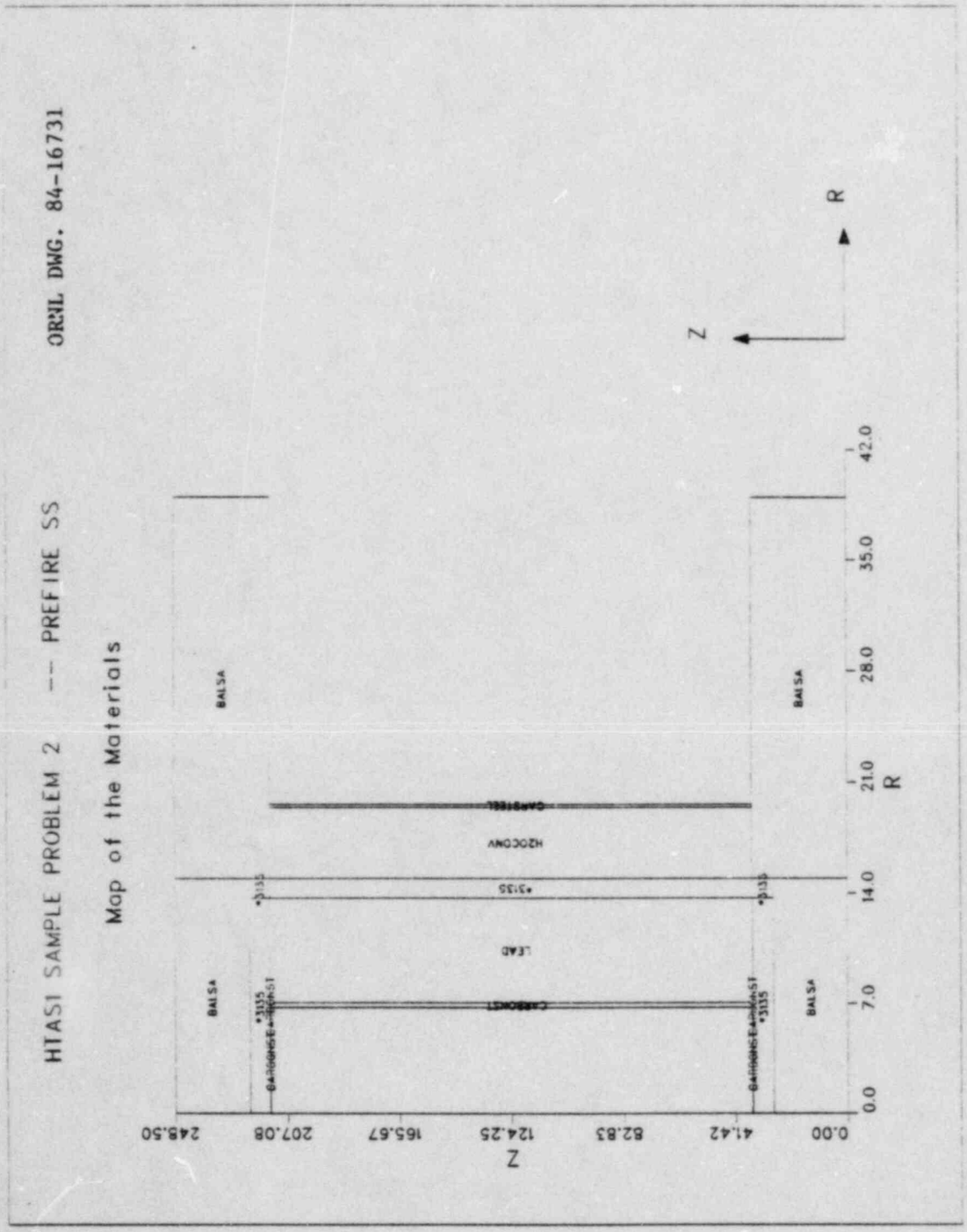


Fig. H1.5.2 (continued)

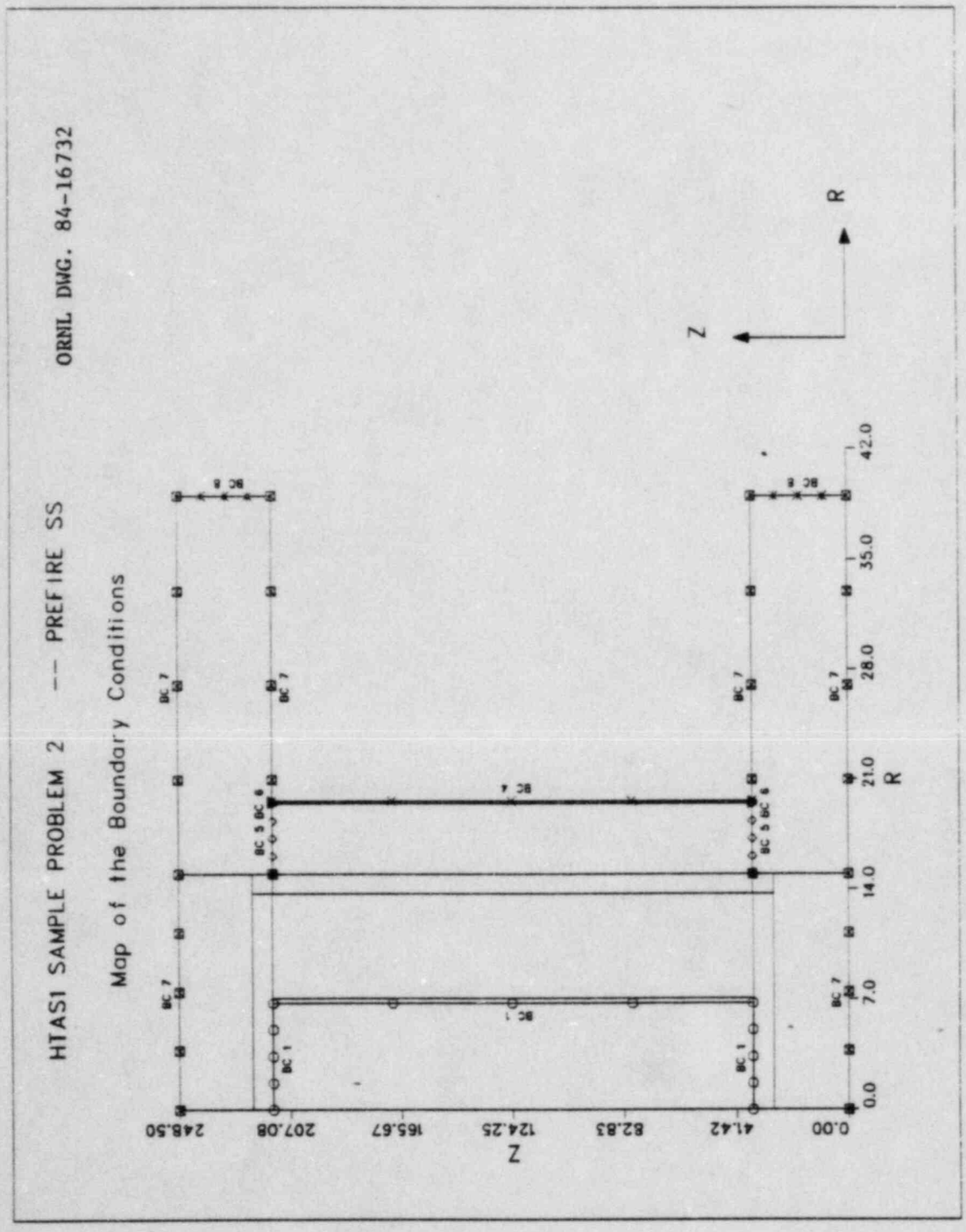


Fig. H1.5.2 (continued)

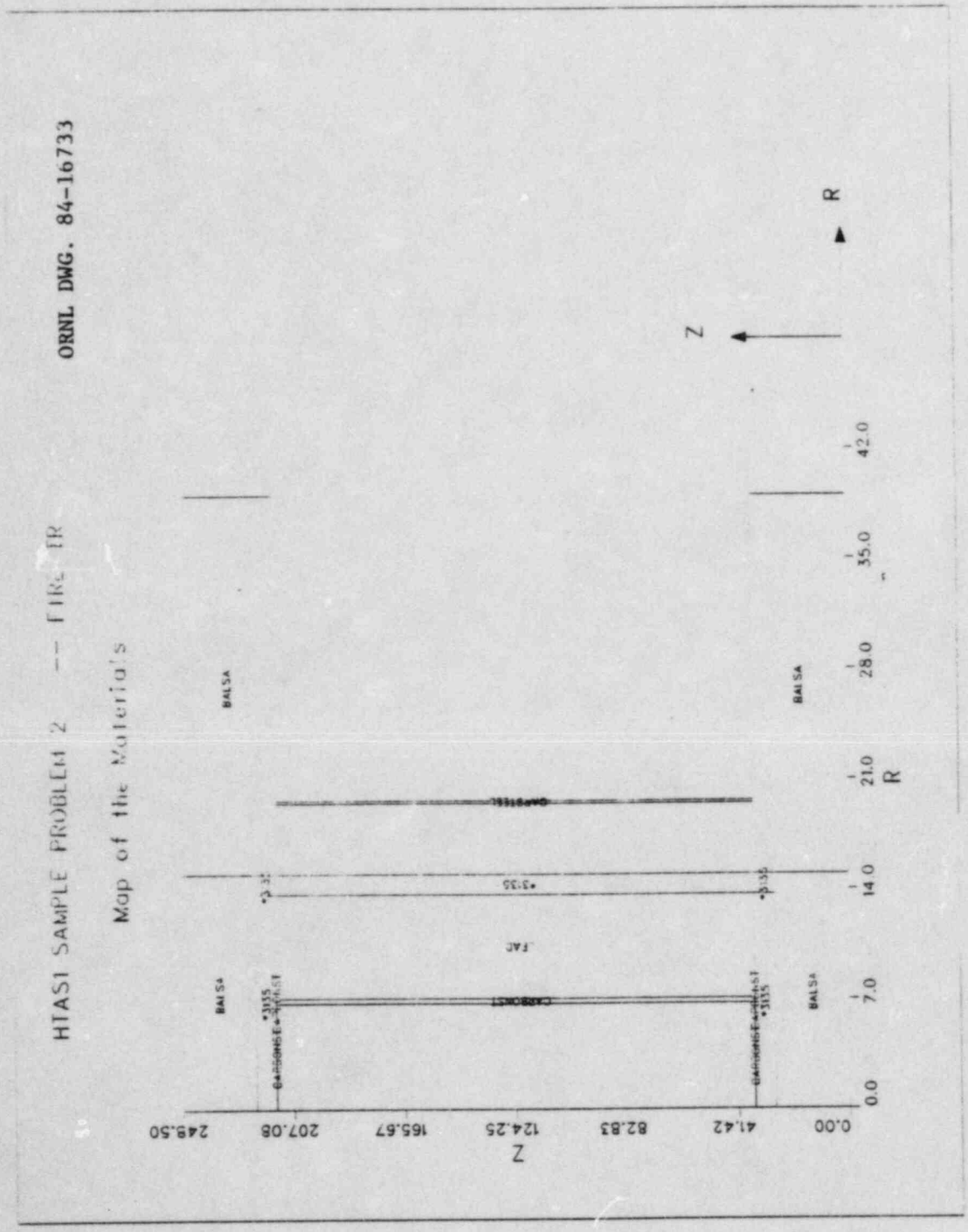


Fig. H1.5.2 (continued)

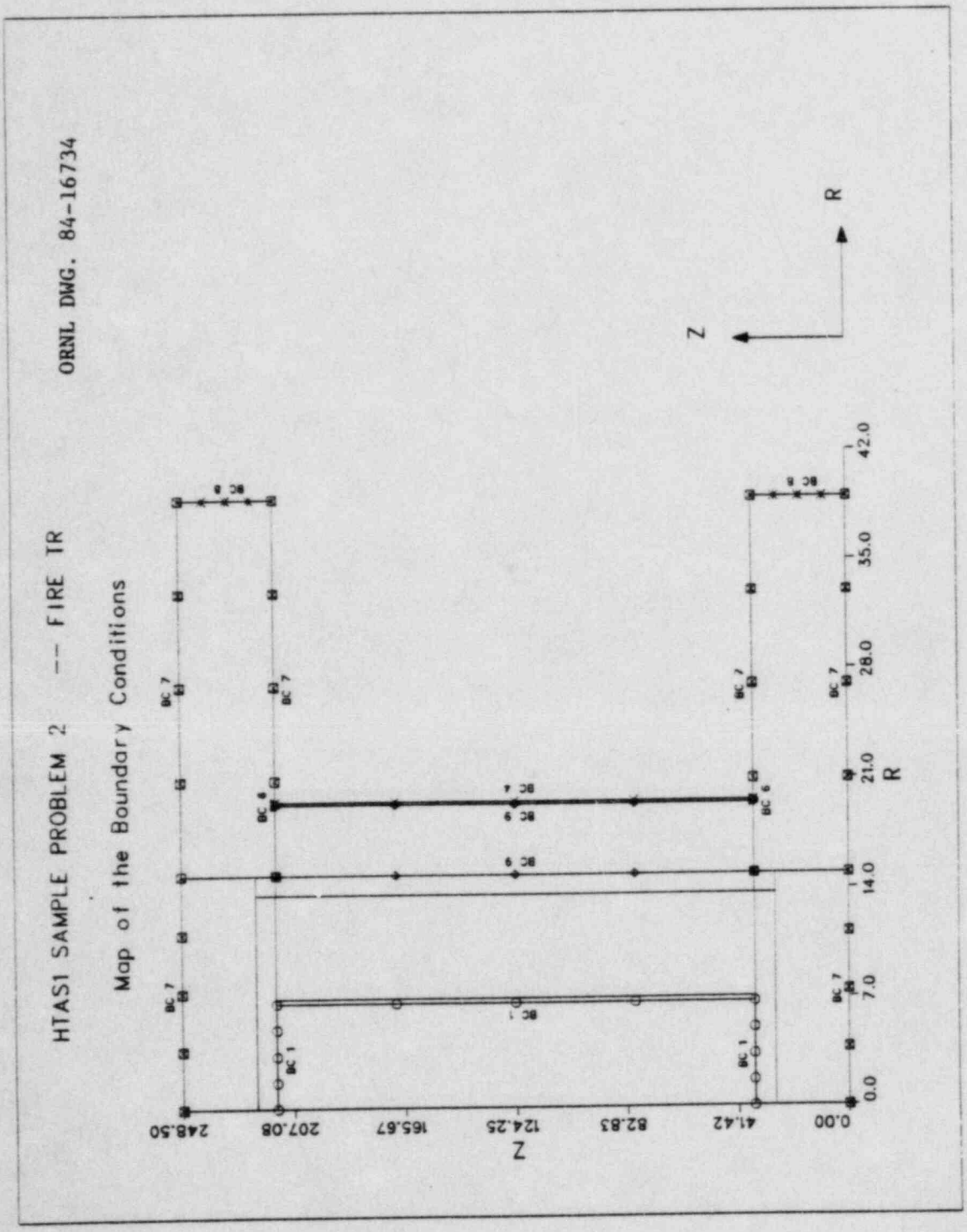


Fig. H1.5.2 (continued)

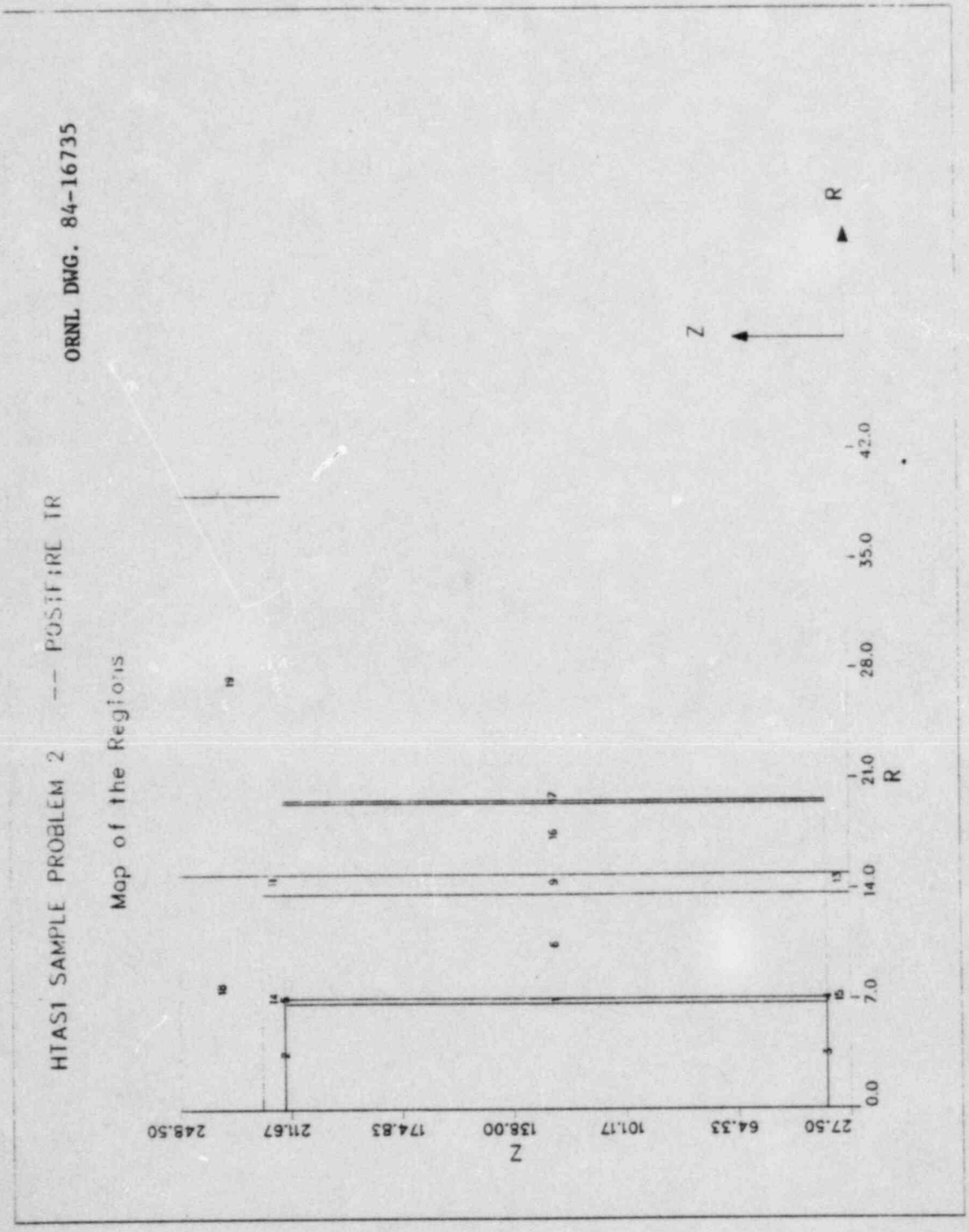


Fig. H1.5.2 (continued)

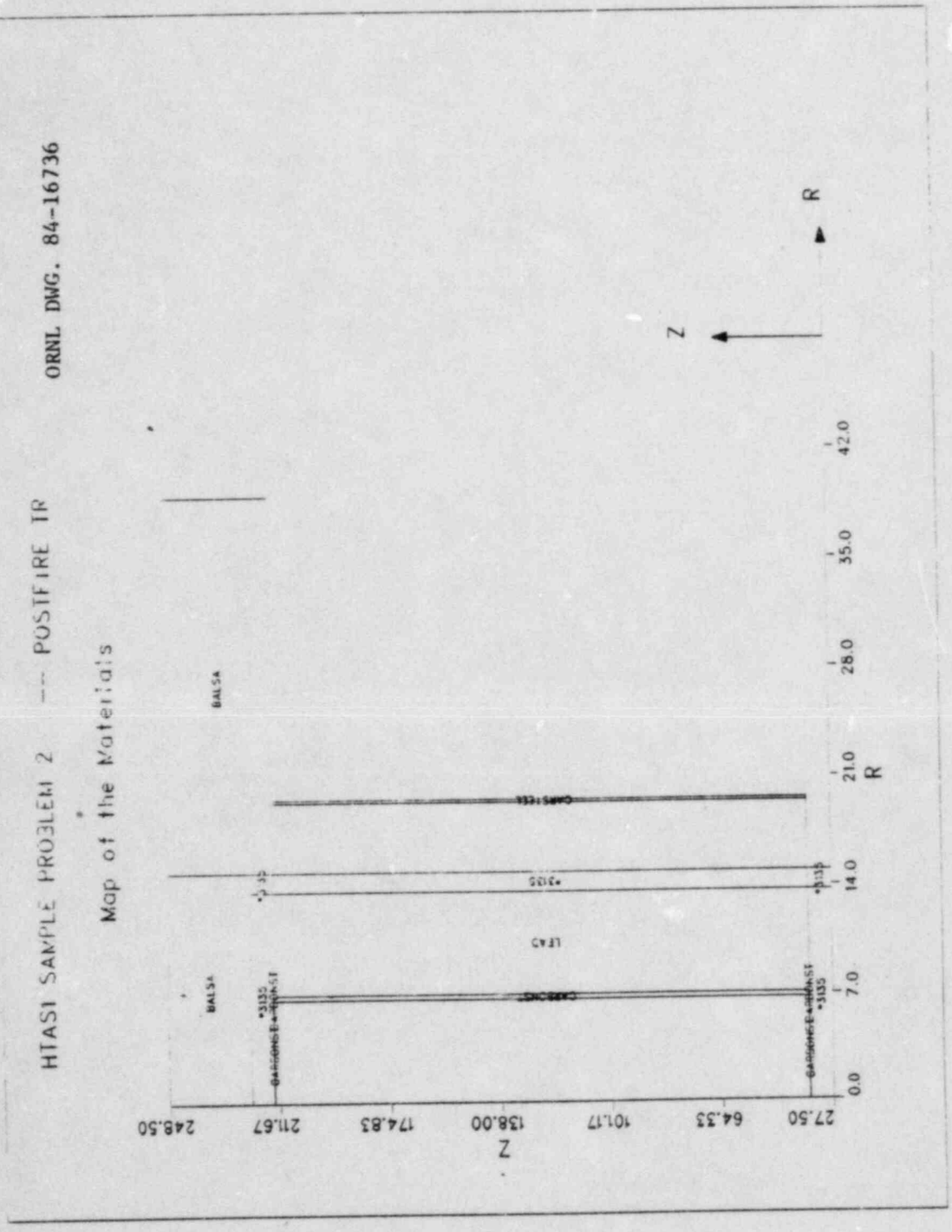


Fig. H1.5.2 (continued)

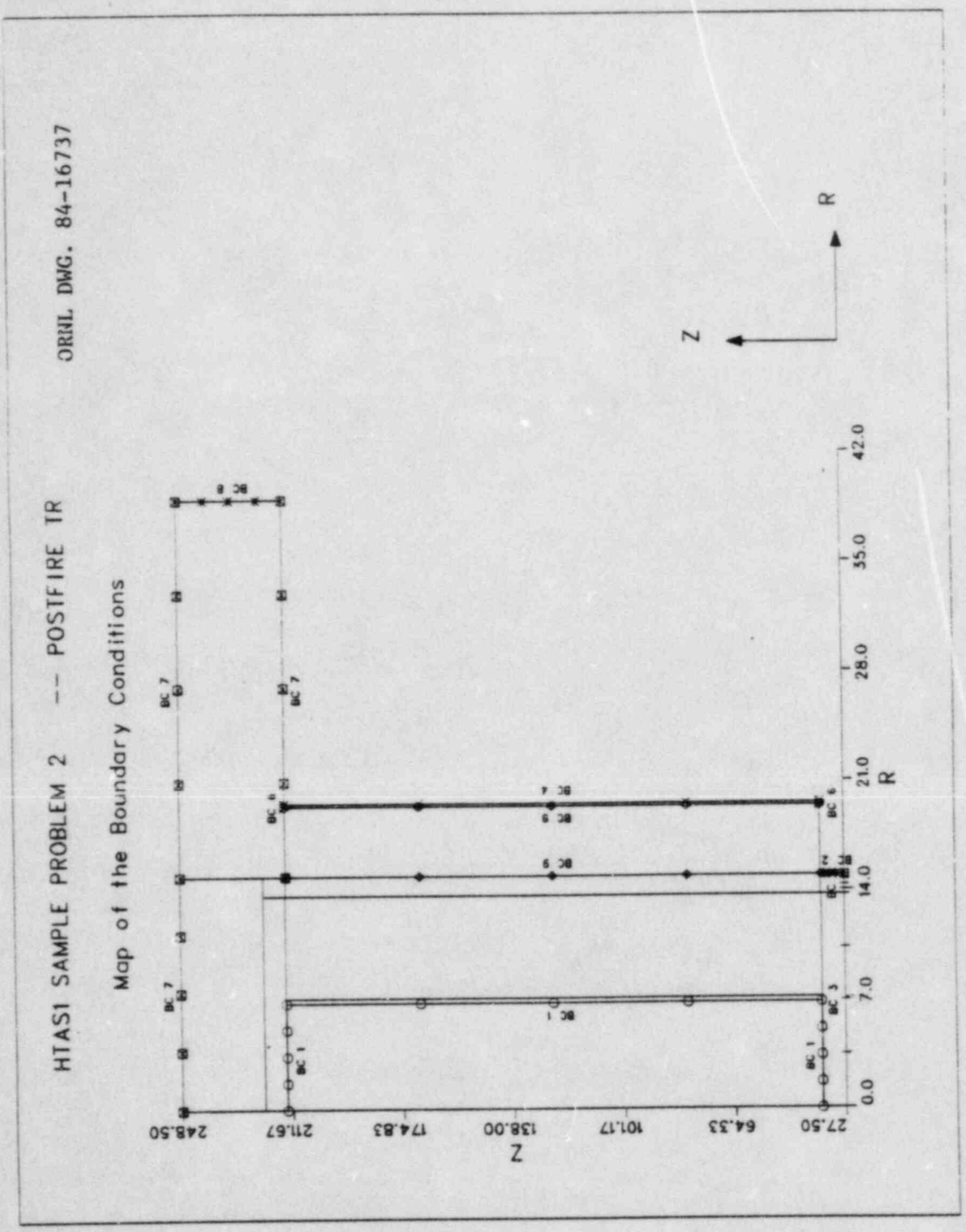


Fig. H1.5.2 (continued)

H1.6 REFERENCES

1. Handbook of Heat Transfer, ed. Warren M. Rohsenow and James P. Hartnett, McGraw-Hill, New York, 1973.

2. A. L. Edwards, A Compilation of Thermal Property Data for Computer Heat-Conduction Calculations, Lawrence Radiation Laboratory, University of California, Livermore, Calif., February 24, 1969 (UCRL-50589).

3. E. R. G. Eckert and R. M. Drake, Jr., Analysis of Heat and Mass Transfer, McGraw-Hill, New York, 1972.

4. R. E. Bolz and G. L. Tuve, ed., Handbook of Tables for Applied Engineering Science, CRC Press, Chemical Rubber Company, Cleveland, Ohio 44128.

5. IF300 Shipping Cask, Consolidated Safety Analysis Report, Nuclear Fuel and Services Division, General Electric, San Jose, Calif. 95125, October, 1979 (NEDO-10084-2).

6. J. H. Van Sant, Conduction Heat Transfer Solutions, Lawrence Livermore National Laboratory, University of California, Livermore, Calif., March, 1980 (UCRL-52863).

7. L. B. Shappert, et al., A Guide for the Design, Fabrication, and Operation of Shipping Casks for Nuclear Applications, Oak Ridge National Laboratory, Oak Ridge, Tenn., February, 1970 (ORNL/NSIC-68).

8. Part 71 of Title 10, Chapter 1, Code of Federal Regulations-Energy, Rules and Regulations, United States Nuclear Regulatory Commission, August 24, 1983.

9. W. H. McAdams, Heat Transmission, McGraw-Hill, New York, 1954.

10. Chen-Ya Lin, W. K. Mueller, and F. Landis, Natural Convection Heat Transfer in Long Horizontal Cylindrical Annuli, Trans. Am. Soc. Mech. Engrs., V. p. 976, 1961.

11. CSD Programmer's Notebook, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tenn., March 10, 1981.

12. DISSPLA User's Manual, Integrated Software Systems Corporation, San Diego, California, 1981.

APPENDIX H1.A. CONTROL CARDS FOR EXECUTING HTAS1 ON ONE OF THE OAK RIDGE IBM SYSTEMS.

The HTAS1 control module is part of the SCALE system at the computing centers at both Oak Ridge National Laboratory (ORNL) and Oak Ridge Gaseous Diffusion Plant (ORGDP). The code was compiled on the FORTRAN IV H Extended compiler with an Optimization Enhancement. This appendix discusses the job control language (JCL) which is required to run this version of the HTAS1 control module with HEATING6. The appendix includes discussion on how HTAS1 uses the CPU time limit to terminate the calculations and how to estimate the computer storage requirements for executing HTAS1. The Appendix contains examples of the JCL necessary to generate region plots and temperature profile plots at ORNL and ORGDP. It also contains an example of the JCL necessary to save the final temperature distribution from an analytical sequence and to restart an analytical sequence. For additional information concerning JCL, the user is referred to the CSD Programmer's Notebook (ref. 11).

The control cards necessary to use this version of HTAS1 at ORNL are presented in Fig. H1.A.1. The first and second control cards in Fig. H1.A.1 are called the JOB statement and JOBPARM statement, respectively. The jobname field on the JOB statement has a format of UIDXXXXX where UID is the userid assigned to the user by Computer Sciences (CS), and the characters XXXXX may be any characters which make the job name unique. The accounting field (ccccc,IOnnnn) consists of the charge number and I/O limit. The field, ccccc, contains a four- or five-digit CS charge number. The I/O limit field, IOnnnn, is a keyword parameter in the job accounting field following the charge number where nnnn consists of one to four digits specifying the maximum number of input/output unit counts allowed for the job in thousands. The programmer-name field which may consist of up to 20 characters contains information such as the user's name and address which is necessary to route the job output back to the user. The CPU time limit field, TIME=(minutes,seconds), specifies the maximum amount of time allowed for the analytical sequence. The minutes field consists of zero to four digits (maximum, 1440) specifying the maximum number of minutes. The seconds field consists of zero to two digits (maximum, 59) specifying the maximum number of seconds. If seconds are not specified, the parentheses can be omitted and the TIME field will have the form, TIME=minutes. If minutes are not specified, the TIME field will have the form, TIME=(,seconds). On the JOBPARM statement, the line limit specification, LINES=mmm, indicates the maximum number of printed lines allowed for the job where mmm is one to three digits representing thousands of lines. The third card is called the EXEC statement, and the maximum region requirements for the job are determined by the REGION=xxxxK field where xxxx represents the maximum amount of core in K bytes.

The control cards necessary to generate region plots on an 11-inch Versatec plotter using this version of HTAS1 at ORNL are presented in Fig. H1.A.2. Plots may be generated on a 36-inch Versatec plotter at ORNL by omitting the REQ= parameter PAPER=600. Plots may be generated on a CALCOMP plotter at ORNL by replacing the REQ= parameters XV36.POP,PAPER=600 with the parameter XC92.POP.

```
//jobname JOB (ccccc,IOnnnn),'programmer name',TIME=(minutes,seconds)
//*
//* This JCL Runs the Basic HTAS1 Control Module at ORNL
//*
/*JOBPARM LINES=mmm
//STEP1 EXEC SCALE,REGION.GO=xxxxK
//GO.SYSIN DD *
#HTAS1
 (Enter HTAS1 Input Data Here)
/*
//
```

Fig. H1.A.1. Job control language for executing HTAS1 at ORNL.

```
//jobname JOB (ccccc,IOnnnn),'programmer name',TIME=(minutes,seconds)
//*
//* This JCL Runs the Basic HTAS1 Control Module at ORNL
//* Region Plots Are Generated on an 11-inch Versatec Plotter
//*
/*JOBPARM LINES=mmm
//STEP1 EXEC SCALE,REGION.GO=xxxxK
//GO.SYSIN DD *
#HTAS1
 (Enter HTAS1 Input Data Here)
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
//  SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=XV36.POP,PAPER=600
/*
//
```

Fig. H1.A.2. Job control language for executing HTAS1 at ORNL and generating region plots on an 11-inch Versatec plotter.

The control cards necessary to generate a compressed plot file containing temperature profile plots and route the file to the user's PDP-10 area using this version of HTAS1 at ORNL are presented in Fig. H1.A.3. The field ppppp,ppp should be replaced with the user's PDP-10 programmer number. The file is named uidxxx.PCH, where the characters uidxxx are the first six characters of the job name. The plot file may be sent from the user's PDP-10 area to various plotting devices on which the plots are generated. The temperature distribution plot data sets written by HEATING6 for the fire transient calculations and the postfire transient calculations are written to logical units 18 and 19, respectively. These logical units are incorporated into the SCALE procedure and satisfactory default parameters are provided. However, to save either of these data sets for subsequent use by HEATPLOT-S (see Sect. F13), the user can employ the following DD card format for storage on one of the semipermanent direct access storage devices at ORNL or ORGDP.

```
//GO.FTnnF001 DD DSN=T.uidcccc.xxxxxxxx,DISP=(NEW,CATLG),UNIT=SPDA
```

The control cards necessary to use this version of HTAS1 at ORGDP are presented in Fig. H1.A.4. The first control card is called the JOB statement, and the jobname field, jobname, charge number field, ccccc, and programmer-name field are similar to the corresponding fields on the JOB statement for ORNL. The second control card is called the CLASS statement. The CPU time limit field, CPU95=yyyz, specifies the maximum amount of time allowed for the analytical sequence where yyy is one to three digits representing the number of time units and z defines the time units by S for seconds, M for minutes, or H for hours. The line limit specification, LINES=mmm, indicates the maximum number of printed lines allowed for the job where mmm is one to three digits representing thousands of lines. The I/O limit specification, IO=nnn.n, consists of one to three digits specifying the maximum number of IO's in thousands with an optional fractional part representing hundreds of IO's. The region limit for the job is specified in the REGION=xxxxK field where xxxx represents the maximum amount of core in K bytes. The region requirements must also be entered on the third control card which is designated as the EXEC statement.

The control cards necessary to generate region plots on a CALCOMP plotter using this version of HTAS1 at ORGDP are presented in Fig. H1.A.5. The control cards necessary to generate temperature profile plots on a CALCOMP plotter using this version of HTAS1 at ORGDP are presented in Fig. H1.A.6.

The maximum CPU time allowed for the analytical sequence is entered on the JCL as indicated above. HTAS1 determines the amount of CPU time remaining in the request and passes it to HEATING6 for each thermal analysis in the analytical sequence. HEATING6 subtracts five seconds from this specified time as a safety factor. After each iteration or time step, a check is made by HEATING6 to determine if the elapsed CPU time exceeds the specified time. If so, HEATING6 attempts to complete all of the output functions and returns to HTAS1 telling it the calculations were terminated prematurely. If the option is active, HTAS1 saves the final temperature distribution calculated by HEATING6 and then returns to the SCALE driver with a request to terminate the calculations.

```
//jobname JOB (ccccc,IOnnnn),'SAVEppppp,pppH ',TIME=(minutes,seconds)
//*
//* This JCL Runs the Basic HTAS1 Control Module at ORNL
//* A Compressed Plot File Containing Temperature Profile Plots
//* is Routed to the User's PDP-10 Area
//*
/*JOBPARM LINES=mmm,CARDS=10000
/*ROUTE PUNCH RMT45
//STEP1 EXEC SCALE,REGION.GO=xxxxK
//GO.SYSIN DD *
#HTAS1
 (Enter HTAS1 Input Data Here)
/*
//GO.COMPOUT DD DISP=(NEW,PASS),DSN=&&HTAS1P,UNIT=SPDA,
//  SPACE=(TRK,(5,1),RLSE)
//STEP2 EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1 DD DSN=&&HTAS1P,DISP=(OLD,DELETE),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000)
//SYSUT2 DD SYSOUT=B,DCB=(RECFM=FB,BLKSIZE=80,LRECL=80)
//
```

Fig. H1.A.3. Job control language for executing HTAS1 at ORNL and storing a compressed plot file containing temperature profile plots in the user's PDP-10 area

```
//jobname JOB (ccccc),'programmer name',
//*
//* This JCL Runs the Basic HTAS1 Control Module at ORGDP
//*
//*CLASS CPU95=yyyz,LINES=mmm,IO=nnn.n,REGION=xxxxK
//STEP1 EXEC SCALE,REGION.GO=xxxxK
//GO.SYSIN DD *
#HTAS1
 (Enter HTAS1 Input Data Here)
/*
//
```

Fig. H1.A.4. Job control language for executing HTAS1 at ORGDP

```
//jobname JOB (ccccc),'programmer name',
//*
//* This JCL Runs the Basic HTAS1 Control Module at ORGDP
//* Region Plots Are Generated on a CALCOMP Plotter
//*
//*CLASS CPU95=yyyz,LINES=mmm,IO=nnn.n,REGION=xxxxK
//STEP1 EXEC SCALE,REGION.GO=xxxxK
//GO.SYSIN DD *
#HTAS1
 (Enter HTAS1 Input Data Here)
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
//  SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=KC92.POP
/*
//
```

Fig. H1.A.5. Job control language for executing HTAS1 at ORGDP and generating region plots on a CALCOMP plotter

```
//jobname JOB (ccccc),'programmer name',
//*
//* This JCL Runs the Basic HTAS1 Control Module at ORGDP
//* Temperature Profile Plots Are Generated on a CALCOMP Plotter
//*
//*CLASS CPU95=yyyz,LINES=mmm,IO=nnn.n,REGION=xxxxK
//STEP1 EXEC SCALE,REGION.GO=xxxxK
//GO.SYSIN DD *
#HTAS1
 (Enter HTAS1 Input Data Here)
/*
//GO.COMPOUT DD UNIT=IN2OU2,DISP=(NEW,KEEP),
//  SPACE=(TRK,(5,1),RLSE),DSN=T.uidccccc.uidxxxxx
//STEP2 EXEC SSPREQ,DSN='T.uidccccc.uidxxxxx'
//SSP.CATLGDSN DD DISP=(OLD,KEEP),VOL=REF=*.STEP1.GO.COMPOUT
//SYSIN DD *
REQ=KC92.POP
/*
//
```

Fig. H1.A.6. Job control language for executing HTAS1 at ORGDP and generating temperature profile plots on a CALCOMP plotter

The REGION requirements for executing HTAS1 vary during the course of the calculations with HEATING6 requiring the largest amount of computer storage. In general, each thermal calculation executed on HEATING6 requires a different amount of computer storage since the model is subject to change and since the numerical technique may vary from one calculation to the next. Usually, the maximum amount of computer storage is required for a steady-state solution using the direct solution technique. The algorithm to determine the amount of computer storage necessary to perform a specific thermal analysis on HEATING6 is quite detailed and is tedious to evaluate. To facilitate the estimation of the REGION size for executing a problem on HTAS1, an upper bound on the computer storage requirements may be determined from the equation

$$\text{REGION} = \{[88\text{MAXRFG} + 28\text{MAXZFG} + 28\text{MAXFGL} + (6 + 16\text{MAXPRS})\text{MAXTBL} + 8\text{MAXPRT} + \text{SIZE}]/1024 + 387\}\text{ K bytes}$$

where

MAXRFG = number of radial fine grid lines,
MAXZFG = number of axial fine grid lines,
MAXFGL = maximum (MAXRFG, MAXZFG),
MAXPRS = maximum number of ordered pairs per tabular function,
MAXTBL = number of tabular functions,
MAXPRT = number of printout times.

SIZE, which is a function of the numerical technique, should be calculated for each thermal analysis in the analytical sequence. The largest value should then be used in the above equation. For transient problems

$$\text{SIZE} = 206\ \text{MAXPTS}.$$

For steady-state problems using the direct solution technique,

$$\text{SIZE} = (158 + 8\text{MAXRFG})\text{MAXPTS}.$$

For steady-state problems using successive overrelaxation,

$$\text{SIZE} = 190\text{MAXPTS}.$$

All of the parameters in the above equations are calculated by HTAS1 and passed to HEATING6. MAXRFG can be determined by summing the number of radial increments on each zone data card and adding one. Likewise, MAXZFG can be determined by summing the number of axial increments on each zone data card and adding one. An upper bound on MAXPTS can be determined as the product of MAXRFG and MAXZFG. MXPRS, MXTBL, and MXPRT may be used as upper bounds for MAXPRS, MAXTBL and MAXPRT, respectively. By using the default values of 20, 21, and 50 for MXPRS, MXTBL, and MXPRT, respectively, Eq. (H1.A.1) may be further simplified to

$$\text{REGION} = \{[88\text{MAXRFG} + 28\text{MAXZFG} + 28\text{MAXFGL} + \text{SIZE}]/1024 + 394\}\text{K bytes}.$$

If it is anticipated that it will be necessary to restart an analytical sequence at a later date, the user can instruct HTAS1 to save its final temperature distribution by entering as the second entry on Card 2 (the one following the Title card) the logical unit number (JTPOUT) on which the data set will be written. A DD statement describing the output unit must appear before the

```
//GO.SYSIN DD *
```

DD statement. The following DD statement will store the data set on one of the semipermanent direct access storage devices at either ORNL or ORGDP.

```
//GO.FTnnF001 DD DSN=T.uidccccc.xxxxxxx,UNIT=SPDA,
// DISP=(NEW,CATLG),SPACE=(TRK,(ppp,ss),RLSE),
// DCB=(RECFM=VBS,BLKSIZE=1061)
```

The logical unit number must replace the field nn, e.g., 30 for logical unit number 30. In the DSN field ccccc must be replaced by a valid computer charge number (if the charge number is less than five digits, zeros must be used on the left to pad it to five digits), and the field xxxxxxx must be replaced by up to eight alphanumeric characters chosen by the user to make the data set name unique (the first must be alphabetic or national, i.e., @, #, $). In the SPACE operand, the field ppp must be replaced by one to three digits indicating the primary allocation of space in tracks, and the field ss must be replaced by one or two digits indicating the secondary allocation of space in tracks. In the majority of cases, ppp can be 5 and ss can be 1.

A data set containing the final temperature distribution from a previous execution of HTAS1 may be used to restart an analytical sequence as long as the nodal configuration from the two cases is compatible. One must instruct HTAS1 to read this initial temperature distribution by entering as the first entry on Card 2 (the one following the Title card) the logical unit number (JTPIN) from which the data set containing these temperatures will be read. A DD statement describing the input unit must appear before the

```
//GO.SYSIN DD *
```

DD statement. If the data set was cataloged when it was created as indicated in the example given above, the following DD statement can be used to identify it.

```
//GO.FTnnF001 DD DISP=SHR,DSN=T.uidccccc.xxxxxxx
```

where the logical unit number must replace the field nn, e.g., 31 for logical unit 31, and the DSN field T.uidccccc.xxxxxxx must be the data set name containing the temperature distribution as created during a previous execution of HTAS1.

If the user creates a data set on one of the semipermanent direct access storage devices at either ORNL or ORGDP, it must satisfy the naming

conventions as discussed in the example given above, it must be cataloged as indicated in the example, and it must not exceed 180 tracks of space. The data set will be retained for a short period of time based on the number of tracks of space allocated to it as indicated in the Table H1.A.1 (ref. 11). The data sets will be automatically deleted at the end of the retention period.

Table H1.A.1. SPDA Retention Times.

Data Set Size in 3330 Tracks	Maximum Retention Time
>180 tracks	NOT ALLOWED
121-180 tracks	1 week
81-120 tracks	2 weeks
41-80 tracks	3 weeks
16-40 tracks	4 weeks
1-15 tracks	5 weeks

APPENDIX H1.B

MICROFICHE OF COMPLETE OUTPUT FROM HTAS1 FOR SAMPLE PROBLEMS

Microfiche of the complete output from HTAS1 for all the sample problems are located at the end of the Appendices. Two microfiche are provided for the second sample problem. The microfiche labeled Sample Problem 2a contains output from the second sample problem when region plots are generated. The microfiche labeled Sample Problem 2b contains output from the second sample problem when the analytical sequence is performed.

NUREG/CR-0200
VOLUME 2
ORNL/NUREG/CSD-2/V2/R2

SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

REVISIONS AND ERRATA SHEETS FOR VOLUME 2: FUNCTIONAL MODULES

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
Under Interagency Agreement 40-549-75

F1.5. INPUT INSTRUCTIONS

The input to BONAMI-S uses the FIDO schemes described in Sect. M10.

Block 1

0$ Logical Unit Assignments [4]

1. MMT - input master library (Default = 23)
2. MWT - scratch for Bondarenko factors (Default = 24)
3. MSC - scratch device (Default = 18)
4. NEW - output master library (Default = 22)

1$ Case Description [6]

1. IGR - geometry
   - 0 homogeneous
   - 1 slab
   - 2 cylinder
   - 3 sphere
2. IZM - number of zones or material regions
3. MS - mixing table length. This is the total number of entries needed to describe the concentrations of all constituents in all mixtures in the problem.
4. IBL - cross section edit option (0/1 - no/yes)
5. IBR - Bondarenko factor edit option (0/1 - no/yes)
6. ISSOPT - Dancoff factor

ISSOPT	Option
0 or 1000	Homogeneous σ_e = 0.0
1 or 1001	Cylindrical Cell (Sauer's Method) Hex Lattice
2 or 1002	Cylindrical Cell (Sauer's Method) Square Lattice
3 or 1003	Symmetric Slab Cell
4 or 1004	Asymmetric Slab Cell
5 or 1005	Isolated Lump (Dancoff = 0.0)

6 or 1006	Cylindrical Cell (Bell Approximation)
7 or 1007	Symmetric Slab Cell (Bell Approximation)
8 or 1008	Cylindrical Cell (Sauer's Method) Hex Lattice w/clad
9 or 1009	Cylindrical Cell (Sauer's Method) Square Lattice w/clad

Note that when ISSOPT > 1000, the code will use potential scattering values in place of total cross section in determining σ_0 values and will not iterate.

2* Floating Point Constants [2]

1. EPS - convergence criteria for the Bondarenko iteration (Default = 0.001)

2. A - geometrical escape probability adjustment factor. See notes below on this parameter. (Default = 0.0)

T Terminate Data Block 1.

Data Block 2

3$ Mixture numbers in the mixing table [MS]

4$ Component (nuclide) identifiers in the mixing table [MS]

5* Concentrations (atoms/barn-cm) in the mixing table [MS]

6$ Mixtures by zone [IZM]

7* Outer radii (cm) by zone [IZM]

8* Temperature (K) by zone [IZM]

9* Extra cross section (cm^{-1}) by zone [IZM]

10$ New identifier for components in the mixing table [MS]

11$ Zone type identifiers 0 - fuel zone

1 - moderator zone

2 - cladding zone

12* Temperature (K) of the nuclide in a one-to-one correspondence with the mixing table arrays.

T Terminate Data Block 2

This concludes the input data required by BONAMI-S.

1. MMT identifications for nuclides from the master library on N1.

2. MWT identifications for nuclides from the working/weighted library on N2.

3. MXT identifications for nuclides from the working/weighted library on N3.

Within each group, the identifications can be in any order.

A facility is provided for creating duplicate sets for a particular nuclide. For example, a resonance nuclide might occur in different compositions, thereby requiring different resonance calculations. To create a new set, place a unique negative number in the identification string after the nuclide's identification which is to be duplicated. As many sets as needed can be had, merely by stringing out negative numbers. All subsequent references to the new sets of data should use a positive identification. The values MMT, MWT, and MXT must be increased to reflect the inclusion of these sets.

3* Resonance Data [IRES*15] (IRES>0)

For each resonance calculation, 15 items are required. These blocks of data are placed, one set after the other, in any order in the 3* array. Note that only one 3* array is input. Resonance calculations can only be performed on resonance nuclides from a master library. The 15 items are:

1. Nuclide's identification number.

2. Temperature (K).

3. Geometry for Nordheim treatment

   0 - homogeneous,

   1 - slab,

   2 - cylindrical,

   3 - spherical.

4. Absorber lump dimension in centimeters; i.e., the thickness of the slab or radius of a sphere or cylinder. A zero is used for the homogeneous case.

5. Dancoff correction factor. This factor corrects for shadowing effects due to the presence of other absorber lumps (e.g., fuel elements) in the neighborhood of the lump being calculated. It is zero for the homogeneous and the one lump case.

6. $(\sigma_m)_{eff}$, the effective moderator cross section per absorber atom.

$$(\sigma_m)_{eff} = \sigma p_o + \sigma_m + \sigma_e$$

where

σ_{p_o} = absorber potential scattering cross section (frequently used values can be found in Table F5.D.5),

σ_m = moderator cross section per absorber atom,

σ_e = effective escape cross section = $\frac{1}{\bar{\ell}Na}$

$\bar{\ell}$ = mean chord length in the lump. The factor, $\bar{\ell}$, is a function of the geometry and can be determined analytically for simple geometries. Monte Carlo routines are available which allow its calculation for complicated geometries.

This cross section is only used in the unresolved calculation and does not effect the Nordheim treatment.

7. N_a, absorber number density in the lump.

8. Treatment of absorber's contribution to the collision density.

   1 - Nordheim integral method.

   2 - narrow resonance approximation.

   3 - infinite mass approximation.

9. Atomic mass of the first moderator in the lump. The resonance routines have provisions for treating two moderators.

10. σ_{s_1}, first moderator scattering cross section per absorber atom.

$$\sigma_{s1} = N_1 \sigma_s \text{ (moderator 1)}/N_a,$$

where N_1 is the number density of the first moderator.

11. Treatment of the first moderator's contribution to the collision density.

   0 - no first moderator.

   1 - Nordheim's integral method.

   2 - asymptotic approximation.

12. Atomic mass of the second moderator in the lump.

13. σ_{s2}, the second moderator scattering cross section per absorber atom.

MT	Description
719	(n,p_c') cross section for continuum specifically not included in σ total (redundant, used for describing outgoing proton)
720	(n,d_0) cross section for ground state
721	(n,d_1) cross section for 1st excited state
722	(n,d_2) cross section for 2nd excited state
.	
.	
.	
738	(n,d_c) cross section for continuum excited state
739	(n,d_c') cross section for continuum specifically not included in σ total (redundant, used for describing outgoing deuteron)
740	(n,t_0) cross section for ground state
741	(n,t_1) cross section for 1st excited state
742	(n,t_2) cross section for 2nd excited state
.	
.	
.	
750	(n,t_c) cross section for continuum excited state
759	(n,t_c') cross section for continuum specifically not included in a σ total (redundant, used for describing outgoing triton)
760	$(n,^3He_0)$ cross section for ground state
761	$(n,^3He_1)$ cross section for 1st excited state
.	
.	
.	
778	$(n,^3He_c)$ cross section for continuum
779	$(n,^3He_c)$ cross section for continuum specifically not included in σ total (redundant, used for describing outgoing ^{3}He)
780	(n,α_0) cross section for ground state
781	(n,α_1) cross section for 1st excited state
.	
.	
.	
798	(n,α_c) cross section for continuum

MT	Description
799	(n,α_c') cross section for continuum specifically not included in σ_T (redundant, used to describe outgoing α)
800-999	(to be assigned)

The following processes are not "ENDF/B" processes but may be added by various preprocessing codes in the SCALE and/or AMPX systems.

1000	transport cross section based on the outscatter approximation. See Section F3.2.30
1001	transport cross section based on the inscatter approximation. See Section F3.2.30
1007	Thermal scattering matrix
1008	Elastic part of thermal scattering matrix
1018	Fission spectrum
1021	These data, which are generated by a preprocessing code for resonance nuclides, are the "bodies" of the $\ell = 0$ resonances for the (n,γ) reaction, weighted over some specified weighting option which is usually $1/E$ in the resonance region. (The "body" of a resonance is the energy range over which the Nordheim integral treatment is used in the NITAWL module to calculate resonance self-shielded data.) When a resonance calculation is <u>not</u> specified for a resonance nuclide in NITAWL, the MT = 1021 and MT = 102 data on the master library are summed and identified as the (n,γ) cross section - MT 102 data - in any cross-section library produced by NITAWL. (Note that when a $1/E$ weighting function is used in the resonance range in XLACS-2, the sum of the MT = 1021 and MT = 102 data are, by definition, the infinite dilution cross sections.) When a resonance calculation <u>is</u> triggered in NITAWL, the MT = 1021 data are not used
1022	Fission "body" of the $\ell = 0$ resonance data which is analogous to the MT = 1021 data
1023	Elastic scattering "body" of the $\ell = 0$ resonance data which is analogous to the MT = 1021 data
1099	Group integral of the weight function
1452	Product of ν times the fission cross section
1500-1501	Same as 1000, 1001 except for gamma-ray cross sections
1527	Gamma-ray energy absorption coefficient factors.

The various sets of dose conversion factors and their corresponding MT numbers can be found in Table F4.4.1.

a / m n

Card J: Format (A6,I5,I5,I5)

a. The word BLOCK.

l,m,n: Each block is located in the zone by three integers: *l*, m, and n. These specify the block as being the *l*th in the X direction, the mth in the Y direction, and the nth in the Z direction, within the given zone. The integers *l*, m, and n run from 1 to the maximum number of blocks in each direction.

a b

Card K: Format [A12,10(I5,A1)]

a. The word MEDIA.

b. A list of the media, sector by sector, in the block. As with other lists, a comma in the A1 field indicates that the list continues; its termination is indicated by the absence of the comma.

Card(s) K′: Format [12(I5,A1)]

The continuation, if required, of the medium list.

a b

Card L: Format [A12,10(I5,A1)]

a. The word SURFACES.

b. A list of the quadric surfaces appearing in the block. Commas in the A1 field indicate that the list continues; a blank indicates the end of the list. The numbers appearing in this list derive from the order in which the surfaces are mathematically described on card R, which will be described later in the input.

Cards L: Format [12(I5,A1)]

The continuation, if needed, of the list begun on card L.

a b

Card M: Format (A4,A2,2213/913)

a. The word SECTOR.

b. The designation of each sector with reference to its position relative to the quadric surfaces. For every sector in the block there must be a card M, which will have as many references as there are surfaces in the block. The status of the sector is listed according to the following key:

+1: The sector is on the positive side of the surface.

−1: The sector is on the negative side of the surface.

0: The surface is not needed in the definition of the sector. The order in which each reference to a quadric surface appears on each card M must correspond to the order in which the quadric surfaces are listed on card L.

If there is only one sector in a block, cards L and M should be omitted.

a

Card Q: Format (I5,11A6)

a. The total number of quadric surfaces in *the entire system*. The alphabetic data in the A6 fields is ignored by the code.

a b c

Card R: Format [4(E10.5,A5,A1)]

Each quadric surface is described by writing the quadratic function whose zeros define the surface, in a fixed field format resembling the normal manner of writing functions. Each term in the function is specified by:

a. The coefficient of the term.

b. May be XSQ, YSQ, ZSQ (used for x^2, y^2, and z^2), XZ, YX, YZ, XY, ZX, ZY, X, Y, Z, or blank.

c. A nonblank character in this field indicates the end of the function. The next function must start on a new card.

NUREG/CR-0200
Volume 2, Section F6
ORNL/NUREG/CSD-2/V2/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

COUPLE

SCALE SYSTEM MODULE TO PROCESS PROBLEM-DEPENDENT CROSS SECTIONS AND NEUTRON SPECTRAL DATA FOR ORIGEN-S ANALYSES

O. W. Hermann

Manuscript Completed: October 1981
Date Published: October 1981
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

The nuclides of class 4 include light elements which are not updated. As in the case of class 2, the cross sections applied are a function of THERM, RES and FAST. Normally, the user specifies that these factors be computed from the flux pertaining to one of the fuel nuclides. Thus, with either type of weighting option, results calculated for nuclides of class 4 reflect this inconsistency. Additionally, the differences given above for nuclides in class 3 apply similarly to class 4.

A few primary conclusions may be derived from the above discussion. Cell weighting, when generally applied to updating all types of nuclides, appears to produce more consistent or correct results in ORIGEN-S cases than those produced from the region weighting option. However, when the objective is mainly the updating and computation of fuel and fission product nuclides, with no calculations required for light elements, there are essentially minor or no differences in the results of the important nuclides due to the selection of type of weighting. Since most of the cross sections in the Card Image Library of ORIGEN-S and ORIGEN were taken from student references such as BNL-325, except for those computed from a fuel-pin lattice cell calculation, these are zone weighted cross sections according to the above definitions.

8. The LMFBR type of library may be updated properly with COUPLE. The Card Image Library for the LMFBR applies a difference rule: THERM, RES and FAST are set to "1.0". COUPLE maintains this rule. Cross sections which are updated in producing the AMPX Weighted Library are updated in the ORIGEN-S Binary Library, while all others remain unchanged.

F6.4.3 OTHER NUCLEAR DATA UPDATING

The updating capability of COUPLE is complete, in the respect that any data or value in any ORIGEN-S Binary Library may be changed through the use of card input (see Data Blocks 6-9 of input description, Subsection F6.5). These features are briefly presented in the following:

1. Any nontransition "nuclear data" may be updated. These data are defined with their "code variable names":

   DIS - Total decay constant.

   Q - Total decay energy released (MeV).

   FG - Fraction of energy Q from photon emission.

   AMPC - Radioactive concentration inhalation guide value, RCG_a.

   WMPC - Radioactive concentration ingestion guide value, RCG_w.

ABUND - Atom percent abundance of a naturally occurring isotope.

2. Photons/disintegration spectra of any nuclide in the photon energy group structure of the library may be updated. (However, a more automatic update from the Master Photon Library is done by ORIGEN-S.)

3. Neutron reaction cross sections of any nuclide for any transition may be updated by card input. Any neutron reaction is permitted, since the type of reaction is simply denoted by input of the target and product nuclides. Also, nonfission removal and fission cross sections may be updated by card input.

4. Fission yield data may be updated indirectly by using the yield fraction of the fission product times the fission cross section of the fissile nuclide as an updated transition. (A separate updating code may be more efficient for processing a complete update of fission yields.)

5. The user may substitute the fission yield data of one fissile isotope for that of another which has no yield data. While these yield data may not be correct, it may be an improvement over the alternative that effectively means all yields are zero for the isotope; and the substitution may, at least, produce more conservative results. This option may be requested when updating by card input, here, or in the "computed cross section updating" using an AMPX Weighted Library.

6. Decay constants and branching fractions may be updated. The product of branching fraction and proper decay constant is simply input for the proper transition between two nuclides to produce the branching fraction updating.

Since cross sections and decay constants can be updated elsewhere, this array allows the change of all other nuclide data except: the prompt spectra applied during spontaneous fission; and separate ν-values, which are updated only from those in the AMPX Weighted Working Library. The updated "nuclear data" is entered in arrays 46$ through 50*, as needed.

13$ Array - Number of fissile nuclides for which substituted fission product yield distributions are added to library. New yield data for nuclides already having yields is entered in Arrays 71$ through 73*. (1 entry)

NFISM - Number of nuclides for which the substitution mode of using "approximate" yield data is requested (0).

Also, requires data in 71$, 72$ and 73* Arrays.

15$ Array - Number of nuclides in the library for which neutron reaction cross sections are to be updated (3 entries).

LBU - Number of updated nuclides, by library (0).

1. Number in Light Element Library.
2. Number in Actinide Library.
3. Number in Fission Product Library, plus number of fission cross sections, plus number of fissile nuclide to fission product nuclide reaction transitions, plus NFISM (13$ Array).

The cross section data are entered in the 71$, 72$ and 73* Arrays, with all fission reactions entered at the end.

16$ Array - Number of nuclides in library for which decay transition constants (radioactive half lives) are to be updated (3 entries).

LBR - Number of updated nuclides, by library (0).

1. Number in Light Element Library.
2. Number in Actinide Library.
3. Number in Fission Product Library.

The radioactive decay constants, $\ln(2)/T_h$ (where T_h is half-life in in seconds), and nuclides are entered in 81$, 82$ and 83* Arrays. (Also, see note on DIS in 47* Array).

T - Data Block 6 terminator.

Data Block 7. - New "Nuclear Data" (Nontransitions), required if LBB non-zero in 12$ Array. The entire Data Block 7, including the "T" must be repeated for each nuclide; or, number of times that Data Block 7 is included in data equals LBB(1) + LBB(2) + LBB(3).

46$ Array - Nuclide definition (2 entries).

1. NUP - Nuclide ID Number.

2. LIB - Library containing nuclide.

   1, for Light Element Library.

   2, for Actinide Library.

   3, for Fission Product Library.

   Order of the data blocks for updating more than one nuclide requires successive values of LIB to never decrease, i.e., all for LIB = 1 in a group, etc. No special order is required of Nuclide ID Nos. within the library group.

47* Array - Nontransition Nuclear Data of Nuclide (8 entries).

1. DIS - Total decay constant. (Note, that updating a transition constant to a daughter nuclide in Data Block 9 causes the code to change DIS, the total decay constant of the parent nuclide, by the same amount. In such instances, do not change DIS here. It is necessary to update DIS only during certain types of data corrections, such as the case when the daughter is missing from the library.)

2. Q - Total energy released by decay as recoverable heat, in MeV.

3. FG - Fraction of energy Q that is from γ-rays (or X-rays).

4. AMPC - Radioactivity concentration inhalation guide, RCG_a.

5. WMPC - Radioactivity concentration ingestion guide, RCG_w.

   (Entry 6 applies to Light Elements, only.)

6. ABUND - Percent abundance of a naturally occurring isotope.

7. Not used presently.

8. Not used presently.

For each of these 8 entries the default is the "old" library value. Thus, skip (with "Ai" to the ith entry) to entries wanted. If a "0.0" is entered, it changes value to 0.0.

NUREG/CR-0200
Volume 2, Section F7
ORNL/NUREG/CSD-2/V2/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

ORIGEN-S
SCALE SYSTEM MODULE TO CALCULATE FUEL DEPLETION, ACTINIDE TRANSMUTATION, FISSION PRODUCT BUILDUP AND DECAY, AND ASSOCIATED RADIATION SOURCE TERMS

O. W. Hermann
R. M. Westfall

Manuscript Completed: November 1980
Date Published: October 1981
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN Nos. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

TABLE OF CONTENTS

FOREWORD F7.*v*

ACKNOWLEDGMENTS F7.*vii*

ABSTRACT F7.*ix*

F7.1. INTRODUCTION TO ORIGEN-S F7.1.1

F7.2. THE ANALYTICAL METHOD F7.2.1

F7.2.1 Statement of the Problem F7.2.1
F7.2.2 Solution with the Matrix Exponential Method F7.2.2
F7.2.3 Solution to the Nuclide Chain Equations F7.2.6
F7.2.4 The Nonhomogeneous Case F7.2.9
F7.2.5 Time Dependence of the Neutron Flux F7.2.10
F7.2.6 Spectral Dependence of the Neutron Cross Sections F7.2.13
F7.2.7 The Volterra Multiplicative Integral Solution F7.2.14
F7.2.8 **Neutron Source Strengths and Spectra** F7.2.27

F7.3. FEATURES AVAILABLE IN PROGRAM F7.3.1

F7.4. RANGE OF APPLICABILITY AND DISCUSSION OF UNCERTAINTY F7.4.1

F7.5. PROGRAM STRUCTURE – DESCRIPTION AND INTERRELATIONSHIP OF MAJOR SUBROUTINES F7.5.1

F7.6. ORIGEN-S INPUT DESCRIPTION F7.6.1

F7.6.1 The Input Structure F7.6.1
F7.6.2 Comment on Condensed and Detailed Input F7.6.2
F7.6.3 Nuclear Data Libraries F7.6.2
F7.6.4 Printed Output F7.6.3
F7.6.5 Clarifying Concentration Units to Use and Flux vs. Power F7.6.5
F7.6.6 Cases and Subcases F7.6.5
F7.6.7 Nuclide or Isotope Identification Number F7.6.6
F7.6.8 Format of Input Data F7.6.6
F7.6.9 Condensed ORIGEN-S Input Description F7.6.7
F7.6.10 Detailed ORIGEN-S Input Description F7.6.16
F7.6.11 Data Note A, Normalizing Spectra to One Fission F7.6.45
F7.6.12 Data Note B, Weight Factors: THERM, RES, FAST F7.6.47

F7.7. SAMPLE CASES F7.7.1

F7.8. I/O UNIT REQUIREMENTS F7.8.1

F7.9. REFERENCES F7.9.1

ABSTRACT

ORIGEN-S computes time-dependent concentrations and source terms of a large number of isotopes, which are simultaneously generated or depleted through neutronic transmutation, fission, radioactive decay, input feed rates and physical or chemical removal rates. The calculations may pertain to fuel irradiation within nuclear reactors, or the storage, management, transportation or subsequent chemical processing of removed fuel elements. The matrix exponential expansion model of the ORIGEN code is unaltered in ORIGEN-S. Essentially all features of ORIGEN were retained, expanded or supplemented within new computations.

The primary objective of ORIGEN-S, as requested by the Nuclear Regulatory Commission, is that the calculations may utilize the multi-energy-group cross sections from any currently processed standardized ENDF/B data base. This purpose has been implemented through the prior execution of codes within either the SCALE System or the AMPX System, developed at the Oak Ridge National Laboratory. These codes compute flux-weighted cross sections, simulating conditions within any given reactor fuel assembly, and convert the data into a library that can be input to ORIGEN-S. Time-dependent libraries may be produced, reflecting fuel composition variations during irradiation. Some of the other objectives included in ORIGEN-S are: the convenience of free-form input; the flexible dimensioning of storage to avoid size restrictions on libraries or problems; the computation of gamma source spectra in any requested energy-group-structure, applying a more complete standardized data base; the determination of neutron absorption rates for all nuclides; and the integration of fission product energies and sources over any decay interval, by applying the Volterra multiplicative integral method.

Presented in the document are: detailed and condensed input instructions, model theory, features available, range of applicability, brief subroutine descriptions, sample input, and I/O requirements. The present revisions to this document reflect two major modifications to the code. The computation of neutron source spectra, as used in the SAS2 control module, has been added. Also, the calculation of neutron source strengths has been significantly improved. The other major change is the addition to the types of data that may be saved for later use. These data can be input and used by PLORIGEN (Section F15), a new code for plotting ORIGEN-S results.

ORIGEN-S is a functional module in the SCALE system and will be one of the modules invoked in the SAS2 control module, presently being developed, or may be applied as a "stand alone" program. It can be used in nuclear reactor and processing plant design studies, radiation safety analyses, and environmental assessments.

F7.2.8 Neutron Source Strengths and Spectra

The neutron source computed by ORIGEN-S includes that produced from both spontaneous fission and (α,n) reactions of heavy nuclides.

The major part of the neutron source is produced from spontaneous fission of the heavy nuclides. Data required to compute the neutron production rate from this process include the spontaneous fission half-life, the average neutron yield per spontaneous fission, ν_{sf}, and the concentration for each contributing nuclide. Spontaneous fission half-lives for the more significant nuclides are those from the updated[23] ORIGEN-S card-image actinide nuclear data library (Sect. M6). These half-lives are taken from ENSDF[12,13] and from Kocher's recent compilation of decay data,[24] both of which contain evaluated measured data. For several less important nuclides, unmeasured half-lives are taken from Ref. 25. These data were estimated with a correlation between measured data and so-called fissility parameters.[26] The ν_{sf} data are taken from Ref. 25. Measured values are available for 21 nuclides, including the most significant. An equation, derived[25] to compute ν_{sf}, produces values which are within two experimental standard deviations for all except three nuclides. This equation is applied for nuclides that do not have measured data.

A significant neutron source is produced from $^{17}O(\alpha,n)$ and $^{18}O(\alpha,n)$ reactions in the UO_2 and other oxygen compounds of the spent fuel. Thin target cross sections for these reactions and alpha stopping power data may be applied to compute neutron yields of the fuel material. Measurements[22] of thin target cross sections for the $^{17}O(\alpha,n)$ and $^{18}O(\alpha,n)$ reactions produced improvement over earlier data.[27,28] Additionally, thick target energy-dependent (α,n) yields for $^{238}U^{NAT}O_2$ were computed,[22] having estimated accuracies within 10%. ORIGEN-S applies these yield data from Ref. 22 to weighted energy averages of alpha energy-intensity data[24] of all nuclides except for those reported[10] for ^{214}Bi, ^{241}Pu and ^{249}Bk, which have very small alpha branching fractions. Decay constants and alpha decay branching fractions are required to compute the (α,n) source. The decay constants are provided directly from the ORIGEN-S Library. However, since alpha decay branching fractions are not supplied directly from an ORIGEN-S Library, the data are provided in the Block Data COMMON/SPECDT/. The data were copied from the current version (updated in 1981-1982) of the ORIGEN-S card image nuclear data library (Sect. M6). Also, since alpha energies are not explicitly included in ORIGEN-S libraries, the weighted averages of the most current (1981) evaluation[24] are also supplied in COMMON/SPECDT/. All of the nuclide yields and associated data can be edited during the execution.

The isotopes ^{242}Cm and ^{244}Cm characteristically produce all except a few percent of the spontaneous fission and (α,n) neutron source in spent PWR fuel over a 10-year decay time. The next largest contribution is usually from the (α,n) reaction of alphas from ^{238}Pu, which is approximately 1 to 2% of the source. Neutron energy spectra of both the spontaneous fission and (α,n) reactions have been determined for the curium isotopes[29,30] and ^{238}Pu.[31] The measured spontaneous fission neutron spectrum of ^{244}Cm was found to be quite similar to that from ^{235}U and ^{252}Cf. Thus, the spectrum for ^{242}Cm was computed[29] from these measurements. The (α,n) neutron spectra were determined by extrapolating the neutron spectrum from Po-α-O source measurements[32] to the alpha energies of ^{242}Cm, ^{244}Cm, and ^{238}Pu. The energy distribution of the spontaneous fission neutron spectrum is computed from the spectra for ^{242}Cm and ^{244}Cm described above, using the calculated concentrations of those two isotopes. This spectrum is then renormalized to include the total neutron source from all spontaneously fissioning isotopes. A similar calculation, using the data for all three isotopes, is performed for the (α,n) neutron spectrum. The spectra are collapsed from the energy-group-structure of the data to either the group-structure input or that of the SCALE library specified by input. The procedure assumes uniform distribution within each group and simply sums the quantities based upon energy fractions common to both groups in the two group-structures. The total neutron source spectrum is then computed as the sum of the spontaneous fission and (α,n) spectra.

f. cubic meters of air containing nuclide quantity to produce density equal to Radiation Concentration Guide (RCG) limit for air.[16]

g. cubic meters of water to equal RCG for water.[16]

h. neutron sources from α, n reaction.

i. neutron sources from spontaneous fission.

j. total gamma source spectra in photons/sec, MeV/sec, or MeV/watt-sec of burnup.

k. neutron source spectra from α,n plus spontaneous fission in neutrons/sec.

15. In general, only tables in the units specified are listed and these are divided into only those groups (see 11.) selected. These may be listed by either nuclide or element. Quantities below the cut-off for the unit may be deleted.

16. The gamma photon release rate spectra may be computed using three different photon constant data bases. The most recent data base[11] has improved quality and can be periodically updated.

17. An energy group structure may be input for producing gamma source spectra. The sources may be punched or saved as an output file. The sources from the three libraries or groups (see 11.) may be combined.

18. The photon data base in the "nuclear data library" may be updated from improved constants in the other data bases.

19. Principal photon sources, by nuclide, may be listed by energy group from the fission products. The percent of source used to select nuclides may be controlled by input.

20. A computation model[17] has been applied to the fission products which produces both time-dependent decay concentrations and their time integral over the time step.

21. Starting concentrations may be punched. Also the totals of many of the tables listed may be punched.

22. All concentrations (g-atoms) at any or all time steps may be saved on a binary tape. A restart feature allows a case to start with any of the saved values.

23. The core storage required for executing the program automatically expands and contracts to fit the problem.

24. All input data, normally punched on cards, may be input from a binary written interface – allowing easy recycle into and out of the program.

25. The removable thermal energy per fission, which is dependent on time and concentration, is computed for each time step by default. Also, an input option allows users to select 200 MeV/fission.

26. There is a table of contents of page numbers starting each subcase.

27. **Substantially improved data have been applied in calculating neutron sources, along with new models for α,n reactions and the energy-dependent neutron spectra (Sect. F7.2.8).**

A version of ORIGEN-S was modified by Williams[19] to perform the "adjoint" computation. The code, named ORIGEN-A, is used for sensitivity analyses, determining the importance of the uncertainty of one nuclide's cross section upon the uncertainty in the ORIGEN-S model computation of the concentration of another nuclide.

F7.4. RANGE OF APPLICABILITY AND DISCUSSION OF UNCERTAINTY

There are two main sources of uncertainty in the results computed by ORIGEN-S. One of these sources arises from the nature of the techniques for removing transition matrix elements that have absolute magnitudes greater than the prescribed matrix norm. The other deviation in the computed concentrations is from the uncertainties in the data base, i.e., from cross sections, decay constants and branching ratios.

The computation of discharged material inventories from reactors is probably the primary function of ORIGEN-S. This function includes predictions of various properties and radiation sources during cooling periods of the reactor discharge, both in storage or within chemical processing stages. Therefore, discussion is directed towards the effect of uncertainties in results with regard to these intended applications.

In discussing the first source of uncertainty, we recall the theory in subsections F7.2.2 and F7.2.3. In the chain A $\rightarrow$ B $\rightarrow$ C, where A and C are long-lived nuclides and B is short-lived or the "queue of short-lived nuclides," B is removed from the chain and the reduced matrix contains an adjusted value in the element for A. Also, the "beginning-of-interval" concentration of C is augmented to properly reflect its increase due to B. The assumptions applied in the algorithm produce only approximately correct results. However, calculations from this method asymptotically approach the true values if smaller time intervals are used for the same time period, or as the computation progresses over longer time periods. In cases involving an irradiation period of one month to three years for an LWR, it becomes necessary, or more appropriate, to apply at least ten time steps to the period. This method usually applies properly to thermal flux levels of 10^{13} to 10^{14} neutrons per cm^2-sec. Such values of the flux level should not be regarded exactly as the upper limit for ten time steps. If higher flux levels are applied, the users should compare answers of cases in which the number of time steps are doubled, or more, and determine for themselves the condition for computing acceptable accuracies. Usually the uncertainties of the major isotopes are of primary importance, while there is less significance in others that are produced in trace quantities. A "sense of proportion" should be applied in establishing the proper set-up of the problem. Thus, the model uncertainty should be compared with that produced from the data base, as discussed below. Early in the decay period the isotopic distribution tends to have a more rapid decrease in some of the isotopes that had reached a generation-decay-equilibrium during irradiation. Later the distribution is dominated with the asymptotic build-up of stable and long-lived nuclides. Thus, the accuracy in the calculated isotopic distribution increases with decay time. It has been found to be effective to start the decay period with a week or month time step and increase each interval size by no more than a factor of 3.

Probably the largest inaccuracy in the results for most nuclides stems from uncertainties in the cross-section data. There are three factors which contribute to this uncertainty: (1) the standard deviations in the measurements of the cross sections; (2) computational approximations in the

processing of cross sections for the representative conditions of the reactor; and (3) the fact that the time linearity assumption of the ORIGEN-S model is not completely valid. The experimental data is continually being improved. In addition to the initial libraries by Bell,[1] there have been improved libraries produced at ORNL through projects by Croff, et al.[20] (for the LWR fueled with U and Pu), and by Williams and Raman[21] (for the EBR-II) and various others. The SAS2 and SAS3 Control Modules in the SCALE System apply a model which simulates reactor conditions in producing cross sections and allows a large number of time-dependent libraries to be used by ORIGEN-S, all in a single computer job. The cross-section uncertainty due to factors (1) and (2) clearly dominates the error source in comparison to that from the model in factor (3) for typical reactor cases.

A large quantity of data other than cross sections is contained in the "nuclear data libraries." Considerations pertinent to this data are listed.

1. The calculation can require a large amount of fission product yield data. The yields differ for each fissile isotope. Also, yields are available to ORIGEN-S for only five fissile nuclides per reactor library as seen in the table is Subsection F7.6.3. The user should be aware that no fission products will be contributed by other fissile nuclides. Also, for a flux exposure of a small fraction of a second, the fission product energy release computed during the first few seconds of cooling is significantly smaller than the experimentally observed value. This may reflect difficulties in measuring yields of short-lived nuclides that are near the beginning of the chain.

2. The gamma source spectra have been greatly improved from that produced by the initial libraries, first by applying the ENDF/B-IV data, and recently, by applying the Master Photon Library.[11-15]

3. Caution is required in decay cases starting with a single nuclide A, in the chain discussed above, since the decay constant ratios may be such that daughters of C are overpredicted.

4. Since ORIGEN-S is a "point model" code, the computed k-infinities can be much less accurate than those from many neutronics codes. A cell-weighted cross-section library for a complete reactor core is required to produce an accurate value of k_∞. The relative changes in their time-dependent values can be useful, but should be interpreted properly.

NSERCH Searches NZ and NPT for index of current nuclide from ENDF B File.

DPRSEC Computes disintegrations per second of current nuclide.

LIBPUN Punches or writes either on unit NPUN the BCD library, or on unit MPHOT (NXTR) the binary library, which is produced from the ENDF/B data base.

CLOSE, CONTIN, LIST and SKPSCT – Special standard ENDF/B-IV file processing routines, for purposes such as reading the next nuclide data list and skipping part of the data in a proper manner.

Routine Name: SRCALC

Calling Routines: XORIGN via NSPEC

Routines Called: CLPS, YIELDN, SUBSTI

Functions: NSPEC sets array pointers, reads ORIGEN-S spent-fuel concentrations, obtains more complete spontaneous fission decay constants in SUBSTI, and calls SRCALC for the remainder of the neutron-source-spectra computation. The SRCALC coding follows the procedure described above (Sect. F7.2.8). Source strengths from major isotopes and the total are obtained for spontaneous fission first, and then for (α,n) reactions, computed by a quadratic data fit in YIELDN. The fine-group spectra are made in the arrays FINESP and FINEAL for the above interactions, respectively, by weighting according to source magnitude. They are collapsed to the energy-group-structure input or that of the SCALE library in CLPS and added together for the final neutron source spectrum.

Fission product yields are included in the libraries for fissile isotopes which appear to be the most significant. The card image libraries contain five isotopes with these yields for each reactor library. These are given in Table F7.6.1. The yields for those not footnoted are from thermal-neutron-induced fissions.

TABLE F7.6.1. FISSILE ISOTOPES WHICH HAVE FISSION PRODUCT YIELDS

Reactor	Isotopes, in the Order Yields are in Library				
HTGR	^{233}U	^{235}U	^{232}Th*	^{238}U*	^{239}Pu
LWR	^{233}U	^{235}U	^{241}Pu**	^{238}U*	^{239}Pu
LMFBR	^{241}Pu**	^{235}U*	^{240}Pu**	^{238}U*	^{239}Pu*
MSBR	^{233}U	^{235}U	^{232}Th*	^{238}U*	^{239}Pu

*Yield is from fission-spectrum-energy neutrons.

**The ^{239}Pu yields were substituted for these isotopes.

F7.6.4 Printed Output

A large quantity of printed output is available from ORIGEN-S. A total of more than 160 different types of tables presently may be printed, with possible options to print about 150 tables in a single subcase. Also, there are error or warning messages that may be printed. In general, the user may control the suppression or print of each table, with a few exceptions. The tables (later referred to by Group Nos.) are divided into the following groups:

1. Concentration results during decay or non-irradiation, given either by quantity or converted to other units (in seven types of units).
2. Concentration results during reactor irradiation, by quantity (in one to three types of units as specified by requested options).
3. The k_∞, relative total neutron production and absorption results, and fractional neutron absorption by nuclide results.
4. Edits of the six data sets of the "card image" nuclear data library.

5. Photon rate and gamma-energy rate spectra for each library, in fixed energy group structure, tables for neutron production from spontaneous fission and α-n reactions, and neutron production rate spectra of the actinides. The 12 tables in this group may be obtained during the decay period only. They are suppressed or printed as a group. The reference to "photon table results" or "photon tables" in the input description means all tables of this group.

6. Photon rate and gamma-energy rate spectra of the sum of the nuclides from all three libraries or the spectra of the sum of the nuclides from a single library, at requested time intervals. The user may supply an energy group structure, or, in some options, use the default. Three gamma line data bases are available: (1) The ORIGEN-S Photon Library (see Sect. M6.3); (2) ENDF/B file data for fission products only (see Sect. M6.6); and (3) the ORNL Master Photon Data Base (also see Sect. M6.7), which partly uses ENDF/B data, but principally applies data from the Evaluated Nuclear Structure Data File (ENSDF)[12-15] at ORNL (supplemented with other photon data such as bremsstrahlung). The spectra in this group are termed "special gamma spectra" in the input description.

7. Edits of photon data for each nuclide while producing new photon libraries, e.g., for an updated ORIGEN-S Library produced from the ORNL Master Photon Data Base or the ENDF data base.

8. Other variations to the spectra of Group 6. One option produces spectra in which only the nuclides having line data are included. Delayed decay spectra of the actinides may be obtained, when special libraries have been made.

9. Integration over the time step intervals for decay period, only, of the fission product nuclides and spectra. These "time integration" results may apply to all decay period fission product tables, which are included in the groups described above. This option simply adds "integral" table results to the output after all of the regular "instantaneous" results are printed for the first five groups. The spectra of Group 6 are only "integral" results. Except for MeV/Fission units of Group 6, the units printed for "integral results" should be multiplied by seconds.

The results or edits in the above groups are separated into tables pertaining to each of the three libraries, except where specified otherwise. The majority of the tables may be suppressed or printed individually. Results in the first three groups, usually, are given for all elapsed times requested in the input (see 60** array in Data Block 6).

Note that there is a print limitation. Output results for Group Nos. 1, 2, 3, 5 and 9 are subject to a fixed format, which does not permit the number of time steps printed to exceed 10 for each subcase. The user may apply more than 10 time steps when print is not requested or by simply using more subcases.

F7.6.9 Condensed ORIGEN-S Input Description

NOTE: **Default values** are given in **parentheses**. (Assume data required unless a condition is specified.) Also, while only a single * or $ is shown with each array identifier, be careful to use the **doubles**, ** and $$, if the **free-field** form of input is **intended**.

DATA BLOCK 1 - New Case.

0$ Data set unit numbers used in case, sometimes required. These are used for opening data sets in allocating case storage. Where units are not used, or the defaults apply, unit numbers should **not** be entered. Most variables are referred to in later input data. (12 entries)

1. KOUT — 5th entry in 3$ array. (6)
2. NDUM — For supplementary FIDO edits. (13)
3. NPUN — 16th entry in 56$ array. (0)
4. NDSETB — 1st entry in 3$ array, if binary library. (21)
5. NDSETF — 1st entry in 3$ array, if formatted lib..y. (0)
6. NVERTR — 32nd entry in 3$ array, if NVERT is positive. (0)
7. NVERTW — 32nd entry in 3$ array, if NVERT is negative. (0)
8. NDFB — 3rd entry in 81$ array, if binary library. (0)
9. NDFF — 3rd entry in 81$ array, if formatted library. (0).
10. LDSET — 2nd entry in 81$ array. (0)
11. NXTR — Binary photon library unit no., required if LNGAM = 6, 11, 12, 13, 14, or 15, (1st entry in 81$ array). May use for read write of concentrations. See "Detailed Input". (0)
12. NDISK — Scratch (binary) data set unit number always required. (11)

1$ New case or blend option trigger. (1 entry)

NOBLND 0/1/N - job terminates/ no blending/ N streams blended. (1)

T Block terminator.

DATA BLOCK 2 - New Library.

TITLEL Title printed in library edit. Format (20A4)

2* Required if NOBLND > 1. (NOBLND entries)

FACT Fraction of each blended stream.

3$ **Library Constants. See Sect. M6, e.g., for library sizes. (33 entries)**

1. NDSET Library unit number. Six unit numbers in 10$ array, if NDSET= -20. (27)
2. NOLIB Number of reactor libraries in NDSET. (4)
3. NTYPE 0/1/>1 card image/binary/future type library. (0)
4. NGRP **-N/0/N neutron spectra group structure from unit N/not used/from 84*. (-82)**
5. KOUT Printed output unit number. (6)
6. MPCTAB 0/1 air, water hazards requested/not requested. (0)
7. INPT 0/1 read photon library from NDSET/cards. (0)
8. IR 1/0 print transition matrix/do not print. (0)
9. LPU 0/N read all actinide data from NDSET/N actinides from cards. (0)

10-15. NN1,NN2,NN3,NN4,NN5,NN6:
NNI 1/0 print Ith library table/do not print. (0)

16. NN7 Trigger for concentration units in 74* array (0):
0/1/2/3 units in gram-atoms/wt. ppm/grams/atom ppm.
17. NN8 -1/0, suppress/print of library prologue. (-1)
18. ITMAX Total number of library nuclides. (850)
19. ILMAX Number of library 1 nuclides. (500)
20. IAMAX Number of library 2 nuclides. (120)
21. IFMAX Number of library 3 nuclides. (500)
22. IZMAX Number of off-diagonal non-zero matrix elements. (3000)
23. NREACT Maximum number of reactions for any nuclide. (7)
24. NFISO Number of fissile isotopes producing fission products. (5)

25. NELEM — Maximum atomic number in calculation. (99)

26. NMO — Month in date of library.

27. NDAY — Day in date.

28. NYR — Year in date.

29. NENAC — Number of actinide photon energy groups. (18)

30. NENLE — Number of light element photon groups. (12)

31. NENFP — Number of fission product photon groups. (12)

32. NVERT — Integral Option Trigger.
0/−N/N not wanted/request integral option (writing unit N)/request integral option (reading unit N). (0)

33. NG — Number of energy groups, required if LNGAM ≠ 0 or 15 (1st entry in 81$ array). (0)

Note, skip option for entries 19, 20, 21, 29, 30 and 31: −1, skips library or photon part of library.

4* Library Constants, required if NTYPE = 0. (4 entries)

1. THERM — Ratio of thermal neutron cross section of 1/v absorber to 2200-m/sec cross section. (1.0)

2. RES — Resonance flux per unit lethargy/thermal flux. (1.0)

3. FAST — 1.45 times ratio of flux > 1 MeV/thermal flux. (1.0)

4. ERR — Truncation limit. (10^{-25})

5$ Library Position Number, required if NTYPE = 0. (1 entry)

NLIBE — Reactor Library position in NDSET (2):

1/2/3/4 for HTGR/LWR/LMFBR/MSBR Reactor

10$ Separate library unit numbers, if NDSET = −20 (6 entries).

54$ Special Options − not required. (12 entries)
(May include 54$ in Data Block 5)

JOPT — Triggers to obtain special option.

JOPT(I) N/0 request option/not requested. Entry positions (1) trigger the following: (1) compute k-infinity, etc., (2) punch concentrations, (3) punch total gamma sources of Library 1 and 3, (4) punch α-n and spontaneous fission sources, (5) punch combined gamma source, (6) punch totals of nuclide tables N of each library, (7) punch for element tables N, (8) not used, (9) use single fissile nuclide for computing γ-spectra/fission, (10) for adjoint-version of code, (11) not used, and (12) use 200 MeV/fission in place of computed energy per fission.

T Block terminator.

DATA BLOCK 3 – Actinide Nuclide cards required only if LPU > 0.

6$ (LPU entries)

NEWCX Nuclide ID Numbers of actinides read in by cards.

T Block terminator.

Extra Card Input: If LPU > 0, read LPU cards for cross section data of the NEWCX nuclides.

DATA BLOCK 4 – Photon Energy Group Structures.

First data array is required, always. See "Detailed Input " to use other arrays.

35$ Required. (1 entry).
0, if photon energy group structure and U-235 fission photon distribution is available for input library or input in 83* array.
1, if any group structures read in this data block. (0)

T Block terminator.

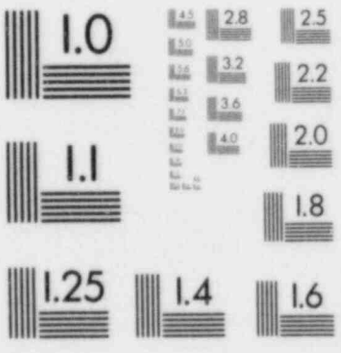
Extra Card Input: If INPT = 1, read in entire photon libraries. (Fixed format)

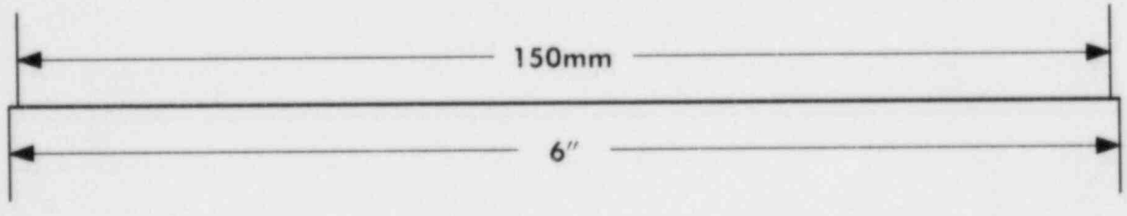
DATA BLOCK 5 – New Subcase with same library.

56$ Subcase Integer Control Constants. (20 entries)

1. MMN Number of irradiation intervals. (0)
2. MOUT Number of time intervals in subcase; (10)
   0, and MSUB = 0, job terminator;
   0, MSUB > 0, writes results of step MSUB **on unit NXTR;**
   -N, results written for steps MSUB to N, inclusive.
3. INDEX 0/1 power read in 58*/read flux in 59*. (0)
4. NTABLE 0, any Group 1 output printed, if not suppressed in 65$;
   1, print summary tables, per 65$. (0)

IMAGE EVALUATION
TEST TARGET (MT-3)





5. MSTAR — Time period for CUTOFF, photon sources. (1)
   0, cut-off feature not used, and JTO = 1 or 4.

6. NGO — Next problem indicator (1):
   −1, new concentrations and same library;
   0, new concentrations and library;
   1, old concentrations and same library;
   2, same as NGO = 1, but old subcase flux, power, and burnup are zeroed;
   3, old concentrations and new library.

7. MPROS — 0, for no continuous chemical processing.
   N, for N groups of elements in a processed group. (0)

8. NPROS — 0, if MPROS = 0;
   N, the maximum number of elements in a process group. (0)

9. MFEED — 0, for no continuous feed. (0)
   N, if N nuclides are continuously fed into system. (0)

10. MSUB — 0, for new concentrations (last NGO = −1 or 0);
    N, if old concentrations are from time interval N;
    −N, where N means the same, but with batch processing;
    N, and MOUT = 0, writes results of step N on **unit NXTR;**
    N, and MOUT = -M, requests steps N to M, inclusive. (0)

11. NTERM — Minimum number of exponential expansion terms. (21)

12. NSHRT — Maximum number of short-lived precursors in chain. (100)

13. NXCMP — Number of new nuclide or element concentrations. (0)
    Also, NXCMP can be used in restart option. See "Detailed Input."

14. NUNIT — 1/2/3/4/5/6 time (in 60*) in sec/min/hrs/days/yrs/other. (4)

15. NTI — 0, no title cards read at Data Block 6;
    1, TITLE read;
    2, BASIS read;
    3, both are read. (0)

16. NPUN — 0/N no punch/punches concentrations, etc., on unit N. (0)

17. JTO — 0, all concentration (or Groups 1, 2 and 5) tables output;
    1, all output results suppressed except per 54$ and 82$;
    2, may suppress only by 65$ or 66$;
    3, suppresses all except photon tables;
    4, suppresses photon tables and by 65$ or 66$. (2)

18. NUC — 0, suppresses "irradiation" nuclide tables;
1, suppressed only by 66§. (0)

19. NEL — Same as NUC for element tables. (0)

20. KBLEND — −1, if subcase uses total of previously blended streams;
0, no blended stream saved or used;
N, concentrations at time period, N, multiplied times next value of FACT are added as part of blend. (0)

57* Subcase Floating Point Constants. (5 entries)

1. TMO — Code uses TIME(1) − TMO as first interval in calculation. (0.0)

2. RHO — Total material density (gm./cm^3). Needed for tables in units of atoms/barn-cm., if NN7 = 1 or 3. (0.0)

3. CUT — An option for deleting elements in tables under 66§ control. If the concentration in units of 74* is less than CUT for any element, print for the element is deleted. CUT is source fraction cutoff in decay subcase. (0.0)

4. FRACPW — (TIME(MMN) − TMO)/(sum of time of all irradiation subcases). (1.0)

5. TCONST — If NUNIT = 6, number of seconds in the special time unit.

T — Block terminator.

DATA BLOCK 6 − Subcase Arrays and Titles

TITLE — Subcase Title, if NTI = 1 or 3. Format (20A4)

BASIS — Basis of Calculation, if NTI = 2 or 3. Format (10A4)

58* — Power required if INDEX = 0 and MMN > 0. (MMN entries)

POWER — Thermal power in time intervals of 60* (consecutive vaues or MMNth value cannot be zero.)

59* — Flux required if INDEX = 1 and MMN > 0. (MMN entries)

FLUX — Thermal flux in neutrons/cm.^{2}-sec, (consecutive values or MMNth values cannot be zero).

73$ ID of Nuclides - required for new case or if NXCMP > 0 and last NGO = -1 or 0. (NXCMP entries)

INUC1 Nuclide ID Number (if NEX1 = 1, 2, or 3) or Element ID Number (if NEX1 = 4) for the concentrations in 74*.

74* Concentrations required with 73$. (NXCMP entries)

XCOM1 Concentrations for nuclides and elements in 73$ in gram-atoms (NN7 = 0), wt. ppm (NN7 = 1), grams (NN7 = 2), or atom ppm (NN7 = 3).

**75$** Library Kinds - required with 73$. (NXCMP entries)

NEX1 1/2/3 where Nuclide ID No. applies to Library 1/2/3 of subsection H.6.3.
4, for element in Light Element Library.

76$ ID of Nuclides - required in MFEED > 0. (MFEED entries)

INUC2 Same as INUC1, applied to continuous feed option.

77* Feed Rates - required if MFEED > 0. (MFEED entries)

XCOM2 **Continuous feed rates or increases in feed rates (gram-atoms/sec/fuel unit).**

78$ Library Kind - required if MFEED > 0. (MFEED entries)

NEX2 Same convention as NEX1, applies to 76$

79* Element Fractions - required if MSUB < 0. (NELEM entries)

FREPRO Fractions of elements retained in batch chemical processing, where entry number = element atomic number.

81$ Gamma Source and Library Update Constants, required if LNGAM ≠ 0. (6 entries)

1. LNGAM 0/1/2/3/>3 no request/source from ENDF data/source from Master Photon Library data/source from ORIGEN-S data on NDSET/updates various libraries (see "Detailed Input"). (0)
2. LDSET Output gamma source unit no., if LNGAM = 1. (51)
3. NDF Input data base unit no., if LNGAM = 1 or 2. (60)
4. MNDF 2/1 - mode of NDF: card image/binary. (2)
5. N1MAX Maximum number of floating point words for ENDF input, or set by code for other input data. (3000)
6. N2MAX Maximum number of nuclide photon lines in data. (1000)

82$ Time Step Source Triggers - required if LNGAM = 1, 2, or 3. (MOUT entries) (0)

M(N) 0/1/2/4/5/6 - not wanted/request fission product source per fission/total source per sec/light element source/actinide source/fission product source, for Nth time interval (if LNGAM = 1, 2 or 3).

M = 4, 5, 6 are not applied and, also, M = 2 denotes fission product source from ENDF data, if LNGAM=1. (Other options set by M are explained in "Detailed Input.")

83* Gamma Energy Group Structure, in eV and descending order.

(NG+1 entries, 3$ array) (required, if LNGAM ≠ 0, 3 or 15).

84* **Neutron Energy Group structure, in eV and descending order - required if NGRP > 0. (NGRP + 1 entries, 3$ array)**

T Block terminator

IA One source title card for each 82$ request. A single library title card, if LNGAM > 3. Format (20A4)

FORMG Variable format of new library, if LNGAM =4 or 5. Requires a blank card, here, if LNGAM = 6, 7, 8, 9 or 10. Format (18A4)

FORMT Variable format - required if JOPT(2) > 0. Format (20A4). Requires 3 cards. The format used for concentrations punched on cards.

END OF CONDENSED INPUT INSTRUCTIONS

Be sure to add following SCALE control card:
END

F7.6.10 Detailed ORIGEN-S Input Description

Either fixed-field, free-field, or user-field type of FIDO input system should be used, except for title cards or supplemental libraries. Note that each parameter (or entry) is named and defined below in the order it appears in the data array. If applying fixed-field formats, data entries should be placed in successive data fields. If using free-field form, $$ or ** should be used with the array identifier and at least one space should be skipped between data entries. Each data block contains one or more arrays followed by the data block terminator letter "T", or "iT", where i is the Data Block Number.

** All available default values are in parentheses, following the data definitions **

DATA BLOCK 1 - Start of First Case

0$ Array Case data set unit numbers, sometimes required. (Not required if reading only a standard ORIGEN-S Library data set on NDSET, i.e., 21 through 28, and writing none.) Variable core storage is dynamically allocated from the region size requested, the program-instruction size and the IO buffer size as set from data in this array. A "0" excludes reserving buffer space. If in doubt, enter all unit numbers that will be used and allow sufficient region size. The user may skip this array in the input (defaults apply), unless one of the following is used: The Line Gamma Source Option (81$ array); the Integral Option (32nd in 3$ array); or, BLKSIZE > 2000 and BUFNO=2 (default) on the input library DD Card. Thus, often the 0$ array is skipped. When a SCALE library is denoted by NGRP (4th entry in 3$ array) and the standard blocksize of the data set is used, adequate space is allocated without an entry for NGRP in the 0$ array. (12 entries)

1. KOUT — KOUT, 5th entry in 3$ array. (6)
2. NDUM — Unit number used to print FIDO edits starting with that of 35$ array. (13)
3. NPUN — NPUN, 16th entry in 56$ array, usually, for punched cards. (Probably not required if BLKSIZE = 80). (0)
4. NDSETB — NDSET, 1st entry in 3$ array. Required only if BUFNO*BLKSIZE > 4000 (in JCL) for NDSET and NTYP = 1, 3rd entry in 3$ array. Not required when using standard JCL deck. When using data set denoted by NDSETF, storage may be saved by setting NDSETB = 0. (21)
5. NDSETF — Same as NDSETB except it applies to card image data set. (0)
6. NVERTR — NVERT, 32nd entry in 3$ array, when NVERT > 0. (Meaning, NVERT has been written.) (0)
7. NVERTW — Same as NVERTR except NVERT < 0, when writing the integral library. (0)
8. NDFB — NDF, when LNGAM ≠ 0 and MNDF = 1, which are in 81$ array. (0)
9. NDFF — Same as NDFB except MNDF = 2, for a card image gamma library. (0)
10. LDSET — LDSET, 2nd entry of 81$ array, and LNGAM ≠ 0. (0)

11. NXTR — Binary photon library or concentrations stored on this unit number, required if LNGAM = 6, 11, 12, 13, 14, or 15, or [10th] NSUB > 0 with [2nd] MOUT = 0. In using restart option (i.e., NXCMP ≤ 0 [13th]), set NXTR negative. Code will read unit number |NXTR|. (0)

12. NDISK — Unit number of a scratch data set needed in code. Not required to enter for standard JCL deck. (11)

1$ Array — Case control flag, always required. (1 entry)

ALL NEW CASES START HERE. (NGO = 0 and NOBLND = 1)

NOBLND — Blending and case control flag (1):

0, for normal termination of job;

1, for case with no blending;

N, if N streams are to be blended. (See KBLEND, 56$ array.)

T — Data block 1 terminator

DATA BLOCK 2 – Start of New Subcase Using New Nuclear Data (Reactor) Library. Start here only if last value of NGO = 0 or 3. (See NGO, 56$ array.)

TITLEL — Library Edit Title, always required. Read in the Format (20A4).

2* Array — Required if NOBLND > 1. (NOBLND number of entries)

FACT — Fraction of each blended stream used in "blending cases", where the specified fraction is taken times all concentrations in the stream. These fractions are in the order pertaining to the subcases that have the flag KBLEND > 0, in 56$ array. (Omit if NOBLND = 1.)

3$ Array — Library Integer Constants, always required, unless all defaults are proper. See override method below. (33 entries)

1. NDSET — **Library data set unit number. (See Sect. M6.)**

   -20, six separate card image library data set unit numbers are entered into 10$ array. (27)

2. NOLIB — Total number of reactor libraries in the data set, including those not used. Ignored, if NTYPE = 1. (4)

3. NTYPE — Trigger denoting type of reactor library format (0):

0, for card image version (as described in Sect. M6).

1, if same type of library in an unformatted or binary data set. Library size parameters read in first record.

>1 and <6, if original library plus additional data read and processed in a user-supplied subroutine MORDAT;

>5 and <10, if library obtained only from subroutine MORDAT.

4. NGRP — **Applies to energy-group structure of neutron spectra. Required, if** [17th] JTO = 2 or 3. (-82)

-N, if the energy-group structure is taken from the SCALE Master Library on unit no. N;

0, if neutron spectra not wanted;

N, if the energy-group structure for N groups is input in 84* array.

5. KOUT — Unit number of ORIGEN-S output results, which may differ from that of FIDO output. (FIDO uses unit given by 1st entry of 0$ array before libraries are written, and unit NDUM, 2nd entry in 0$, thereafter.) (6)

6. MPCTAB — Option flag for calculation of inhalation and water ingestion hazards (0):

0, if these results are requested for a subcase using the library;

1, if calculation not requested. (Also, see 65$ data for each subcase. Storage saved, if suppressed here. However, code always executes properly for the "0" default.)

7. INPT — Library photon data input option (0):

0, if photon library read from NDSET;

1, photon libraries (files 4, 5 and 6) are read from cards following Data Block 4.

8. IR — Debug output option (0):

1, the transition matrix elements derived from the libraries will be printed;

0, print is suppressed.

9. LPU — Actinide library data cross section input option (0):

   0, if entire actinide data from NDSET;

   N, if cross sections from cards override library data for "N" actinide nuclides. The Nuclide ID Numbers are entered in 6$ array and the new cross section cards follow Data Block 3.

10. NN1 — Output option for the Light Elements and Structural Materials Nuclear Data Library (0):

    1, table is printed;

    0, print is suppressed.

11. NN2 — 1, Actinide Nuclear Data Library is printed;

    0, print is suppressed. (0)

12. NN3 — 1, Fission Products Nuclear Data Library is printed;

    0, print is suppressed. (0)

13. NN4 — 1, Light Element Photon Library is printed;

    0, print is supressed. (0)

14. NN5 — 1, Actinide Photon Library is printed;

    0, print is suppressed. (0)

15. NN6 — 1, Fission Products Photon Library is printed;

    0, print is suppressed. (0)

16. NN7 — Indicator of type of units to be used for the concentrations in 74* array (for all subcases). (0)

    0, input units are gram-atoms;

    1, input units are weight parts per million;

    2, input units are grams;

    3, input units are atom parts per million.

17. NN8 — Library output option (-1)

    -1, the library history prologue and radioactive hazard library table are suppressed;

    0, these are printed.

18. ITMAX ⩾ total number of NDSET library nuclides. This is sum of nuclides in the three libraries referred to below (ILMAX, IAMAX, IFMAX), exclusive of additions using LPU option. **Library sizes are given in Sect. M6.2.**

    If NTYPE = 1, for binary library, many of these entries are not used. **See NOTE, at end of 3$ Array.** Also, see the same NOTE, if using small card image library. (850)

19. ILMAX Same meaning as ITMAX applied to Light Element Library;

    -1, if the library is not wanted. (500)

20. IAMAX Same meaning as ITMAX applied to Actinide Library;

    -1, if the library is not wanted. (120)

21. IFMAX Same meaning as ITMAX applied to Fission Product Library;

    -1, if the library is not wanted. (500)

    (If none of the nuclides of a library are required for a case, both region size and computer time are saved by the corresponding setting of ILMAX, IAMAX, or IFMAX to "-1". The "DD Cards" in the Job Control Language should not be changed.)

22. IZMAX Total number of reactions from one nuclide to another nuclide for all libraries. Code determines this value, as variable NON, for storage of permanent arrays so that it is all right to input a somewhat excessive value. Code prints message, if NON exceeds IZMAX, and states Nuclide ID Number. Note, the code adds 4*LPU to the input or default value. When in doubt, set IZMAX = 5*ITMAX for most libraries. (3000)

23. NREACT Maximum number of reactions (transitions due to neutron capture or decay) that any single nuclide in library will make. If too small, code increments NREACT by one and processes library again (for 5 times before terminating). Do not set less than 5 for usual libraries. Normally, the default is best. (7)

24. NFISO Number of fissionable isotopes in the reactor library for which fission product yields are given. These isotopes are "built in" the 25$ array for card image libraries. (5)

25. NELEM — Largest atomic number of nuclides in all libraries. Maximum allowable value of NELEM is 99. (It is suggested that user not truncate libraries with NELEM, except in unusual cases in which the user is certain that the library retains all valid chains. The storage saved is normally not very important. (99)

26. NMO — Month, for date nuclear data library was made. (0)

27. NDAY — Day of month of library. (0)

28. NYR — Year library was made. Date printed, only if NN8 = 0. (0)

29. NENAC — Number of energy groups in Actinide Photon Library. (18)

30. NENLE — Number of groups in Light Element Photon Library. (12)

31. NENFP — Number of groups in Fission Products Photon Library. (12) (If a photon library is not needed and the corresponding value of ILMAX, IAMAX or IFMAX was not set to "-1", the photon part of the library can be skipped by setting NENLE, NENAC and or NENFP to "-1".)

32. NVERT — Integral Option Trigger. (If used, requires entry for NVERTR or NVERTW in 0$ array). (0)

0, Integral Option not wanted;

±N, in addition to computing the regular "instantaneous" concentrations, the code computes results integrated over the interval of each time step using the Volterra multiplicative method, as applied by Lee[17] (for only the fission products during decay). The "instantaneous" concentration tables (for all units specified in 65$ array) are printed first, and following a message that integrals will be presented, the same type of fission product tables are printed containing the integral results. Note, that the units of these integral tables should be interpreted as the units shown times seconds.

-N, writes the utility sub-matrices data set on N. (Requires NVERTW, 7th in 0$ array.)

N, applies data set from N, using less CPU time. (Requires NVERTR, 6th in 0$ array. Never enter both NVERTW and NVERTR.)

The integral option is not operable with a fission product library that includes data for any transition that pertains to a change in mass number (A), e.g., the library containing branching due to delayed neutron decay (described in Sect. M6.7).

33. NG — Number of energy groups in 83* array, required if LNGAM ≠ 0 or 15 (1st entry of 81$ array.) (0)

NOTE on override of 3$ array by binary library data: This note applies to parameters ITMAX to NENFP (entries 18-31), inclusive. If NTYPE = 1, the correct values of these parameters are supplied from the binary library data set and the user may **not** override these parameters with one exception: a "-1" entry for NENAC, NENLE and/or NENFP will skip the use of the specified photon libraries. **Also**, if NTYPE = 0 and NDSET = 27 (for the small card image library), the defaults for ITMAX to NENFP are satisfactory.

4* Array Library Floating Point Constants, required if NTYPE = 0. (4 entries).

1. THERM*[1] — Ratio of the neutron cross section for a 1/v absorber with a population of neutrons having a Maxwell-Boltzmann distribution of energies, which is specified by an absolute temperature, T, to the cross section with 2200-m sec neutrons.[7] An equation for computing THERM is

$$\text{THERM} = \sqrt{\frac{\pi}{4}\frac{T_o}{T}}\,, \quad T_o = 293.16°K.$$

Default is 1.0. (See Data Note B in Section F7.6.12 on THERM, RES and FAST application and Section F7.2.6 for additional theory.)

2. RES — Ratio of the resonance flux per unit lethargy to the thermal neutron flux. (1.0)

3. FAST — 1.45 times the ratio of flux above 1 MeV to the thermal neutron flux. (1.0)

4. ERR — A truncation error limit below which the values computed by the code will be considered to be zero by the code. If a zero is entered, code changes to default value. Set ~ 10^{-30} if MeV fission spectrum or the Integral Method (32nd of 3$ array) is requested. (10^{-25})

---

*[1] THERM, RES, and FAST have meaning only for thermal reactors. For the LMFBR, the cross sections on NDSET are already spectrum averaged, and THERM, RES, and FAST are equal to 1.0 (Set by Code for LMFBR - do not enter.)

5$ Array Library ... ion Number, required if NTYPE = 0. (1 entry)

1. NLIBE — Integer for reactor library type for this problem (2):

   1, for HTGR Reactor;

   2, for LWR Reactor;

   3, for LMFBR Reactor;

   4, for MSBR Reactor;

   -2, for library processed by CASDAR Code.*; Then THERM, RES, and FAST are set equal to 1.0, 0.0, and 1.0, respectively, within the code.

10$ Array Six library unit numbers, if NDSET = -20. (6 entries)

Often this data array is not required, since the total library will be contained in six sequential files which all have the same unit number.

1. Light Element Nuclear Data unit number.
2. Actinide Nuclear Data unit number.
3. Fission Product Nuclear Data unit number
4. Light Element Photon Data unit number
5. Actinide Photon Data unit number
6. Fission Product Photon Data unit number.

54$ Array Special Options, not required. (12 entries) (May also include this array in Data Block 5).

JOPT(I) ≠ 0, request option I, as defined below for position "I".

= 0, option I not requested. (At start, "0" is the default.)

---

*G. W. Cunningham, Computer Sciences Division (ORNL), private communication. (Special Light Element Libraries.)

DATA BLOCK 4 - Photon Energy Group Structures and Flag.

Almost always, the user should set LFLAG = 0, skip the other data and terminate this Data Block with a T. Data here does **not** pertain to the making or use of new Photon Libraries per 81$ array. After Photon Library was produced by **user-supplied** data, **then**, this Data Block would be needed.

35$ Array Always required. (1 entry)

LFLAG 0, if no more data read in this block and case uses default values. These are correct **for ORIGEN-S Photon Libraries (Sect. M6).**

1, if more data read in this block. **(See Section M6.3 for explanation of data used here.)**(0)

Warning: **Never** read photon data here unless a corresponding library has been made. (0)

36* Array Optional. (NENAC + 1 entries)

EACTGP Photon energy group structure boundaries for actinide nuclides (MeV, in ascending order). Used in printed table column of photon tables.

37* Array Optional. (NENLE + 1 entries)

EGROUP Photon energy group structure for light element nuclides (MeV, in ascending order). Used in printed table column.

38* Array Optional. (NENAC entries)

SFGAMA **Photons per fission for U-235 in the group structure given by EACTGP (no longer used).**

39* Array Optional. (NENFP + 1 entries)

EFPGRP Photon energy group structure for fission product nuclides (MeV, in ascending order). Used in printed table column.

Note, that any new group structure given in this data block (for printing new group structures by average energy) must be exactly the same as the group structure used in making the photon library.

T Data Block 4 terminator

Extra Card Input:

If INPT = 1 (7th entry in 3$ array) the entire photon library (files 4, 5, and 6) are read here. **Formats are specified in Sect. M6.2.3.**

DATA BLOCK 5 - Subcase Control Data

This is the first block when starting a new problem with old nuclear data library. (Start subcase here only if last value of NGO = -1, 1, or 2, in 56$ data.)

54$ Array May be included (see Data Block 2).

56$ Array Subcase Integer Control Constants, always required. (20 entries)

1. MMN — Number of time intervals during irradiation period (0);

   0, for post-irradiation or decay-only subcase.

2. MOUT — Total number of time intervals in this subcase (10);

   0, for job termination (subcase starting with 56$ array);

   0, and MSUB > 0, the concentrations at last time step MSUB are written on unit no. NXTR (in 0$ array). This "dummy" subcase gives method for saving results in a binary data set. Only 56$ is read, with NGO = 1 plus the "T". Next 56$ is for regular, or another dummy, subcase;

   -N, and MSUB ⩾ 0, the results in g-atoms for steps MSUB through N, inclusive, are written on unit no. NXTR.

3. INDEX — 0, reactor specific thermal power input, 58* array (0);

   1, flux will be input in 59* array. (Not used in decay subcase.)

4. NTABLE — 0, all isotopes and all times will be given in decay output, unless suppressed with flag in 65$ array or (17th entry of 56$ array) JTO (0);

   1, only summaries of most important isotopes at alternate time periods are output.

5. MSTAR — Output suppression and photon source indicator. Code omits printing isotopes where values are below a cutoff (61* array) for time period MSTAR and principal photon sources are printed for time period MSTAR.

   0, cutoff feature not used and JTO = 1 or 4, which suppresses photon tables. (1)

6. NGO — Flag to indicate the type of the next subcase and data block the user must start using for new data (1):

   -1, if next problem has new initial concentrations, or has blend of old concentrations, and has same nuclear data library. Next subcase begins with Data Block 5.

0, if next problem has new initial concentrations and a new library. If NOBLND = 1, next case begins with Data Block 1, 1$ array. If "blending case" not finished, subcase begins with Data Block 2. Never repeat 0$ array data except after "END" and another "=ORIGENS" cards.

1, for a "continuation" problem in which concentrations found in present calculation and same library will be used. Next subcase will begin with Data Block 5.

2, for same conditions as NGO = 1 (the "continuation" problem), except that flux, power and accumulated burnup from prior subcases are set to zero for use in obtaining averages and accumulated burnup of next subcase. Note that otherwise the flux or power is weighted by FRACPW in 57* array.

3, for a "continuation" problem in which concentrations found in present calculation and a new library will be used. Next subcase begins with Data Block 2.

7. MPROS — A "continuous chemical processing" option (0):

0, for no processing;

N, where N groups of nuclides are continuously removed by chemical processing during reactor exposure period, only, and each group includes those elements with the same removal constant. See 62*, 63$, and 64$ arrays.

8. NPROS — Used with MPROS (0):

0, for no continuous chemical processing;

N, where N is the maximum number of elements in any of the groups with same removal constant. See 64$ array.

9. MFEED — A "continuous feed" option (0):

0, for no continuous feed;

N, where N nuclides are continuously fed into system as in a fluid fuel reactor. See 76$, 77* and 78$ arrays.

10. MSUB — 0, if new initial concentrations used in this problem (0);

N, for a "continuation" subcase where the initial concentrations are taken from the time period N of the last subcase;

-N, where time period N provides concentrations. However, specified factors of given elements are removed through batch processing. See 79* array for specifying the factors, if NLIBE ≠ 1. Special "built-in" factors are used, if NLIBE = 1.

N, and MOUT = 0, writes results of step "N" on unit no. NXTR.

Use MOUT = -M to request results for steps MSUB through M, inclusive, with MSUB = 0 permitted to denote the initial concentrations to subcase. See [2nd] MOUT for details.

11. NTERM — Minimum number of terms to be allowed in series expansion of matrix exponential, needed to assure convergence for long nuclide chains. (21)

12. NSHRT — Maximum number of short-lived precursors to a long-lived isotope in any decay chain. Suggest default. (100)

13. NXCMP — Number of nuclides or elements for which new concentrations are used as initial conditions to this problem.

See 73$, 74* and 75$ arrays. (0)

Always required for first subcase or if last subcase had a value of NGO = -1 or 0. Also, NXCMP can be used to start with the concentrations made on data set NXTR in a previous job, where MOUT and MSUB (of 56$ array) are input to properly write the concentrations from one or more steps. Set NXCMP ≤ 0 in this subcase. Those from the first step saved are used in this subcase, if NXCMP = 0. The set of concentrations written in the |NXCMP| position (or, the |NXCMP| time a set was saved) on NXTR are used, if NXCMP < 0. Then, omit 73$, 74* and 75$ arrays.

14. NUNIT — Trigger specifying units of the times read in 60* array (4):

1, for time in seconds;

2, for minutes;

3, for hours;

4, for days;

5, for years;

6, if other units requested. This requires (1) a card for TUNIT near beginning of Data Block 6, and (2) a value for TCONST (5th entry in 57* array).

74* Array Required with 73$ array. (NXCMP entries)

XCOM1 The concentration for the nuclide or element at the corresponding entry in 73$ array. If NN7 = 0, units of XCOM1 are gram-atoms per unit of fuel used in 58* (if power is input). If NN7 = 1, units are weight parts per million (wt ppm). If NN7 = 2, units are grams. If NN7 = 3, units are atom ppm.

75$ Array Required with 73$ array. (NXCMP entries)

NEX1 1, if corresponding entry in 73$ array is in Light Element Library;

2, for Actinide Library;

3, for Fission Product Library;

4, for element when using relative abundance in Light Element Library.

76$ Array Required if [9th] MFEED > 0. (MFEED entries)

INUC2 Follows same convention as INUC1 in 73$ array, but applies to nuclide with a "continuous feed rate".

77* Array Required with 76$ array. (MFEED entries)

XCOM2 **Continuous feed rate (or, change in feed rate) in gram-atoms per second per unit of fuel for nuclide at corresponding entry in 76$ array. Units are changed, as for XCOM1, if NN7 ≠ 0. A special rule is used if this option was also requested in the previous subcase; then, the data in XCOM2 are added to the previous rates or rates are unchanged, if XCOM2 is skipped.**

78$ Array Required with 76$ array. (MFEED entries)

NEX2 Follows same convention as NEX1 in 75$ array but applies to 76$ array.

79* Array Required if [10th] MSUB < 0. (NELEM entries)

FREPRO The fraction of the concentration of an element that is retained in the stream being used in problem after "batch. chemical processing." Entry number 1 applies to hydrogen and entry number N applies to element of atomic number = N. (Sixth entry is for carbon, and 92nd entry is for uranium). If element not present, a number must still be entered, although it has no effect. A conservative rule is to enter "1.0" for an element if none of it is known to be removed. (0.0)

81$ Gamma Decay Source or Library Update Constants (6 entries) - required for computing either the "special gamma source spectra" listed in Group 6 of Section F7.6.4, or updated photon libraries. The spectra obtained in the "photon table results" (of Group 5) are in a fixed group structure, apply to each library separately, are computed only during a decay subcase, and may use a less accurate data base. However, the more specialized spectra obtained with the option here have the following attributes:

1. Any energy group structure may be applied.
2. The user may select nuclide photon spectra from one of three different data bases.
3. The Master Photon Library data base[11,12] may be more frequently and easily updated.
4. The spectra, except for that from ENDF/B data base,[2,9] can be obtained either for the sum of nuclides in each library, separately, or for the sum of nuclides in all three libraries.
5. The decay spectra may be computed during **either** an **irradiation** or a **decay** subcase. The spectra during decay periods may be applied to spent fuel shielding problems, while the spectra during irradiation represents the part of the gamma energy in the reactor contributed by decaying nuclides. (Note that an irradiation period spectrum **does not** represent the **total** gamma rays **released** in the reactor, which should include, also, the instantaneous gamma rays associated with fission and gamma emission during neutron capture.)

Gamma decay sources are requested by setting LNGAM to 1, 2 or 3 in this data array, to select the desired data base. Other options should be set in the 82$ array. Also, the 83* array group structure, (LNGAM ≠ 3) NG in the 3$ array, and title cards are required. The production of updated libraries are requested with LNGAM > 3.

1. LNGAM Type of procedure and data base requested. (Note, some of the procedures require an entry for either NDFB or NDFF, 8th or 9th entry in 0$ array - never both)

   0, none requested, or omit 81$, 82$ and 83* arrays.

   1, applies the ENDF/B (currently, ENDF B-IV) data base[2] to **fission products (see Sect. M6.6). This procedure produces the following:**

   (a) printed fission product decay spectra as specified by 82$ and 83* arrays.

0, no source requested for the time step.

1, requires LNGAM = 1, 2 or 3, producing a special fission product spectrum for the time step. The units of printed spectra are MeV/fission and photons/fission. Do not use this option unless the ORIGEN case was set up to force the units to be correct (usually for one fissile isotope). See Data Note A in Section F7.6.11.

2, invokes the procedure requested by LNGAM = 1, 2 or 3, which produces a gamma source spectrum for the Nth time steps. The units of the printed spectra are MeV/second and photons/second. This source production option is more general than that resulting from M = 1 and the spectrum from any irradiation or decay time step may be requested. There is a difference, depending upon LNGAM, in the number of nuclides included in the source spectra; for LNGAM = 1, using the ENDF/B data base, the source pertains to the sum of all fission products, only; and, for LNGAM = 2 or 3, using either the Master Photon Data Base or the ORIGEN-S Photon Libraries, the source pertains to the sum of nuclides in all three libraries. Sources for separate libraries may be requested with other values of M. There are two different treatments used to account for the gamma energy of nuclides which have no spectrum data included in the data base:

(a) For LNGAM = 1, the spectrum is increased by one photon/disintegration in the group that includes the total gamma energy of the nuclide having no spectrum data.

(b) For LNGAM = 2 or 3, the ratio is computed which equals the sum of the gamma energy of all nuclides divided by the sum of the gamma energy of only those nuclides having photon data. After the total spectrum of nuclides having data is computed, it is renormalized by multiplying by the above energy ratio. The energy is conserved, with the uncertain fraction of the spectrum having the same distribution as that produced from all nuclides that have spectrum data. The total energies are printed for the sum from all nuclides both with and without spectrum data.

Note of warning in using LNGAM = 3. This is a request for summing the spectra computed from the three separate ORIGEN-S libraries. The request **implies** that the user is aware that the group structures of the three ORIGEN-S Photon Libraries are compatible (either the same, or one is a subset of another and the maximum boundaries are equal).

3. produces the gamma spectra of the decay of actinide neutron captures from data in a special library. This option is restricted to a very limited use and cannot be made generally available until the ENDF/B-V data file is processed. Similar results can now be computed from the Master Photon Data Base.

4. produces gamma source for only the nuclides included in the Light Element Library. It may be requested only for LNGAM = 2 or 3.

5. produces Actinide Library gamma source for LNGAM = 2 or 3.

6. produces Fission Product Library gamma source for LNGAM = 2 or 3. (Important Note: The user may not change to different non-zero values of M at later time steps.)

20 + M, or 21, triggers the printout lines stating that the Master Photon Data Base contains, for a given nuclide, gamma ray intensity data at an energy outside the range of the requested group structure. The message begins with "Photon could not be binned". First, for LNGAM = 2, determine (from the above), the value of M desired and add 20. Set to 21, if LNGAM > 3. The option applies only to cases using the Master Photon Data Base, or, LNGAM = 2 or LNGAM >3. (Note there are pages of lines written, if energy range does not start near zero; thus, when running several cases, the user may not find it desirable to use this trigger except for a single trial case.)

−1, is the same as M = 1 except that only the nuclides with gamma-line data are used in the LNGAM procedure.

−2, −4, −5, and −6, produce gamma decay source spectra computed from only those nuclides which have spectrum data. This allows the user to add in other data for those nuclides having no spectrum data.

83* Gamma Energy Group Structure, in eV, for spectra or library updates requested if LNGAM ≠ 0, 3 or 15. Boundaries are in descending order. (NG + 1 entries, 33rd in 3$ array).

The energy group structures which correspond to that of input ORIGEN Photon Libraries (on NDSET) are the defaults obtained by skipping this array, after the proper input of NG (33rd in 3$ array). NG = 12 obtains the Light Element and Fission Product structures, while NG = 18 obtains that for the actinides.

84* Neutron Energy Group Structure, in eV, for neutron spectra requested by JTO = 2 or 3 and NGRP > 0. Boundaries are in descending order. (NGRP + 1 entries, 4th in 3$ array)

T Data Block 6 Terminator

F7.7. SAMPLE CASES

Case 1 PWR Fuel Exposure and Decay

A scoping study is desired for a PWR, using the smaller initial ORIGEN Library. Let the reactor operate at a power of 30 Megawatts continually for 1100 days. List the complete inventory in gram-atoms per metric ton of uranium during both the exposure period and decay period. Also, list the results for the decay period in curies, total thermal power, the gamma source spectra (or, "photon table results") produced by the input nuclear data library and the neutron source. Apply decay times of 10, 30, 60, 90, 120, 160, 270, 365, 1096, and 3652.5 days. List by nuclide, only. The input concentrations are given in the data. Set THERM = 0.632, RES = 0.333 and FAST = 2. The input data follows:

```
=ORIGEN
0$$ A5 27 E 1$$ 1 1T
PWR  NUCLEAR  DATA - SAMPLE CASE 1
4**  0.632  0.333  2  E  5$$  2  2T
35$$  0  4T
56$$  10  A13  50  4  3  0  2  1  E  57**  A3  1-14  E  5T
PWR - 3.3% ENRICHED U
  MT OF URANIUM CHARGED TO REACTOR
58**  F30  60**  81110  1100
66$$  1  A5  1  A9  1  E
73$$    60120   130270  140280  140290  220460  220470  220480  220490
220500  240500  240520  240530  240540  250550  260540  260560  260570  260580
270590  280580  280600  280610  280620  280640  400900  400910  400920  400940
400960  410930  420920  420940  420950  420960  420970  420980  421000  501120
501140  501150  501160  501170  501180  501190  501200  501220  501240
922350  922380  922340

74**   1.5  4.0   .607   .034   .304   .277  2.771  .204  .2  5.04
57.423  6.415  1.574  0.327  4.037  61.018  1.439  0.31  0.915  111.862
41.783  1.869  5.645  1.609  1421.122  306.725  462.239  460.074  72.5  10.258
.957  .532  .926  .958  .546  1.357  .54  .321  .219  .113
4.681  2.47  7.729  2.739  10.392  1.467  1.823  140.4  4062  1.13

75$$  47R1  3R2  6T
56$$  0  10  A10  10  A17  2  E  5T
60**  10  30  60  90  120  160  270  365  1096  3652.5
65$$  1  5Z  1  2Z  1  11Z  1  5Z  1  2Z  1  11Z  1 5Z  1 2Z  1  E
61** 5R1-3  1+6  1+4
6T
56$$  F0  5T
END
```

(Job execution requires: 270K; 700 I/O; and 15 sec on IBM 370/3033.)

Case 2 Time Integrated Fission Product Gamma Energy Spectra and Curies

Calculations are requested concerning fission product properties produced from exposing one gram of ^{235}U (as UO_2) to a thermal (or LWR) flux of 10^4 neutrons-cm^{-2}-sec^{-1} for 1200 days in six intervals. Compute the time integral of the fission product gamma energy spectra and nuclides in Ci-sec (with a cutoff of 10^{-30} Ci-sec) over the first three months, the fourth month and the thirteenth month. Apply: (1) the large fission product library and the σ_f = 409 barn for ^{235}U, which is in data set on unit 28 (and edit); (2) the binary mode Master Photon Library on unit 52; (3) either, the utility sub-matrix data set for the integral option on unit 66, if available, or create it during this job; and (4) an energy group structure range of 0 - 11 MeV with the lowest two energy intervals input as 0 - 2×10^5, 2×10^5 - 10^6 and the remaining intervals of 10^6 eV for the spectra. The input data, using an available data set on unit 66, follows:

```
=ORIGENS
0$$ A5 28 66 A8 52 E 1$$ 1 1T
SAMPLE CASE 2 - LARGE FISSION PRODUCT LIBRARY
3$$ 28 A11 1 1 A16 2 A18 922 -1 101 821 A32 66 12
4** 0.632 0.333 2 E 54$$ A9 2 E 2T
35$$ 0 4T
56$$ 6 6 1 A13 1 4 3 E 5T
U-235 FISSION PRODUCTS, SIGMA-F = 409 BARNS
 PER GRAM U-235 AT 10**4 N/(SEC-CM**2)
60** 41 200 1200 73$$ 922350 74** 1 75$$ 2 59** F1+4 6T
56$$ 0 4 A10 6 A14 4 0 0 4 E 5T
60** 120 150 365.25 395.25 61** F1-30 65$$ A49 1 E
81$$ 2 0 52 1 E 82$$ 6 6 0 6 83** 91 11+6 1+6 2+5 0 6T
GRAM U-235, 10**4 NEUTRONS/(SEC-CM**2), FIRST 3 MO. INTEGRAL SPECTRUM
GRAM U-235, 10**4 NEUTRONS/(SEC-CM**2), 4TH MO. INTEGRAL SPECTRUM
GRAM U-235, 10**4 NEUTRONS/(SEC-CM**2), 13TH MO. INTEGRAL SPECTRUM
56$$ F0 5T
END
```

(Job execution requires: 270K; 1500 I/O; and 44 sec on IBM 370/3033.)

F7.8. I/O UNIT REQUIREMENTS

In addition to units for the libraries the user has selected along with the card input and print units, the only (scratch) data set always required is on unit number NDISK. Also, if the user wishes to print FIDO edits on pages separate to the ORIGEN-S print, the use of unit NDUM is provided. The binary output units, NXTR, NVERT and LDSET, are required at the user's option. The following shows all logical units by name, default number, and purpose. Usually these are included in ORNL and ORGDP IBM.

UNIT	DEFAULT	PURPOSE
N5	5	Card input
KOUT	6	Printed output
NDUM	13	FIDO edits
NPUN	0	Card punch
NDISK	11	Scratch
NDSET*	27	BCD ORIGEN library input
NDSET	0	Binary ORIGEN library input
NVERT	0	Make or use in integral option
NDF	60	BCD photon data base input
NDF	0	Binary photon data base input
LDSET	51	Binary gamma source output
NXTR	0	Optional binary input output unit
ND30	94	Binary problem data interface
NGRP	82	Neutron energy-group structure

*Composed of six files, usually sequential. However, these six BCD data sets also may be on the six different units specified in IOS array, after setting NDSET = −20.

The input libraries usually have BLKSIZE=2000 in the DCB specifications of the IBM JCL. ORIGEN-S opens and then closes all data sets indicated by the 0$ array in order to allocate sufficient core storage for I/O buffers needed during execution. This appears to work properly for libraries with BLKSIZE=2000 and with no BUFL specified. However, libraries with a larger blocksize and with no BUFL specified usually result in a case that fails, because the large buffer is not reserved and may be already used. There may not be enough core in the total buffer storage area, since its size is limited to contain only the buffers in the order in which the data sets were opened. The failure is eliminated by setting BUFL to the following two conditions:

1. BUFL > BLKSIZE
2. BUFL * BUFNO = n * 2K - 16

   where,

   n = an integer

   k = 1024

F7.9. REFERENCES

1. M. J. Bell, *ORIGEN – The ORNL Isotope Generation and Depletion Code*, ORNL-4628 (CCC-217), May 1973.
2. D. Garber, et al., *ENDF-102, Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF*, Brookhaven National Laboratory, BNL-NCS-50496, March 1975.
3. H. Bateman, *Proc. Cambridge Phil. Soc., 15*, 423, 1910.
4. S. J. Ball and R. K. Adams, *MATEXP, A General Purpose Digital Computer Program for Solving Ordinary Differential Equations by the Matrix Exponential Method*, ORNL/TM-1933, August 1967.
5. L. Lapidus and R. Luus, *Optimal Control of Engineering Processes*, pp. 45-49, Bleisdell Publishing Co., Waltham, Mass., 1967.
6. D. R. Vondy, *Development of a General Method of Explicit Solution to the Nuclide Chain Equations for Digital Machine Calculations*, ORNL/TM-361, October 1962.
7. C. H. Westcott, *The Specification of Neutron Flux and Effective Cross Sections in Reactor Calculations*, Atomic Energy of Canada Limited, Chalk River, Ontario, CRRP-662, August 1956.
8. C. W. Kee, *A Revised Light Element Library for the ORIGEN Code*, ORNL/TM-4896, May 1975.
9. C. W. Kee, et al., "Processing and Testing of ENDF/B-IV Fission Product and Transmutation Data," *Trans. Am. Nucl. Soc. 19*, 398-99, 1974.
10. C. M. Lederer, et al., *Table of Isotopes*, 6th ed., Wiley, New York, 1967.
11. A. G. Croff, et al., *Updated Decay and Photon Libraries for the ORIGEN Code*, ORNL/TM-6055, February 1979.
12. W. B. Ewbank, et al., *Nuclear Structure Data File: A Manual for Preparation of Data Sheets*, ORNL-5054, June 1975.
13. W. B. Ewbank, "Evaluated Nuclear Structure Data File (ENSDF) for Basic and Applied Research," paper presented at the Fifth International CODATA Conference, Boulder, CO, June 1976.
14. M. J. Martin (ed.), *Nuclear Decay Data for Selected Radionuclides*, ORNL-5114, March 1976.
15. D. C. Kocher, *Nuclear Decay Data for Radionuclides Occurring in Routine Releases from Nuclear Fuel Cycle Facilities*, ORNL/NUREG/TM-102, August 1977.
16. Code of Federal Regulations, Title 10, Part 20.
17. C. E. Lee, *The Calculation of Isotopic Mass and Energy Production by a Matrix Operator Method*, Los Alamos Scientific Laboratory, LA-6483-MS, September 1976.

18. O. W. Hermann, et al., "Multigroup Time-Dependent Delayed Gamma Spectra from Fission and Activation of Uranium and Plutonium," *Trans. Am. Nuc. Soc. 26*, 571, June 1977.

19. M. L. Williams and C. R. Weisbin, *Sensitivity and Uncertainty Analysis for Functionals of the Time-dependent Nuclide Density Field*, ORNL-5393 (ENDF-263), April 1978.

20. A. G. Croff, et al., *Revised Uranium-Plutonium Cycle PWR and BWR Models for the ORIGEN Computer Code*, ORNL/TM-6051, September 1978.

21. M. L. Williams and S. Raman, "Calculational and Experimental Analysis of a ^{239}Pu Sample Irradiated in EBR-II," *Trans. Am. Nucl. Soc. 30*, 706, November 1978.

22. J. K. Bair and J. Gomez del Campo, "Neutron Yields from Alpha-Particle Bombardment," *Nucl. Sci. Engr. 71*, 18 (1979).

23. J. C. Ryman, "ORIGEN-S Data Libraries," Section M6 of *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation*, NUREG/CR-0200, Vol. 3 (April 1983).

24. D. C. Kocher, *Radioactive Decay Data Tables*, ORNL, DOE/TIC-11026 (1981).

25. D. L. Johnson, *Evaluation of Neutron Yields from Spontaneous Fission of Transuranic Isotopes*, Hanford Engineering Development Laboratory, HEDL-SA-973 (1975).

26. W. J. Swiatecki, *Phys. Rev. 100*, 937 (1955).

27. J. K. Bair and F. X. Haas, *Phys. Rev. C7*, 1356 (1973).

28. J. K. Bair and H. B. Willard, *Phys. Rev. 128*, 299 (1962).

29. D. H. Stoddard, *Radiation Properties of* ^{244}Cm Produced for Isotopic Power Generators, Savannah River Laboratory, DP-939 (1964).

30. S. J. Rimshaw and E. E. Ketchen, *Curium Data Sheets*, ORNL-4357 (January 1969).

31. D. H. Stoddard and E. L. Albenesium, *Radiation Properties of* ^{238}*Pu Produced for Isotopic Power Generators*, Savannah River Laboratory, DP-984 (July 1965).

32. A. G. Khabakhpashev, "The Spectrum of Neutrons from a Po-α-O Source," *Atomnaya Energiya 7*, 71 (1959).

NUREG/CR-0200
Volume 2, Section F9
ORNL/NUREG/CSD-2/V2/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

MORSE-SGC/S FOR THE SCALE SYSTEM

J. T. West*
T. J. Hoffman*
M. B. Emmett

Manuscript Completed: October 1981
Date Published: October 1981
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

*Formerly with the Computer Sciences Division

F9.A MORSE-SGC INPUT INSTRUCTIONS

F9.A.1 MORSE DATA BLOCK

Card A (20A4)

Title Card

Card B (4X,Z12)

RANDOM - starting random number seed

F9.A.2 MORSE DATA BLOCK 2 (FIDO INPUT)

1$ (14)

1.	IADJM	set >0 for an adjoint problem. All input data should still be in the forward mode; the program will adjoint it whenever IADJM>0.
2.	NSTRT	number of particles per batch.
3.	NMOST	maximum number of particles allowed for in the bank(s); may equal NSTRT+1 if no splitting, fission, and secondary generation.
4.	NITS	number of batches.
5.	NQUIT	number of sets of NITS batches to be run without new input data.
6.	NOT USED	set =0.
7.	ISTAT	set >0 to store Legendre coefficients.
8.	NSPLT	set >0 if splitting is allowed.
9.	NKILL	set >0 if Russian roulette is allowed.
10.	NPAST	set >0 if exponential transform is invoked (subroutine DIREC is called).
11.	NOLEAK	set >0 if nonleakage is invoked.
12.	IEBIAS	set >0 if energy biasing is allowed.
13.	NKCALC	the number of the first batch to be included in the estimate of k; if ≤0 no estimate of k is made.

14. NORMF — the weight standards and fission weights are unchanged if ≤0; otherwise, fission weights will be multiplied at the end of each batch by the latest estimate of k, and the weight standards are multiplied by the ratio of fission weights produced in previous batch to the average starting weight for the previous batch. For time-dependent subcritical systems, NORMF should be >0.

2$ (5)

1. MEDIA — number of cross-section media.
2. NMIX — number of mixing operations (elements times density operations) to be performed.
3. MEDALB — =0 if no specular reflection is used; >0 if specular reflection is used; value of MEDALB is used as the reflecting media.
4. MXREG — number of regions described by the geometry input.
5. MFISTP — set >0 if fissions are allowed.

3$ (20)

1. NNGA — number of neutron groups to be analyzed.
2. NGGA — number of gamma groups to be analyzed.
3. NNGTP — set >0 if a completely coupled neutron-gamma problem is desired (both neutrons and gammas are treated as primary particles). NNGA and NGGA must be equal to the number of neutron and gamma groups on the input cross section tape if this option is used.
4. NOT USED — set = 0
5. NOT USED — set = 0
6. NDSN — number of array analysis collision edits; see input discussion.
7. NDSG — if NPAST is >0, NDSG determines the direction of PATHLENGTH STRETCHING used in subroutine DIREC. The options available are:

   = 1 pathlength stretched in the + X direction (local)

three-dimensional matrix of numbers, with the cell position increasing in the positive X, Y, and Z directions, respectively. Each set of orientation data consists of the following parameters, separated by one or more blanks.

LTYPE The cell entry. LTYPE may be negative (array #), zero (empty cell), or positive (universe #).

IX1 The starting point in the X direction. IX1 must be at least 1 and $\leq$NXMAX.

IX2 The ending point in the X direction. IX2 must be at least 1 and $\leq$NXMAX.

INCX The number of cells by which increments are made in the positive X direction. INCX must be >0 and $\leq$NXMAX.

IY1 The starting point in the Y direction. IY1 must be at least 1 and $\leq$NYMAX.

IY2 The ending point in the Y direction. IY2 must be at least 1 and $\leq$NYMAX.

INCY The number of cells by which increments are made in the positive Y direction. INCY must be >0 and $\leq$NYMAX.

IZ1 The starting point in the Z direction. IZ1 must be at least 1 and $\leq$NZMAX.

IZ2 The starting point in the Z direction. IZ2 must be at least 1 and $\leq$NZMAX.

INCZ The number of cells by which increments are made in the positive Z direction. INCZ must be >0 and $\leq$NZMAX.

ISTP = 0, read another set of data,
$\neq$ 0, do not read any more mixed-cell orientation data.

An important feature of this type of data description is that, if any portion of an array is defined in a conflicting manner, the last card to define that portion will be the one that determines the array's cell type configuration. To utilize this feature, one can fill an entire array with the most prevalent cell type and then superimpose the other cell types in their proper places to accurately describe the array. The last set of mixed-cell orientation data must have a nonzero entry in the last field.

IOP = 2 - Standard FIDO Array Input

The array being entered is integer; therefore, it is a "$" or "$$" array. The array may be entered in the standard free-form

FIDO format. The description for each lattice array is entered as a single array block with FIDO. The FIDO integer array number entered is the array number being described plus 100. The data is entered and each array description is terminated with a "T." All standard FIDO repeat options are available for entering the data. Array 1 would be entered as the "101$$" FIDO array terminated with a "T." The process would continue until all array descriptions have been entered. The format for the data entry is the same as the description for free-form input. All x entries for the first y row and first z level are entered, then all x entries for the second y row and first z level are entered. This process continues until the entire first z level has been described. Then the second z level is described until the entire array has been described. Then the geometry array description or a given array is terminated with a "T."

A.10 Universe Type

One entry is required for each universe modeled in the combinatorial geometry, starting with Universe 1. The entries should be either a zero or a 1: a zero if the universe is "combinatorial" and a 1 if the universe is "simple." A "simple" universe is a universe composed of concentric zones, where every zone completely surrounds the zone inside of it. Furthermore, input zones in a simple universe may be only one code zone and may be described by only one or two bodies. Tracking through "simple" universes is about 30% or more faster than through regular combinatorial geometry tracking, although the modeling capability is limited. Simple universes may be combined with regular combinatorial universes in arrays without any problems.

--- End of Geometry Input ---

F9.A.4 MORSE DATA BLOCK 3 (FIDO INPUT)

6* (Ten entries)

1. TMAX Maximum cpu time in minutes allowed for the problem on the computer.

2. TCUT Age in seconds at which particles are retired; if TCUT=0, no time kill is performed.

3. NABP Neutron angular biasing parameter to bias the source particles to the first point detector. Set < 0 if not using angular biasing in SOURCE subroutine.

4. NABOG Gamma angular biasing parameter; used to bias secondary gamma particles to the first point detector location; enter zero for no biasing; see input description for further discussion.

5. XMIN
6. XMAX
7. YMIN
8. YMAX
9. ZMIN
10. ZMAX

SOURCE SAMPLING VOLUME BOUNDARIES (absolute coordinates). Source coordinates are sampled uniformly within the bounds input by the user. If the user has specified a source media, IGQPT, then only points in the specified media will be accepted.

7* (NNGA)

1. (VELTH(I), I=1,NNGA) The neutron velocities for all energy groups being analyzed. If not input, calculated values will be used based on the group energy limits.

8* (4*MAXGP*MXREG) required only if NSPLT+NKILL+NPAST >0

1. ((WTHI(I,J),I=1,MAXGP), J=1,MXREG) weight above which splitting will occur.

2. ((WTLO(I,J),I=1,MAXGP), J=1,MXREG) weight below which Russian roulette is played.

3. ((WTAV(I,J),I=1,MAXGP), J=1,MXREG) weight parameters for particles surviving Russian roulette.

4. ((PATH(I,J),I=1,MAXGP), J=1,MXREG) pathlength stretching parameters for use in exponential transform (usually 0 ≤ PATH <1)

9* ((NNGA+NGGA)*MXREG) required only if IEBIAS >0.

1. ((EPROB(I,J),I=1,NNGA+NGGA,J=1,MXREG) values of the relative energy importance of particles leaving a collision in region J going to group I.

10* (MXREG+((NNGA+NGGA)*MEDIA) required only if MFISTP >0. Initialized values will be used if not input.

1. (FWLO(I),I=1,MXREG) values of the weight to be assigned to fission neutrons. (Initialized to 1.0 if not input.)

2. ((FSE(I,J),I=1,NNGA+NGGA,J=1,MEDIA) fraction of fission-induced source particles in group I for medium J. (Uses fission spectrum of element with largest $\nu\Sigma_f$ if not input.)

11* NNGA*MXREG if IADJM ≤0 required only if coupled neutron-gamma
or
NGGA*MXREG if IADJM >0 ray problem, but not completely coupled.

1. ((GWLO(I,J),I=1,NNGA or NGGA),J=1,MXREG) probability of generating a gamma at each collision. Alternately (depending on GPROB) the values of the weight to be assigned to the secondary particles being generated. NGGA groups are read for each region in a forward problem and NGGA for an adjoint.

12$ (NMIX)

1. (KM(I),I=1,NMIX) MORSE media number.

13$ (NMIX)

1. (KM(I),I=1,NMIX) element or ICE identifies.

14* (NMIX) (14*Array not input if using an ICE tape.)

1. (RHO(I),I=1,NMIX) density.

15$ NOT USED

16$ NOT USED

17* (NGPFS) required only if ISOUR ≤0

1. (FS(I),I=1,NGPFS) the unnormalized fraction of source particles in each group. Uses FSE of media number 1 if not input and MFISTP >0.

18* (NGPFS) required only if ISOUR ≤0 and ISBIAS ≠0.

1. (BFS(I),I=1,NGPFS) the relative importance of a source in group I.

T (Block 3 must be terminated with a T.)

F9.A.5 MORSE DATA BLOCK 4 (ANALYSIS DATA)

Card AA (20A4)

Title or units for total response for all detectors.

Card AB (20A4*NRESP)

NRESP* title or units for each total response for all detectors.

MOR is called by ALOCAT, which was called from MPROG. It simply calls MORSE without performing any function other than reordering the arguments in the call.

MORSE is called from MOR. This is the controlling routine for the MORSE execution. The entire program flow results from calls in MORSE.

MPROG was the main program in MORSE-SGC. In this version it is a subroutine called by either QMORSE or MORSEG, the two entry points in the SCALE version of MORSE-SGC/S. It reads the first block of data, and calls ALOCAT, which calls MOR, which calls subroutine MORSE.

MSOUR is called from BATCH at the beginning of a particle batch. It calls FSOUR to obtain the fission bank for criticality problems. It calls SOURCE for fixed source problems. It calls GETNT and STORNT to update the neutron particle bank.

MWLIST is called from XSEC2. It prints out the transfer matrices in a variable format.

NDBARC is called from BATCH for the end of batch processing. NDBARC updates the array edits at the end of each batch and normalizes the result. The mean and the square of each batch estimate is computed in NDBARC.

NDBTCH is called from BATCH for the end of batch processing. It adds the average responses for each point detector at the end of a batch to the results for the run. The square and the result are tabulated for later computation of the mean and the fractional standard deviation at the end of the run.

NORML is called by GOMST to compute the normal direction to the surface of an albedo media. It obtains the body number, NASC, from the combinatorial common PAREM. The body encountered must be referenced in the albedo zone description or the code will print a fatal error. The body surface number is LSURF. This is the only routine in MORSE using LSURF.

NRUN is called from MORSE to perform analysis processing at the completion of the run. It works with the arrays prepared by NDBTCH. VAR2 and VAR3 are called to compute the fractional standard deviations. The point detector results and the fluence results for the point detectors are printed out by NRUN.

NSIGTA is called from NXTCOL and EUCLID to look up the total cross section for a given energy group and media. The media is checked to verify its validity. The total cross section for an internal void is returned as zero.

NXTCOL is called from SUPGRP to track a particle to its next collision site. It calls GOMST to control the tracking. It returns when the particle has had a collision, leaked from the system, or encountered an albedo boundary. It will optionally call BDRYX for boundary crossing estimates.

OUTPT is called by BATCH at the end of a batch and by MORSE at the end of a run. It prints out the summary information for the collisions, boundary crossings, splittings, leaks, etc., for the end of a batch. At the end of a run it prints out the neutron death table and the region counters; scattering, fission, Russian roulette kills, survivals, etc. The region counters and the Russian roulette and splitting arrays require considerable memory, especially when many regions are modeled. K-effective for criticality runs is computed and printed out by OUTPT.

OUTPT2 is called from OUTPT to print out two-dimensional arrays.

PRT1D is called from XSEC2 to print out one-dimensional cross sections.

PRT1D2 is called from XSEC7 to print out two-dimensional cross section arrays.

PTHETA is called from RELCOL and RELCOLA to determine the probability of scattering through angle for each of the possible energy transfers. All possible downscatters and upscatters are computed.

PTPT is called from RELCOL, SDATA, and SGAM to compute the distance from the collision site to the point detector. The point detector location is in global coordinates, while the collision location is in local coordinates. CLEV is called by PTPT to compute the global coordinate of the collision site.

Q function is called from ANGLES, BADMOM, and FIND to compute the value of an orthogonal polynomial at a position X.

QMORSE and MORSEG are the two entry points to initiate MORSE execution. The entry point taken determines whether the input is read from cards or from binary files in SCALE.

RANNO is the assembler random number generator.

RELCOL is called from SUPGRP at a collision site. It calculates collided flux estimates to the point detectors by calling PTHETA, EUCLID, and FLUXST. RELCOL makes only one estimate to each point detector from each collision site, as opposed to RELCOA which makes multiple estimates.

RELCOA is used to make estimates to point detector sites from each collision site. IT calls PTHETA, EUCLID, PTPT, and FLUXST. Estimates are made to each possible group transfer given the fixed angle of scatter to travel to a point detector.

RESET is called to reset array tracking variables destroyed by point detector estimation. During tracking, estimates may be made from collision sites to point detectors. After the collided flux estimate has been made, the random walk is continued. It is necessary to reset the particle nesting table to its previous condition at the collision site before continuing the particle track.

RINPU1 is called from MORSE. It prints out most of the MORSE input edit and reads in a large portion of the user's input. When MORSE is running outside of SCALE, RINPU1 calls FIDAS to read the input. When MORSE is running as a part of SCALE, RINPU1 reads a binary input file by calling QOREAD. ITYPE in label common SCALOP determines which way MORSE is running.

RINPU2 is called from MORSE. It computes the array pointers for data storage and the amount of supergrouping required. Many data arrays are shifted around in memory by RINPU2 to their final locations. XSEC3, XSEC4, XSEC5, XSEC6, and XSEC8 are called by RINPU2.

RITNUT is a utility routine, which may be called to dump out the contents of label common NUTRON containing particle tracking information, coordinate, weight, direction cosines, etc.

RNDIN loads a seed into the random number generator RANNO.

RNDOUT gives the last random number returned by RANNO, without changing the random number sequence.

SCORIN is called by RINPU2. It prints out the point detector coordinates, response functions, energy bank boundaries, and angle bank boundaries related to point detector scoring.

SDATA is called from MSOUR to compute the uncollided flux estimate to a point detector from a source point. It calls PTPT, EUCLID, and FLUXST. The switch flag in the call to FLUXST is set not to add the uncollided flux estimate to the collided flux estimate.

SGAM is called from FBANK and GSTORE to compute the uncollided flux estimate to point detectors from secondary particle birthsites. SGAM is identical to SDATA. It calls PTPT, EUCLID, and FLUXST. The switch flag in the call to FLUXST is set not to add the uncollided flux to the collided flux estimate.

SHIFT is called to move data from one area of memory to another area of memory.

SHUFLE is called from RINPU2. It computes the multigroup cross section pointers after the cross section storage has been supergrouped.

SIZLX is called from RINPU2. It computes the space required for cross section storage. This determines the amount of supergrouping and the size of each supergroup. This is a function of available core memory, the number of energy groups, and the number of media.

SORIN is called from RINPU2. It computes the cumulative distribution for source energy spectrum for both forward and adjoint problems. The distributions are normalized, and are ordered depending on whether a forward or adjoint case is being run. The cumulative distribution is increasing to 1 for a forward problem, and decreasing from 1 for an adjoint problem.

SORK sorts the neutron bank by supergroup at the beginnning of each supergroup. During a batch, particles from the most populous supergroup will be tracked, then the next most populous supergroup, etc. SORK calls STORNT and GETNT in ordering the neutron bank. SORK is called from BATCH.

SOURCE is called from MSOUR. It calls CALI to determine the location of a source particle. The version of SOURCE in MORSE-SGC/S samples uniformly from XMIN to XMAX, YMIN to YMAX, and ZMIN to ZMAX. If the user has specified a source media, only those points sampled in the media will be saved as a valid source particle. Angular biasing is provided as an option in SOURCE. Isotropic selection of initial starting cosines is available by default in SOURCE by calling GTISO.

STBTCH is called by BATCH. It clears the response and detector analysis arrays at the beginning of each batch by calling CLEAR.

STRUN is called from MORSE. The user may supply his own version of STRUN. The version supplied does nothing. It is called at the start of a new run.

SUPER1 is called from RINPU2. This routine calculates the lengths of each of the supergroups stored in memory.

SUPER2 is called from RINPU2. This routine constructs a table giving the supergroup number for each energy group number.

SUPGRP is called from BATCH. This routine controls the random walk for each particle in a supergroup. It calls NXTCOL for particle tracking. It calls routines to handle the secondary particle generation and banking. Russian roulette and splitting is performed in SUPGRP. COLISN is called from SUPGRP for collision processing.

TIMER is a utility routine maintaining local and global clocks for MORSE during execution.

TRNPSE is called from XSEC4 to transpose an array.

VAR2 is called to compute the fractional standard deviation of the analysis arrays. VAR2 works on two-dimensional arrays and can independently weight batches. VAR2 is called from NRUN.

VAR3 is called from NRUN. It is the same as VAR2 except it can work on three-dimensional arrays.

XNPUT is called from BATCH. This routine retrieves cross section data from disk for a given supergroup during batch processing.

XSEC1 is called from RINPU1. It reads the ID record and energy group structure from an AMPX working tape.

XSEC2 is called from RINPU1. It reads the cross section data off the AMPX working tape for elements in the mixing table and stores them on disk. It also calculates the storage space that will be needed to store the mixed cross sections.

XSEC3 is called from RINPU2. It mixes the cross sections and stores the mixed cross sections on disk. It also selects FSE from xsec data if it wasn't read in.

XSEC4 is called from RINPU2. This routine will invert the cross sections for an adjoint problem. It sums the neutron to gamma transfer arrays into one-dimensional gamma production cross sections. It accumulates and normalizes the two-dimensional cross sections as needed. It inverts FSE array also. Not called if previously mixed cross sections (JXTAPE > 0).

XSEC5 is called from RINPU2. It calculates the angles and probabilities for collision processing. Cumulative angular probability distributions are constructed for each energy transfer. This data is stored by supergroup on disk.

XSEC6 is called from RINPU2 before XSEC5. This routine reads cross sections off disk, reorders them into supergroups, and stores them back on disk.

XSEC7 is called from RINPU1. It reads a preprocessed, premixed ICE (see Sect. F8) tape, and retrieves the mixtures requested. XSEC8 is called from RINPU2. It performs similar functions as XSEC5 and XSEC6, except for an ICE premixed tape.

TABLE OF CONTENTS

Page

List of Figures vii

List of Tables ix

Acknowledgments xi

Abstract xiii

F10.1. INTRODUCTION F10.1.1

F10.2. NUMERICAL TECHNIQUE F10.2.1

F10.2.1. Statement of the Problem F10.2.1
F10.2.2. Steady-State Heat Conduction F10.2.4
F10.2.2.1. Point Successive Overrelaxation Iteration F10.2.4
F10.2.2.2. Direct Solution Technique F10.2.7
F10.2.3. Transient Heat Conduction F10.2.10
F10.2.3.1. Classical Explicit Procedure F10.2.10
F10.2.3.2. Levy's Modification to the Classical Explicit Procedure F10.2.11
F10.2.3.3. Implicit Procedure F10.2.12
F10.2.3.4. Variable Time Step Size F10.2.17
F10.2.4. Temperature-Dependent Thermal Properties F10.2.19
F10.2.5. Boundary Conditions F10.2.20
F10.2.6. Finned Surfaces F10.2.22
F10.2.7. Change of Phase F10.2.29
F10.2.8. Initial Temperatures F10.2.31

F10.3. SUGGESTIONS ON USE OF HEATING6 ..... F10.3.1

F10.3.1. General F10.3.1
F10.3.2. Regions F10.3.1
F10.3.3. Lattice Arrangement F10.3.5
F10.3.4. Analytical and Tabular Functions F10.3.5
F10.3.5. Input Parameters F10.3.6
F10.3.6. User-Supplied Subroutines F10.3.8
F10.3.7. Selection of Features for a Particular Calculation F10.3.11
F10.3.7.1. Default Value Modifications F10.3.11
F10.3.7.2. Variably Dimensioned Arrays F10.3.21
F10.3.7.3. Optional Features F10.3.23
F10.3.7.4. Maximum Bandwidth for Coefficient Matrix for Steady-State Direct Solution Technique F10.3.23
F10.3.7.5. Input/Output Logical Unit Numbers F10.3.27
F10.3.7.6. User-Supplied Format Codes for Output Maps F10.3.32
F10.3.7.7. User-Supplied Material Dividers for Output Maps F10.3.33

TABLE OF CONTENTS (Contd.)

F10.3.7.8. Output of Variably Dimensioned Array Pointer Values F10.3.33
F10.3.7.9. HEATING6 Material Properties Libraries F10.3.37
F10.3.8. Determination of HEATING6 Core Requirements F10.3.38

F10.4. PROGRAM STRUCTURE F10.4.1

F10.5. INPUT DESCRIPTION F10.5.1

F10.5.1. General F10.5.1
F10.5.2. NAMELIST /OPTION/ F10.5.4
F10.5.3. Card 1 - Title of Problem F10.5.4
F10.5.4. Card 2 - Problem Control Parameters F10.5.7
F10.5.4.1. Maximum CPU Time (I) F10.5.7
F10.5.4.2. Geometry Type (I) F10.5.7
F10.5.4.3. Type of Problem (I) F10.5.7
F10.5.4.4. Initial Time (R) F10.5.8
F10.5.4.5. Final Time for Transient Calculation (R) F10.5.8
F10.5.4.6. Time Increment (R) F10.5.9
F10.5.4.7. Levy's Explicit Method Option (I) F10.5.9
F10.5.4.8. Temperature Units (I) F10.5.9
F10.5.4.9. Problem Continuation Indicator (I) F10.5.10
F10.5.5. Card 3 - Input/Output Parameters F10.5.10
F10.5.5.1. Graphical Output (I) F10.5.10
F10.5.5.2. Frequency of Output for Plots (I) F10.5.11
F10.5.5.3. Special Monitoring of Temperatures (I) F10.5.11
F10.5.5.4. Initial Temperature Input Option (I) F10.5.11
F10.5.5.5. Final Temperature Output Option (I) F10.5.12
F10.5.5.6. Standard Output Map Flag for Three-Dimensional Problems (I) F10.5.12
F10.5.5.7. Transient Output (I) F10.5.13
F10.5.5.8. Problem Status Unit for Remote Users (I) F10.5.13
F10.5.5.9. Output of Selected Information (I) F10.5.13
F10.5.5.10. Melting Ratio Map Flag (I) F10.5.14
F10.5.6. Region Data Block (Cards R1 and R2) F10.5.15
F10.5.6.1. Block Indicator Card F10.5.15
F10.5.6.2. Card R1 ..... F10.5.15
F10.5.6.3. Card R2 ..... F10.5.16
F10.5.7. Material Data Block (Cards M and PC) F10.5.16
F10.5.7.1. Block Indicator Card F10.5.17
F10.5.7.2. Card M (Standard Form) F10.5.17
F10.5.7.3. Card PC F10.5.17
F10.5.7.4. Card M (Material Properties Library) F10.5.18
F10.5.8. Initial Temperature Function Data Block (Card I) F10.5.19
F10.5.8.1. Block Indicator Card F10.5.19
F10.5.8.2. Card I F10.5.19
F10.5.9. Heat Generation Function Data Block (Card G) F10.5.20
F10.5.9.1. Block Indicator Card F10.5.20
F10.5.9.2. Card G F10.5.20

The fin effectiveness η_f is the ratio of the heat transfer from the fin to that from the base area of the fin assuming convection to the fluid at the same temperature and convective heat transfer coefficient. The fluid temperature and heat transfer coefficient and fin material thermal properties are assumed constant along the length of the fin. This concept is extended to finned surface effectiveness η_s. The finned surface effectiveness is the ratio of the heat transferred by the fin surface and the surface between the fins to that transferred by the surface if the fins were removed.

$$\eta_S = \frac{Q_{fin} + Q_w}{Q_{base}} = R_{Sf}\,\psi + R_{Sw} \qquad \text{(F10.2.60)}$$

where

Q_{fin} = heat flow from surface of fin,

Q_w = heat flow from portion of base surface not covered by fin,

Q_{base} = heat flow from surface assuming fins removed,

ψ = fin efficiency = $\dfrac{Q_{fin}}{Q_{fin}^{(k=\infty)}}$,

$Q_{fin}(k=\infty)$ = heat flow from fin surface assuming fin thermal conductivity is infinite,

R_{Sf} = (fin surface area)/(unit area of base surface),

R_{Sw} = (base surface area not covered by fins)/(unit area of base surface).

In HEATING6, the fin effectiveness is used to modify the effective conductance for the boundary condition on the right side of Eq. (F10.2.57) by adjusting the effective heat transfer coefficient in Eq. (F10.2.59) as follows:

$$h = h_c\eta + h_r\left[(T_f^*)^2 + (T_b^*)^2\right]\left[T_f^* + T_b^*\right] + h_m\eta\left|T_f^* - T_b^*\right|^n e \qquad \text{(F10.2.61)}$$

where

$\eta = \eta_S$ = finned surface effectiveness for that boundary condition and will be the same for both natural or forced convection.

Tables F10.2.2 and F10.2.3 describe the fin effectiveness models implemented in HEATING6. The fin effectiveness is determined from simple geometric input and the material properties of the surface node. If the surface node's material properties or the boundary heat transfer coefficient are temperature-dependent, the fin effectiveness is updated whenever the temperature-dependent properties are updated.

A similar approximation was utilized to allow transient analysis of finned surfaces even though the models in Table F10.2.3 were developed for steady-state applications only. The fin heat capacity was added to the heat capacity of the node to approximate the transient effect of the fin on the surface node. In general, the transient response of the surface node's temperature is a very complex calculation and could not be included in this technique. If this heat capacity modification is not desired, it can be eliminated by setting HCFAC < 0.0. HCFAC is a multiplier for the heat capacity term and can be any positive value (see input description) and is applied as in Eq. (F10.2.62) below. The specific heat and density can be temperature-dependent and will cause the updating of the fin effectiveness and fin heat capacity.

$$C_n = \left[\sum_{i=1}^{8} C_{P_i} \rho_i V_i \right] + \sum_{i=1}^{L_f} C_{P_f} \rho_i V_{f_i} \, HCFAC_i \qquad \text{(F10.2.62)}$$

where Eq. (F10.2.62) is a modification of Eq. (F10.2.2),

C_n = heat capacity for node n,

L_f = number of finned surfaces associated with node n,

C_{P_i} = specific heat for fin material associated with the ith portion of node n's surface area,

ρ_i = density of fin material associated with the ith portion of node n's surface area,

V_{f_i} = volume of fin associated with the ith portion of node n's surface area.
$= R_{VA} A_i$,

R_{VA} = fin volume/(unit area of finned surface), and

A_i = surface area for the ith portion of node n's surface area associated with a finned boundary condition.

Fin effectiveness models not included in Table F10.2.2 can be supplied in a user-supplied subroutine, FINUSR.

F10.5.4.6 *Time Increment (R)*

This entry (the sixth on Card 2) contains the initial time increment, Δt [see Eq. (F10.2.25)], for transient problems that will be solved by one of the explicit techniques. For transient problems which will be solved using the implicit procedure, this entry may be zero. For the Classical Explicit Procedure, the time increment must satisfy certain criteria to insure stability [see Eq. (F10.2.26)]. The HEATING6 Program calculates the stability criterion and writes, "THE STABILITY CRITERION IS (the value of criterion) FOR POINT (lattice point yielding the above value)". If the input time increment is less than or equal to the stability criterion, then the code writes, "THE INPUT TIME INCREMENT SATISFIES THE STABILITY CRITERION," and uses the input value. If the input time increment is too large (but not more than a factor of 10 too large), the program sets Δt equal to the calculated stability criterion and writes, "THE INPUT TIME INCREMENT DOES NOT SATISFY THE STABILITY CRITERION–THE TIME INCREMENT IS NOW = (value of Δt used in subsequent calculations)." If the input Δt is too large by more than a factor of 10, the program writes, "CASE DELETED–INPUT TIME INCREMENT EXCEEDS THE STABILITY CRITERION BY MORE THAN A FACTOR OF 10," and goes on to the next problem. **For steady-state-only problems, this entry may be zero. This entry must be a non-negative real number. If the Classical Explicit Procedure or Levy's explicit method is to be used and if this entry is zero, HEATING6 will choose the initial time step equal to the stability criterion for the Classical Explicit Procedure.**

F10.5.4.7 *Levy's Explicit Method Option (I)*

The seventh entry (an integer) is the factor by which the stable time increment is multiplied to form the time increment for Levy's explicit method. If the entry is zero or 1, then Levy's method will not be used. If Levy's explicit method is specified (greater than zero), then the problem is not terminated if the input time increment is more than an order of magnitude larger than the stability criterion as noted in Sect. F10.5.4.6. Instead, the time increment is initially set equal to the stability criterion. HEATING6 always executes 10 time increments using a stable time step prior to using Levy's explicit method. This entry may be zero for steady-state-only (Type 1) problems.

F10.5.4.8 *Temperature Units (I)*

For problems involving radiation, Fahrenheit or Celsius temperatures associated with calculating the effective thermal conductances due to radiative heat flow are converted to Rankine or Kelvin temperatures, respectively. Otherwise, temperatures are assumed to be in absolute units and no conversion is made. Entry 8 is used only for problems involving radiation. It indicates that the units of temperature are either in degrees Fahrenheit, degrees Celsius, or absolute degrees. This entry is an integer.

Entry 6	*Temperature*
0	°F
1	°C
2	Absolute

F10.5.4.9 *Problem Continuation Indicator (I)*

JOBCON is an integer flag to indicate whether the problem is a new case or a continuation of a previous case. If JOBCON < 1, a new case is being run. If JOBCON ≥ 1, a continuation of the previous case is being run.

If JOBCON = 1, the initial temperature distribution will be calculated using the information from the REGIONS and INITIAL TEMPERATURES Data Blocks from the previous case subject to any changes defined by the current case. However, any initial temperatures specified in the TEMPERATURES OF NODES Data Block applying to this case will override initial temperatures calculated from the REGIONS and INITIAL TEMPERATURES Data Blocks.

If JOBCON = 2, this problem will begin with the initial temperature distribution of the preceding problem. The initial time will be equal to the initial time of the preceding problem.

If JOBCON = 3, this problem will begin with the final temperature distribution of the preceding problem. The initial time will be equal to the initial time of the preceding problem.

If JOBCON = 4, the initial temperature distribution of this problem will be equal to the final temperature distribution of the preceding problem. The initial time will be equal to the final time of the preceding problem.

If JOBCON = 5, the initial temperature distribution of this problem will be equal to the initial temperature distribution of the previous problem. The initial time will be the value of the fourth entry on this card.

If JOBCON = 6, this problem will begin with the final temperature distribution of the preceding problem. The initial time of this problem will be the value of the fourth entry on this card.

If the initial temperatures of certain nodes are specified on the IT cards, the specified temperatures will be used regardless of the value of JOBCON.

F10.5.5. Card 3 - Input/Output Parameters

Each entry on this card is an integer.

F10.5.5.1 *Graphical Output (I)*

The first entry (JPLOT) on Card 3 indicates whether or not graphical output is desired.

F10.5.5.7 *Transient Output (I)*

For transient problems, the output may be specified in either of two ways:

(1) The temperature distribution may be printed at equally spaced times. To specify this option, the seventh entry on Card 3 must contain the number of initial time steps between outputs. For example, if a value of 5 is entered, the temperature distribution will be printed at times whose spacing is equal to five times the initial time step.

(2) The temperature distribution may be printed at unequal time increments. To choose this option, the actual output times will be entered in the PRINTOUT TIMES data block and the seventh entry on Card 3 is ignored.

The temperature distribution is automatically printed prior to the first time step for transient calculations and prior to the first iteration for steady-state problems. Also, the temperature distribution at the end of either a transient or steady-state calculation is automatically printed. For steady-state-only calculations or for transient problems containing a PRINTOUT TIMES data block, this entry may be zero.

F10.5.5.8 *Problem Status Unit for Remote Users (I)*

This feature is designed to allow remote users to determine the status of their problem without having to wait for normal computer turnaround. If a positive number is entered in the eighth field of Card 3, then this directs the code to write error messages and selected information concerning the status of the problem on logical unit IERROR. The user must supply the appropriate data or file definition parameters to define the specified unit. As an example, the user may wish to use this option to have information routed to a remote terminal. One can locate data errors and resubmit the corrected problem without having to wait for the main listing from the computer. Using this concept, one could possibly speed up turnaround time. Enter zero if this option is not desired. IERROR appears in labeled common /IOUNIT/ and its value may be changed by using NAMELIST /OPTION/ in the input data or by using a BLOCK DATA subprogram.

F10.5.5.9 *Output of Selected Information (I)*

The option to output selected information during the calculations exists to aid the user in the detection of input data errors and to assist in debugging and understanding cases that are not performing as anticipated. Several types of output are available using this feature. They include: a nodal description which consists of tables indicating the neighbors and regions corresponding to each node; a list of nodal connectors for surface-to-surface boundary conditions; thermal property tables containing the temperature, capacitance, power, and ordered pairs consisting of each neighbor and the effective conductance between the node and the neighbor for each node in the model along with information concerning the temperature

dependence for the heat capacitance, effective conductance, and power associated with each node; tables containing phase change information at each node; and the temperature distribution after each iteration or time step. The thermal property tables may be output either for the initial time level only or for each time the thermal properties are reevaluated. The temperature distribution may be output at the end of each iteration in the steady-state techniques, at the end of each time step for the explicit transient methods, and at the end of each iteration in either the nonlinear or the linear loop for the implicit transient calculations. The user is reminded that these temperature distributions have not satisfied the convergence criteria. All output using this feature is generated on the standard output unit. The user should exercise caution in using this feature since it will generate excessive output.

The output features desired for a particular case are defined to the code by entering the appropriate integer in the ninth field on Card 3. The value of this integer is determined as follows. The value of the flag for each desired option is located in the following list, and the sum of these values is the integer that defines this option.

Option	*Value of Flag*
Nodal Description	1
Capacitance, Power, Conductance at Initial Time Level	10
Capacitance, Power, Conductance at Every Reevaluation	20
Temperature Distribution Every Iteration, Steady State	100 or 200
Temperature Distribution Every Time Step, Transient Explicit	100 or 200
Temperature Distribution Every Nonlinear Iteration, Transient Implicit	100
Temperature Distribution Every Linear Iteration, Transient Implicit	200

Only one option involving output of thermal properties may be specified for a case. Also, only one option involving output of temperature distribution may be specified for a case.

As an example, assume the user wants the thermal properties output at the initial time level (value of flag is 10) and the temperature distribution output at every linear iteration (value of flag is 200). Output for the nodal description is not desired for this case. Then, to define these output features to the code, the ninth entry on Card 3 must contain the integer 210.

F10.5.5.10. ***Melting Ratio Map Flag (I)***

This entry (NPRTMR), the tenth, is a flag to control the printing of the nodal melting ratio map following each transient temperature map for problems involving phase change calculations. If the entry is 1, HEATING6 omits the nodal melting ratio map. Otherwise, the map is printed. Omitting this map can reduce the printed output by up to 50%.

Corrected 12/84

Table F10.5.5. Summary and Format of Input Data for HEATING 6

Entry 1	Entry 2	Entry 3	Entry 4	Entry 5	Entry 6	Entry 7	Entry 8	Entry 9	Card
JOBDES - Job Description - Up to 72 Alphanumeric Characters									Card 1 (F10.5.3)
MXCPU(I) Maximum CPU time for IBM 3033 or IBM 360/195 to execute.	MGEOM(I) Geometry type 1-ABE 5-RTZ 2-XY 7-XI 3-RZ 8-XZ 4-R 9-Z 5-Z 10-R (sphere)	MTYPE(I) Problem type. 1 steady state -1 transient (See Sec. F10.5.4.2 for other types.)	TIME(R) Initial time. (See Sec. F10.5.4.5 for more information.)	FTIME(R) Final time.	DELTA(R) Time increment for transient problem involving explicit technique.	RINFCT(I) Factor by which stable time increment is increased if Levy's explicit method is used.	IDEGBR(I) Temperature units. 1 implies °C and radiation is involved. 2 implies absolute degrees. 0 otherwise.	JUNCOND(I) Case type 0 for new case. 1 or greater for continuation of previous case. (See Sec. F10.5.4.8)	Card 2 (F10.5.4)
JPLOT (I) Plot data set flag. Non-zero implies data will be written on unit JPLOT. Negative implies data will be added to data from previous runs read from unit JPLOTR.	MOUTPT(I) Number of time steps between outputs for plots on unit number JPLOT. If zero, output is at normal printcycle.	NTS(I) Number of iterations or time steps between output for special monitoring of temperatures.	JIFIN(I) Flag for reading initial temperature distribution on logical unit ITFIN. Positive, formatted. Negative, unformatted.	JTPOUT(I) Flag for writing final temperature distribution on logical unit ITPOUT. Positive, formatted. Negative, unformatted.	IDPLANE(I) Three-dimensional output map flag. 0 for XY planes, 1 for XZ planes, 2 for YZ planes.	NDTA(I) Number of time steps between standard output of temperature maps.	JERROR(I) Flag for output of selected data indicating problem status [illegible] JERROR.	IMONIN(I) Flag for output of selected information during calculations. Entry 10: NPRTMAT(I) Flag for output of melting ratio map.	Card 3 (F10.5.5)
REGIONS									
NUMRO (I) Region number.	MATL (I) Region material number. (Card 4) Enter zero for gap region.	XMIN (R) Smaller X or R region dimension.	XMAX (R) Larger X or R region dimension.	THLT (R) Smaller Y or Θ region dimension.	YMRT(R) Larger Y or Θ region dimension.	IDBE(R) Smaller Z region dimension.	IZPE(R) Larger Z region dimension.		Card 3* (F.10.5.6) Maximum MAXREG
ITEM (I) Region initial temperature function number. (Card 3)	NUEN (I) Region heat generation function number. (Card 5)	NREGIN (I) Boundary condition on smaller X or R. (Card 8)	NREOUT (I) Boundary condition on larger X or R. (Card 8)	NBELT (I) Boundary condition on smaller Y or Θ. (Card 8)	NBURT(I) Boundary condition on larger Y or Θ. (Card 8)	NBEIN(I) Boundary condition on smaller Z. (Card 8)	NBEPO(I) Boundary condition on larger Z. (Card 8)		Card 4* (F.10.5.6)
MATERIALS									
MAT (I) Material number.	MATNAM (A) Material name	CONDUC (R) Conductivity if constant.	DENSTY (R) Density if constant. Enter zero for steady state only.	SPHEAT (R) Specific heat if constant. Enter zero for steady state only.	MCONTP(I) Conductivity temperature-dependent function number. NOTE: For the 5th through 9th entries, positive integer implies analytical function and negative integer implies tabular function.	MDENSY(I) Density temperature-dependent function number. Enter zero for steady state only.	MSPHTP(I) Specific heat temperature-dependent function number. Enter zero for steady state only.	MCP(I) Phase change flag. Zero implies no phase changes.	Card 4* (F10.5.7) Maximum MAXMAT
QLIN (R) Phase-change or transition temperature.	XLHM (R) Latent heat.	NOTE: This card is only present for materials with change of phase capabilities. There can only be MXECP such materials.							Card 7C (F10.5.7)
OR									
MAT (I) Material number.	MATNAM (A) Material name. Library material number preceded by *.	MUNCH (I) Data unit conversion flag.	XC (R) Unit conversion factor for density.	RC (R) Unit conversion factor for specific heat.	KK(R) Unit conversion factor for conductivity.	KT(R) Unit conversion factor (mult.) for temperature.	ETF(R) Unit conversion factor (add.) for temperature.	MCP(I) Phase change flag. Zero implies no phase changes.	Card 4* (F10.5.7)
INITIAL TEMPERATURES									
INTM (I) Initial temperature function number.	IENFIN (R) Initial temperature, if constant.	ITMD (I) X- or R-dependent function number.	IINTE (I) Y- or Θ-dependent function number.	ITMZ (I) Z-dependent function number.	NOTE: For the 3rd, 4th, and 5th entries, positive integer implies analytical function, negative integer implies tabular function.				Card 5 (F10.5.8) Maximum MAXINT
HEAT GENERATIONS									

NQH (I) Heat generation function number.	QDHO (R) Volumetric heat generation rate, if constant.	NQHTN (I) Time-dependent function number	NQHTP (I) Temperature-dependent function number.	NQHX (I) X- or R-dependent function number.	NQHTX(I) X- or R-dependent function number.	NQHZ(I) Z-dependent function number.	NOTE: For the 1st through 7th entries, positive integer implies analytical function and negative implies tabular function.	Card 4 (F10.5.8) Maximum MAXNOR

BOUNDARY CONDITIONS

NBSTY (I) Boundary number.	NBSTYP (I) Boundary type 1-surface-to-boundary 2-isothermal 3-surface-to-surface	NBSTOW (R) Boundary temperature, T_b Enter zero for NBSTYP=3	NBSTTN (I) Time-dependent function number for boundary temperature NOTE: For the 6th through 7th entries, positive integer implies analytical function and negative integer implies tabular function. All should be zero for NBSTYP=3.	NBSTXN(I) X- or R-dependent function number for boundary temperature.	NBSTTW(I) T- or R-dependent function number for boundary temperature.	NBSTZN(I) Z-dependent function number for boundary temperature.		Card B1 (F10.5.10) Maximum MAXBOC
NBCONV (R) Forced convective heat transfer coefficient. h_c	BKRAD (R) Radiative coefficient. $h_r = F \cdot \sigma \cdot \epsilon$ σ=Stefan-Boltzmann constant. F=Radiation shape factor	NBKAT (R) Natural convective coefficient. h_n	BKKAT (R) Natural convective exponent n_n	NFLUX (R) Prescribed heat flux (positive if going to surface.) q_p	INNFLG(I) Parameter flag. 0-no additional cards 1-B1 card only 2-B5 card only 3-B1 & B5 cards	IFUNTP(I) Fin effectiveness flag. 0-no B6 card 1-10 fin effectiveness tables F10.5.3) 11-user supplied		Card B2 (F10.5.10)
NBKCTN (I) Forced convective heat transfer coefficient time-dependent function.	NBKRTN (I) Radiative coefficient time-dependent function	NBKNTN (I) Natural convective coefficient time-dependent function.	NBKETN (I) Natural convective exponent time-dependent function.	NBKFTN (I) Prescribed heat flux time-dependent function.	Positive integer implies analytical function. Negative integer implies tabular function.			Card B3 (F10.5.10)
NBKCTF (I) Forced convective heat transfer coefficient temperature-dependent function.	NBKRTF (I) Radiative coefficient temperature-dependent function.	NBKNTF (I) Natural convective coefficient temperature-dependent function.	NBKETF (I) Natural convective exponent temperature-dependent function.	NBKFTF (I) Prescribed heat flux temperature-dependent function.	Positive integer implies analytical function. Negative integer implies tabular function.			Card B4 (F10.5.10)
FINLEN (R) Fin height from base surface.	DELBAS (R) Half width of fin at base.	DELTIP (R) Half width of fin at tip.	Φ_1 (R) Fin spacing.	S2 (R) Spine spacing.	NEFAC(R) Fin heat capacity multiplier. Negative implies no adjustment to node. (Default, 1.0)			Card B5 (F10.5.10)

XGRID

XU (R) Smallest X or R grose lattice line dimension.	NX (R) Next X or R grose lattice line dimension.	NX (R) Next X or R grose lattice line dimension.	NOTE: Maximum of MAXGXL.	Card L1 (F10.5.11)
NDMS (I) Number of DIVISIONS between corresponding X or R grose line and the following line.	YDMS (I)	NDMS (I)	NOTE: Must have one less entry than in L1 cards. Maximum of MAXGFL	Card M1 (F10.5.11)

YGRID

YHG (R) Y or R	NOTE: Same as L1 cards except for Y or R direction. Maximum of MAXGXL	Card L2 (F10.5.12)
NDYMS (I) Y or R	NOTE: Same as M1 cards except for Y or R direction. Maximum of MAXGFL	Card M2 (F10.5.12)

ZGRID

ZG (R) Z	NOTE: Same as L1 cards except for Z direction. Maximum of MAXGXL	Card L3 (F10.5.13)
NDZG (I) Z	NOTE: Same as M1 cards except for Z direction. Maximum of MAXGFL	Card M3 (F10.5.13)

APPENDIX F10.A

CONTROL CARDS FOR OAK RIDGE IBM SYSTEM

A version of HEATING6 with the default array sizes and options as shown in Table F10.5.2 is stored on disks at the computing centers at Oak Ridge National Laboratory (ORNL) and Oak Ridge Gaseous Diffusion Plant (ORGDP). The code was compiled on the FORTRAN IV H Extended Compiler with an Optimization Enhancement. This appendix discusses the job control language (JCL) which is required to run this version of HEATING6 and the modifications to the JCL necessary for some of the optional uses of the code. This information will allow the user to solve most of his problems with the code. For additional information concerning JCL, the user is referred to the *CSD Programmer's Notebook* (Ref. 15).

The control cards necessary to use this version of HEATING6 at ORNL are presented in Fig. F10.A-1. The first and second control cards are specific to the ORNL site. The first two cards of Fig. F10.A-2 provide the site specific control cards for the ORGDP site. The job name has a format of UIDXXXXX where UID is the userid assigned to the user, and the characters XXXXX are any characters which make the job unique. The accounting field (ccccc) contains a four- or five-digit charge number. The programmer-name field contains information necessary to mail the job output back to the user.

As an example of the steps required to prepare the JCL for various options, assume the problem to be solved contains 1500 nodes and three boundary conditions. Assume that the thermal conductivity will be evaluated by a user-supplied subroutine. Assume also that the user wishes to save the final temperature distribution for restart purposes and to save the temperature distribution at the standard output time to later generate graphical output. Further assume that 20 printout times have been specified. Otherwise, all default values apply.

Note that the JCL in the examples that follow use the SCALE procedure, i.e., EXEC SCALE. This procedure has a number of default I/O units specified with particular parameters. A user who uses these unit numbers must verify that the default parameters are adequate for their use (see Ref. 15 for parameter information). The safest approach is to use units not in the procedure, i.e., units 7 and 29-77. DD cards which override the SCALE procedure units must precede those not in the procedure.

F10.A-1. Overriding Subroutines in HEATING6

As indicated in Section F10.3.6, the user may wish to supply one or more subroutines to evaluate certain parameters for a specific model. The format of the JCL required to run HEATING6 with user-supplied subroutines overriding default subroutines is presented in Fig. F10.A-2.

```
//jobname JOB (ccccc,IOxxxx),'programmer name',TIME=(xx,y)
//*JOBPARM LINES=xxx
//SCALE EXEC SCALE,REGION=xxxxK
 (Insert DD cards for optional units here)
//GO.SYSIN DD *
#HEATING6
 (Insert data decks here)
END
/*
//
```

Fig. F10.A-1. Format of JCL for Executing HEATING6 through SCALE at ORNL

```
//jobname JOB (ccccc),'programmer name'
//*CLASS CPU95=yyyz,REGION=xxxxK,LINES=xxx,IO=xxx.y
//COMPILE EXEC FORTQC
//FORT.SYSIN DD *
 (Insert FORTRAN deck here)
/*
//SCALE EXEC SCALE,REGION=xxxxK
 INCLUDE MODULES(HEATING6)
 INCLUDE OVLY(HEATING6)
 NAME HEATING6
/*
 (Insert DD cards for optional units here)
//GO.SYSIN DD *
#HEATING6
 (Insert data decks here)
END
/*
//
```

Fig. F10.A-2. Format of JCL for Executing HEATING6 through SCALE at ORGDP While Overriding FORTRAN Routines

F10.A-2. Final Temperature Distribution Output Unit

A data set containing the final temperature distribution from a previous run may be used to restart the calculations. If the user wishes to generate this data set in the event the problem needs to be continued, he must choose a logical unit number, e.g., 7. Then ITPOUT may be set to 7 using the NAMELIST /OPTION/ feature and the type of data set, formatted or unformatted, must be selected using the fifth entry on Card 3. A DD card describing the output unit must appear before the

```
//GO.SYSIN DD *
```

card. If it is desired to punch the final temperature distribution on cards, a positive integer, e.g., 1, will appear as the fifth entry on Card 3, and the DD card

```
//GO.FT07F001 DD SYSOUT=B
```

will be used to define the output unit. On the other hand, if the data set containing the final temperature distribution is to be generated in unformatted form and stored on one of the semipermanent direct access storage devices, a negative integer, e.g., -1, would appear as the fifth entry on Card 3. The DD card

```
//GO.FT07F001 DD DSN=T.uidccccc.xxxxxxxx,UNIT=SPDA,
// DISP=(NEW,CATLG),SPACE=(TRK,(2,1),RLSE),
// DCB=(RECFM=VBS,BLKSIZE=1061)
```

would be used to define output unit 7 where the data set name (DSN=T.uidccccc.xxxxxxxx) must adhere to the conventions described in Chapter 15 of the *CSD Programmer's Notebook* (Ref. 15). The SPACE and DCB parameters were derived using the storage capacity tables for 3330's in Chapter 15 of Ref. 15. In the data set name, uid is the three-character userid of the data set "owner," ccccc is the five-digit charge number of the data set "owner" (if the charge number is less than five digits, zeros on the left should pad it to five digits), and xxxxxxxx is the low-level index consisting of up to eight alphanumeric characters chosen by the user to make the data set name unique. The DCB parameters were derived as follows. As indicated in Sect. F10.3.7.5, three records are written when this data set is created. The first record is 72 bytes in length, the second is 16 bytes, and the third is 12N bytes where N is the number of nodes in the problem. Thus, the RECFM parameter is chosen as VBS. In the example given at the start of this appendix, N is 1500 which means the last record will be 18000 bytes in length. The storage capacity for a single track on a 3330 without keys is 13030 bytes. If the block size (BLKSIZE) is chosen as 1061, 11 records can be written per track; and this data set will require only two tracks as indicated by the SPACE parameter. The RLSE option is specified to release any unused disk space after the data set has been created. The operating system will allocate two buffers for the output unit with each buffer requiring about 1K bytes of core. Thus, the DCB parameters chosen in this example will require an additional 2K bytes of core for execution of HEATING6.

F10.A-3. Graphical Output Unit

A data set which can be used to create graphical output will be generated if the first entry on Card 3 contains a nonzero integer. Entry 2 on Card 3 specifies the frequency of output on this data set (see Sects. F10.5.5.1 and F10.5.5.2 for more details). Assume that the user wishes for this data set to be written on unit 39 for the example described above. Then the user may set IPLOT to 39 using NAMELIST /OPTION/, must set the second entry on Card 3 to a positive integer, e.g., 1, and must leave the second entry blank since the output is desired at each standard printout. The user must supply a DD card of the form

```
//GO.FT39F0001 DD DSN=T.uidccccc.xxxxxxxx,UNIT=SPDA,
//DISP=(NEW,CATLG),SPACE=(TRK,(30,2),RLSE),
//DCB=(RECFM=VBS,BLKSIZE=3156,DSORG=PS)
```

to define output unit 39. Like the example shown in Appendix F10.A.2, this DD card will generate a data set on one of the semipermanent direct access storage devices. The data set name must follow the same conventions as presented in Appendix F10.A.2. The SPACE and DCB parameters were derived as follows. Records 1-5 as shown in Table F10.3.5 contain preliminary information and are written only once. Records 6 and 7 appear in pairs and are written each time the temperature distribution is output. The most frequently written large record is 12028 bytes in length. Since the data set will be created on a 3330 disk, an efficient block size that would not require an excessive amount of core for the buffer space is 3156. Almost one track of disk space will be required for the output of each temperature distribution. There will be 20 regular printouts plus the output of the initial temperature distribution. The preliminary information will require approximately two tracks. Thus, the total disk space required to store the data set will be approximately 25 tracks. In the example, 30 tracks of space were requested with a secondary allocation of two tracks. The RLSE option is specified to release any unused disk space after the data set has been created. Since the records will be variable in length, the RECFM option of VBS has been specified. The operating system will allocate two buffers for output unit 39 with each buffer requiring a little more than 3K bytes of core. Thus, the DCB parameters chosen in this example may require an additional 6K bytes of core for execution of HEATING6.

F10.A-4. Sample JCL

The control cards that are necessary to run the example presented in the first part of this appendix on one of the Oak Ridge IBM computing systems are depicted in Fig. F10.A-3. The total core requirement is approximately 760K bytes (752K for the code, 2K for unit 7, and 6K for unit 39). A total of 770K bytes of core has been requested to give a small cushion of 10K bytes.

Corrected 12/84

```
//jobname JOB (ccccc),'programmer name'
//*CLASS CPU95=yyyz,REGION=xxxxK,LINES=xxx,IO=xxx.y
//COMPILE EXEC FORTQC
//FORT.SYSIN DD *
 (Insert user-supplied FORTRAN subroutine CONDTN here)
/*
//SCALE EXEC SCALE,REGION=xxxxK
//LKED.SYSIN DD *
 INCLUDE MODULES(HEATING6)
 INCLUDE OVLY(HEATING6)
 NAME HEATING6
/*
//GO.FT07F001 DD DSN=T.uidccccc.SAMPTEMP,UNIT=SPDA,
// DISP=(NEW,CATLG),SPACE=(TRK,(2,1),RLSE),
// DCB=(RECFM=VBS,BLKSIZE=1061)
//GO.FT39F001 DD DSN=T.uidccccc.SAMPPLOT,UNIT=SPDA,
// DISP=(NEW,CATLG),SPACE=(TRK,(30,2),RLSE),
// DCB=(RECFM=VBS,BLKSIZE=3156,DSORG=PS)
//GO.SYSIN DD *
#HEATING6
 (Insert data decks here)
END
/*
//
```

Fig. F10.A-3. Format of JCL for Executing HEATING6 through SCALE for the Example Discussed in Appendix F10.A

NUREG/CR-0200
Volume 2, Section F11
ORNL/NUREG/CSD-2/V1/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

KENO V.a
AN IMPROVED MONTE CARLO CRITICALITY PROGRAM WITH SUPERGROUPING

L. M. Petrie
N. F. Landers

Manuscript Completed: March 1983
Date Published: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75

NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

CONTENTS

LIST OF FIGURES F11.vii

LIST OF TABLES F11.xiii

ACKNOWLEDGMENTS F11.xix

ABSTRACT F11.xxi

F11.1 INTRODUCTION TO KENO V.a F11.1.1

F11.2 THEORY AND TECHNIQUES F11.2.1

F11.2.1 THE TRANSPORT EQUATION F11.2.1
F11.2.2 COLLISION TREATMENT F11.2.5
F11.2.3 BIASING OR WEIGHTING F11.2.6
F11.2.4 DIFFERENTIAL ALBEDOS F11.2.9
F11.2.5 SUPERGROUPING F11.2.9
F11.2.6 RESTART F11.2.10
F11.2.7 GEOMETRY F11.2.11
F11.2.8 REFERENCES F11.2.11

F11.3 LOGICAL PROGRAM FLOW F11.3.1

F11.3.1 PROGRAM INITIATION F11.3.1
F11.3.2 PARAMETER DATA F11.3.2
F11.3.3 OVERALL PROGRAM FLOW F11.3.4
F11.3.4 PROBLEM DESCRIPTION F11.3.6

F11.3.4.1 Array Data F11.3.9
F11.3.4.2 Geometry Data F11.3.10
F11.3.4.3 Mixing Table Data F11.3.12
F11.3.4.4 Plot Data F11.3.13
F11.3.4.5 Biasing Data F11.3.14
F11.3.4.6 Restart Data F11.3.15
F11.3.4.7 Generate Remaining Data F11.3.16

F11.3.5 CREATE A MIXED CROSS SECTION DATA FILE F11.3.18

F11.3.5.1 Cross-Section Mixing F11.3.20
F11.3.5.2 Generate Angles and Probabilities F11.3.21

F11.3.6 WRITE CROSS SECTIONS ON DIRECT ACCESS FILE F11.3.22

F11.3.7 WRITE INPUT DATA ON RESTART FILE F11.3.23

F11.3.8 GENERATE ALBEDO CROSS SECTION CORRESPONDENCE TABLES F11.3.24

F11.3.9 GENERATE SUPERGROUPED DATA F11.3.25

F11.3.9.1 Print Macroscopic Cross Sections F11.3.27
F11.3.9.2 Write Supergroup Data File F11.3.28

F11.3.10 PROCESS GEOMETRY F11.3.29

F11.3.10.1 Load Data From the Direct Access File F11.3.31
F11.3.10.2 Check the Geometry Data F11.3.33

F11.3.10.2.1 Determine Distances F11.3.35

F11.3.10.3 Calculate Volumes F11.3.36

F11.3.11 LOAD BIASING OR WEIGHTING DATA F11.3.37

F11.3.12 GENERATE PRINTER PLOT F11.3.38

F11.3.13 PROCESS HISTORIES BY SUPERGROUP F11.3.39

F11.3.13.1 Load Calculated Restart Data F11.3.42
F11.3.13.2 Generate Initial Source Distribution F11.3.43
F11.3.13.3 Track Individual Histories F11.3.46
F11.3.13.4 Provide the Next Generation Source F11.3.54
F11.3.13.5 End of Generation Processing F11.3.55

F11.3.14 END OF PROBLEM PROCESSING F11.3.56

F11.3.15 PRINT FLUXES F11.3.57

F11.3.16 REFERENCES F11.3.58

F11.4 KENO V DATA GUIDE F11.4.1

F11.4.1 KENO V INPUT OUTLINE F11.4.1
F11.4.2 PROCEDURE FOR DATA INPUT F11.4.3
F11.4.3 TITLE AND PARAMETER DATA F11.4.4
F11.4.4 GEOMETRY DATA F11.4.8
F11.4.5 ARRAY DATA F11.4.19
F11.4.6 ALBEDO DATA F11.4.26
F11.4.7 BIASING OR WEIGHTING DATA F11.4.29
F11.4.8 START DATA F11.4.32
F11.4.9 EXTRA 1-D XSECS ID'S DATA F11.4.35
F11.4.10 MIXING TABLE DATA F11.4.36
F11.4.11 PLOT DATA F11.4.37
F11.4.12 REFERENCE F11.4.41

F11.5 NOTES FOR KENO V USERS F11.5.1

F11.5.1 DATA ENTRY F11.5.1
F11.5.2 DEFAULT LOGICAL UNIT NUMBERS FOR KENO V F11.5.2
F11.5.3 PARAMETER INPUT F11.5.3

F11.5.4 CROSS SECTIONS F11.5.3

F11.5.4.1 Use a Premixed Cross-Section Data File F11.5.3
F11.5.4.2 Use an AMPX Working Library F11.5.4
F11.5.4.3 Number of Scattering Angles F11.5.4
F11.5.4.4 Cross-Section Message Cutoff 11.5.4

F11.5.5 MIXING TABLE F11.5.5
F11.5.6 GEOMETRY F11.5.6

F11.5.6.1 Use of Holes in the Geometry F11.5.23
F11.5.6.2 Nesting Holes F11.5.29
F11.5.6.3 Multiple Arrays F11.5.39
F11.5.6.4 Arrays and Holes F11.5.49
F11.5.6.5 Specification of the Global System F11.5.63

F11.5.7 ALTERNATIVE SAMPLE PROBLEM MOCK-UPS F11.5.69

F11.5.7.1 Sample Problem 12, First Alternative F11.5.69
F11.5.7.2 Sample Problem 12, Second Alternative F11.5.71
F11.5.7.3 Sample Problem 13, Alternative F11.5.72

F11.5.8 BIASING OR WEIGHTING DATA F11.5.73
F11.5.9 PRINTER PLOTS F11.5.75
F11.5.10 RESTART CAPABILITIES F11.5.109
F11.5.11 RANDOM SEQUENCE F11.5.110
F11.5.12 MATRIX K-EFFECTIVE F11.5.120
F11.5.13 DEVIATIONS F11.5.121
F11.5.14 GENERATION TIME AND LIFETIME F11.5.121
F11.5.15 REFERENCE F11.5.121

F11.6 DESCRIPTION OF OUTPUT F11.6.1

F11.6.1 HEADER PAGE F11.6.1
F11.6.2 PROGRAM VERIFICATION INFORMATION F11.6.1
F11.6.3 TABLES OF PARAMETER DATA F11.6.2
F11.6.4 UNPROCESSED GEOMETRY INPUT DATA F11.6.4
F11.6.5 TABLE OF DATA SETS USED IN THE PROBLEM F11.6.4
F11.6.6 MIXING TABLE DATA F11.6.5
F11.6.7 ALBEDO CROSS SECTION CORRESPONDENCE F11.6.6
F11.6.8 1-D MACROSCOPIC CROSS SECTIONS F11.6.7
F11.6.9 EXTRA 1-D CROSS SECTIONS F11.6.8
F11.6.10 2-D MACROSCOPIC CROSS SECTIONS F11.6.8
F11.6.11 PROBABILITIES AND ANGLES F11.6.9
F11.6.12 TABLE OF ADDITIONAL INFORMATION F11.6.9
F11.6.13 TABLE OF SPACE AND SUPERGROUP INFORMATION F11.6.11
F11.6.14 ARRAY SUMMARY F11.6.12
F11.6.15 GEOMETRY DATA F11.6.13
F11.6.16 UNIT ORIENTATION DESCRIPTION F11.6.14
F11.6.17 VOLUME INFORMATION F11.6.14

F11.6.18 BIASING INFORMATION F11.6.15
F11.6.19 GROUP-DEPENDENT WEIGHTS F11.6.16
F11.6.20 PICTURE OR PLOT REPRESENTATION F11.6.16
F11.6.21 CONCLUDE PRE-TRACKING EDITS F11.6.18
F11.6.22 PRINT STARTING POINTS F11.6.19
F11.6.23 K-EFFECTIVES BY GENERATION F11.6.20
F11.6.24 FINAL K-EFFECTIVE EDIT F11.6.22
F11.6.25 PLOT OF AVERAGE K-EFFECTIVE BY GENERATION RUN F11.6.23
F11.6.26 PLOT OF AVERAGE K-EFFECTIVE BY GENERATION SKIPPED F11.6.23
F11.6.27 FINAL EDIT OF FISSIONS, ABSORPTIONS, AND LEAKAGE F11.6.24
F11.6.28 MATRIX K-EFFECTIVE BY UNIT LOCATION F11.6.26
F11.6.29 FISSION PRODUCTION BY POSITION INDEX MATRIX F11.6.27
F11.6.30 SOURCE VECTOR BY POSITION INDEX F11.6.27
F11.6.31 COFACTOR K-EFFECTIVE BY POSITION INDEX F11.6.28
F11.6.32 MATRIX K-EFFECTIVE BY UNIT NUMBER F11.6.28
F11.6.33 FISSION PRODUCTION BY UNIT NUMBER MATRIX F11.6.29
F11.6.34 SOURCE VECTOR BY UNIT NUMBER F11.6.29
F11.6.35 COFACTOR K-EFFECTIVE BY UNIT NUMBER F11.6.30
F11.6.36 MATRIX K-EFFECTIVE BY HOLE NUMBER F11.6.30
F11.6.37 FISSION PRODUCTION BY HOLE NUMBER MATRIX F11.6.30
F11.6.38 SOURCE VECTOR BY HOLE NUMBER F11.6.31
F11.6.39 COFACTOR K-EFFECTIVE BY HOLE NUMBER F11.6.31
F11.6.40 MATRIX K-EFFECTIVE BY ARRAY NUMBER F11.6.31
F11.6.41 FISSION PRODUCTION BY ARRAY NUMBER MATRIX F11.6.32
F11.6.42 SOURCE VECTOR BY ARRAY NUMBER F11.6.32
F11.6.43 COFACTOR K-EFFECTIVE BY ARRAY NUMBER F11.6.32
F11.6.44 FISSION DENSITY EDIT F11.6.33
F11.6.45 FLUX EDIT F11.6.33
F11.6.46 FREQUENCY DISTRIBUTIONS F11.6.34

F11.7 WARNING MESSAGES AND ERROR MESSAGES F11.7.1

F11.7.1 MESSAGES F11.7.1
F11.7.2 STOP CODES F11.7.32

F11.A APPENDIX A - ALPHABETICAL INDEX OF SUBROUTINES F11.A.1

F11.B APPENDIX B - ALPHABETICAL INDEX OF COMMONS F11.B.1

F11.C APPENDIX C - KENO-V INPUT SUMMARY F11.C.1

F11.D APPENDIX D - SAMPLE PROBLEMS F11.D.1

F11.D.1 REFERENCES F11.D.14

F11.E APPENDIX E - KENO-V SAMPLE PROBLEM OUTPUT F11.E.1

F11.E.1 SELECTED OUTPUT FROM SAMPLE PROBLEMS F11.E.1
F11.E.2 MICROFICHE OF COMPLETE SAMPLE PROBLEM OUTPUT F11.E.59

F11.F APPENDIX F - MICROFICHE CROSS REFERENCE OF VARIABLES USED IN KENO V F11.F.1

LIST OF FIGURES

Fig. F11.2.1. Analytic estimate of the relationship between WTLOW and the variance, σ_x^2, when WTAVG is 0.5 ..... F11.2.7

Fig. F11.2.2. Analytic estimate of the relationship between WTAVG and the variance, σ_x^2, when WTLOW is 1/6 ..... F11.2.8

Fig. F11.3.1. Flow chart of program initiation F11.3.1

Fig. F11.3.2. Flow chart of parameter reading F11.3.2

Fig. F11.3.3. Flow chart of overall program flow F11.3.4

Fig. F11.3.4. Flow chart of input data reading F11.3.6

Fig. F11.3.5. Flow chart for reading array data F11.3.9

Fig. F11.3.6. Flow chart for reading geometry data F11.3.10

Fig. F11.3.7. Flow chart for reading mixing table data F11.3.12

Fig. F11.3.8. Flow chart for reading plot data F11.3.13

Fig. F11.3.9. Flow chart for reading biasing data F11.3.14

Fig. F11.3.10. Flow chart for reading restart data F11.3.15

Fig. F11.3.11. Flow chart for generating remaining data F11.3.16

Fig. F11.3.12. Flow chart for mixing cross sections F11.3.18

Fig. F11.3.13. Flow chart of mixing operations F11.3.20

Fig. F11.3.14. Flow chart of angle and probability generation F11.3.21

Fig. F11.3.15. Flow chart of mixed cross-section processing F11.3.22

Fig. F11.3.16. Flow chart for writing data on the restart data file F11.3.23

Fig. F11.3.17. Flow chart for generating correspondence arrays F11.3.24

Fig. F11.3.18. Flow chart of super group creation F11.3.25

Fig. F11.3.19. Flow chart for printing macroscopic cross sections F11.3.27

Fig. F11.3.20. Flow chart for loading supergrouped data F11.3.28

Fig. F11.3.21. Flow chart for processing geometry F11.3.29

Fig. F11.3.22. Flow chart for loading data from direct access F11.3.31

Fig. F11.3.23. Flow chart of geometry checking procedure F11.3.33

LIST OF FIGURES (cont.)

Fig. F11.3.24. Flow chart for distance to intersection F11.3.35

Fig. F11.3.25. Flow chart for calculating volumes F11.3.36

Fig. F11.3.26. Flow chart for loading biasing data F11.3.37

Fig. F11.3.27. Flow chart for pictures or plots F11.3.38

Fig. F11.3.28. Flow chart of tracking routines F11.3.39

Fig. F11.3.29. Flow chart for loading calculated restart data F11.3.42

Fig. F11.3.30. Flow chart for providing initial source distribution F11.3.43

Fig. F11.3.31. Flow chart for tracking individual histories F11.3.46

Fig. F11.3.32. Logic flow chart for subroutine TRACK F11.3.49

Fig. F11.3.33. Flow chart for providing the next generation source F11.3.54

Fig. F11.3.34. Flow chart for end of generation processing F11.3.55

Fig. F11.3.35. Flow chart for end of problem processing F11.3.56

Fig. F11.3.36. Flow chart for printing fluxes F11.3.57

Fig. F11.4.1. Examples of geometry not allowed in KENO F11.4.8

Fig. F11.4.2. Examples of correct KENO units F11.4.9

Fig. F11.4.3. Example demonstrating HOLE capability in KENO F11.4.9

Fig. F11.4.4. Example of array construction F11.4.10

Fig. F11.4.5. Example of an array composed of units containing arrays and holes F11.4.10

Fig. F11.4.6. Partially filled hemisphere or hemicylinder less than half full F11.4.15

Fig. F11.4.7. Partially filled hemisphere or hemicylinder more than half full F11.4.15

Fig. F11.5.1. Stack of disks with a square cap F11.5.10

Fig. F11.5.2. Two large cylinders joined axially by a small cylinder F11.5.16

Fig. F11.5.3. Two large cylinders radially connected by a small cylinder F11.5.17

Fig. F11.5.4. KENO V approximations of cylindrical intersections F11.5.17

Fig. F11.5.5. Two small cylinders joined axially by a large cylinder F11.5.20

Fig. F11.5.6. Close-packed rods in an annulus F11.5.24

LIST OF FIGURES (cont.)

Fig. F11.5.7. Annular rods in triangular pitch lattice F11.5.26

Fig. F11.5.8. Configuration using nested holes F11.5.30

Fig. F11.5.9. Complicated geometric arrangement represented by Unit 7 ..... F11.5.31

Fig. F11.5.10. Geometric component represented by Unit 4 ..... F11.5.32

Fig. F11.5.11a. Left half of printer plot F11.5.37

Fig. F11.5.11b. Right half of printer plot F11.5.38

Fig. F11.5.12. x-y plot of mixed array F11.5.42

Fig. F11.5.13. x-z plot of mixed array F11.5.43

Fig. F11.5.14. Two racks of uranium buttons F11.5.45

Fig. F11.5.15. Entire storage array in the room F11.5.47

Fig. F11.5.16. Uranium rods in a cylindrical container F11.5.49

Fig. F11.5.17. x-y slice of uranium rods in a cylindrical container F11.5.52

Fig. F11.5.18. x-z slice of uranium rods in a cylindrical container F11.5.53

Fig. F11.5.19. Typical PWR shipping cask F11.5.54

Fig. F11.5.20. Simple units F11.5.55

Fig. F11.5.21. Quarter section of fuel pin array F11.5.56

Fig. F11.5.22. Quarter section of fuel assembly F11.5.57

Fig. F11.5.23. 2 x 6 array of B4C rods F11.5.58

Fig. F11.5.24. Central array F11.5.58

Fig. F11.5.25. Long B4C rod arrays F11.5.59

Fig. F11.5.26. Central array with long B4C arrays F11.5.60

Fig. F11.5.27. Two fuel assemblies and B4C rods F11.5.60

Fig. F11.5.28. Associated data for single unit mixture map F11.5.78

Fig. F11.5.29. Mixture map of single unit with centered origin F11.5.79

Fig. F11.5.30. Associated data for single unit unit map F11.5.79

LIST OF FIGURES (cont.)

Fig. F11.5.31. Unit map printer plot for single unit with centered origin F11.5.80

Fig. F11.5.32. Associated data for mixture map of single unit with offset origin F11.5.81

Fig. F11.5.33. Mixture map of single unit with offset origin F11.5.82

Fig. F11.5.34. Associated data for unit map of single unit with offset origin F11.5.83

Fig. F11.5.35. x-y plot of 2x2x2 bare array F11.5.84

Fig. F11.5.36. x-z plot of 2x2x2 bare array F11.5.85

Fig. F11.5.37. Plot data for x-y slice of example 4 ..... F11.5.87

Fig. F11.5.38. x-y plot of 2x2x2 reflected array F11.5.87

Fig. F11.5.39. Plot data for enlarged x-y slice of example 4 ..... F11.5.88

Fig. F11.5.40. Enlarged x-y plot of 2x2x2 reflected array F11.5.88

Fig. F11.5.41. Plot data for x-z slice of example 4 ..... F11.5.89

Fig. F11.5.42. x-z plot for 2x2x2 reflected array F11.5.90

Fig. F11.5.43. Plot data for enlarged x-z slice of example 4 ..... F11.5.91

Fig. F11.5.44. Enlarged x-z plot of 2x2x2 reflected array F11.5.92

Fig. F11.5.45. Plot data for x-y slice of example 5 ..... F11.5.95

Fig. F11.5.46. x-y plot of 2x2x2 reflected array with centered origin F11.5.95

Fig. F11.5.47. Plot data for an enlarged x-y slice of example 5 ..... F11.5.96

Fig. F11.5.48. Enlarged x-y plot of 2x2x2 reflected array with centered origin F11.5.96

Fig. F11.5.49. Plot data for x-z slice of example 5 ..... F11.5.97

Fig. F11.5.50. x-z plot of reflected 2x2x2 array with centered origin F11.5.98

Fig. F11.5.51. Plot data for enlarged x-z slice of example 5 ..... F11.5.99

Fig. F11.5.52. Enlarged x-z plot of reflected 2x2x2 array with centered origin F11.5.100

Fig. F11.5.53. Mixture map of nested holes problem F11.5.104

Fig. F11.5.54. Plot data for unit map of nested holes F11.5.104

Fig. F11.5.55. Unit map of nested holes problem F11.5.105

LIST OF FIGURES (cont.)

Fig. F11.5.56. Plot data for x-z slice of storage array F11.5.106

Fig. F11.5.57. x-z plot of storage array F11.5.107

Fig. F11.5.58. Plot data for x-y slice of storage array F11.5.108

Fig. F11.5.59. x-y plot of storage array F11.5.108

Fig. F11.6.1. Sample KENO V header page F11.6.1

Fig. F11.6.2. Sample program verification table F11.6.2

Fig. F11.6.3. Sample table of numeric parameter data F11.6.2

Fig. F11.6.4. Sample table of logical parameter data F11.6.3

Fig. F11.6.5. Example of unprocessed geometry input data F11.6.3

Fig. F11.6.6. Sample table of data sets used in the problem F11.6.5

Fig. F11.6.7. Example of mixing table data F11.6.6

Fig. F11.6.8. Cumulative probabilities for correlating the albedo energy group to the cross-section energy group F11.6.6

Fig. F11.6.9. Cumulative probabilities for correlating the cross-section energy group to the albedo energy group F11.6.7

Fig. F11.6.10. Example of macroscopic 1D cross sections F11.6.7

Fig. F11.6.11. Example of 2-D macroscopic cross sections F11.6.8

Fig. F11.6.12. Example of macroscopic probabilities F11.6.9

Fig. F11.6.13. Example of macroscopic angles F11.6.9

Fig. F11.6.14. Sample table of additional information F11.6.10

Fig. F11.6.15. Sample table of space and supergroup information F11.6.12

Fig. F11.6.16. Example of array summary F11.6.12

Fig. F11.6.17. Example of geometry region data F11.6.13

Fig. F11.6.18. Example of unit orientation description F11.6.14

Fig. F11.6.19. Sample volume information F11.6.15

Fig. F11.6.20. Biasing information F11.6.16

Fig. F11.6.21. Example of biasing data F11.6.16

LIST OF FIGURES (cont.)

Fig. F11.6.22. Summary of picture symbols and coordinates F11.6.17

Fig. F11.6.23. Sample plot representation F11.6.17

Fig. F11.6.24. Pre-tracking edits F11.6.18

Fig. F11.6.25. Example of initial source points F11.6.19

Fig. F11.6.26. Example of k-effectives by generation F11.6.21

Fig. F11.6.27. Example of the final k-effective edit F11.6.22

Fig. F11.6.28. Sample plot of average k-effective by generation run F11.6.23

Fig. F11.6.29. Sample plot of average k-effective by generation skipped F11.6.24

Fig. F11.6.30. Sample of the final edit of fissions, absorptions, and leakage with region-dependent information suppressed F11.6.25

Fig. F11.6.31. Sample of the final edit of fissions, absorptions, and leakage by region F11.6.26

Fig. F11.6.32. Example of matrix k-effective by unit location F11.6.26

Fig. F11.6.33. Sample fission production matrix by position index F11.6.27

Fig. F11.6.34. Example of source vector by position index F11.6.27

Fig. F11.6.35. Example of cofactor k-effective by position index F11.6.28

Fig. F11.6.36. Example of matrix k-effective by unit number F11.6.28

Fig. F11.6.37. An example of the fission probability matrix by unit F11.6.29

Fig. F11.6.38. Example of the source vector by unit F11.6.29

Fig. F11.6.39. Example of cofactor k-effective by unit number F11.6.30

Fig. F11.6.40. Example of matrix k-effective by hole number F11.6.30

Fig. F11.6.41. Example of fission production matrix by hole F11.6.30

Fig. F11.6.42. Example of source vector by hole number F11.6.31

Fig. F11.6.43. Example of cofactor k-effective by hole number F11.6.31

Fig. F11.6.44. Example of matrix k-effective by array number F11.6.31

Fig. F11.6.45. An example of the fission production matrix by array number F11.6.32

LIST OF FIGURES (cont.)

Fig. F11.6.46. Example of source vector by array number F11.6.32

Fig. F11.6.47. Example of cofactor k-effective by array number F11.6.32

Fig. F11.6.48. Example of the fission density edit F11.6.33

Fig. F11.6.49. An example of a flux edit F11.6.33

Fig. F11.6.50. Example of the frequency distributions F11.6.34

Fig. F11.D.1. Critical 2C8 bare assembly F11.D.20

Fig. F11.D.2. Half of the paraffin reflected 2C8 assembly before the top reflector was added F11.D.21

Fig. F11.D.3. Critical assembly of 4 solution units and 4 metal units F11.D.22

Fig. F11.D.4. Drawing of two cuboids in an annulus critical assembly F11.D.23

Fig. F11.D.5. Drawing of the cylinder in an annulus critical assembly F11.D.23

Fig. F11.D.6. Drawing of a critical assembly consisting of a uranium sphere on a plexiglas collar with a cylindrical water reflector F11.D.24

Fig. F11.D.7. Perspective of critical 1F27 experiment F11.D.25

Fig. F11.D.8. View of a 27-unit cubic array with 2.54-cm-thick plexiglas reflector on five sides and a 15.24-cm-thick paraffin base F11.D.26

Fig. F11.D.9. Typical arrangement for critical experiments with interacting arrays of aluminum cylinders containing enriched ^{235}U solutions F11.D.27

Fig. F11.D.10. Schematic of bare partially filled sphere experiment inside a 9.5-ft-diameter, 9-ft-high steel tank F11.D.28

LIST OF TABLES

Table F11.4.1. Types of input data F11.4.1

Table F11.4.2. FIDO like input for mixed box orientation fill option F11.4.23

Table F11.4.3. Face codes for entering boundary (albedo) conditions F11.4.27

Table F11.4.4. Albedo names available on the KENO V albedo library for use with the face codes F11.4.28

Table F11.4.5. ID's, group structure and incremental thicknesses for weighting data available on the weighting library F11.4.30

Table F11.4.6. Starting distributions available in KENO V F11.4.33

Table F11.4.7. Summary of parameter data F11.4.43

Table F11.4.8. Summary of array data F11.4.44

Table F11.4.9. Summary of biasing data F11.4.45

Table F11.4.10. Summary of boundary condition data F11.4.46

Table F11.4.11. Summary of geometry data F11.4.47

Table F11.4.12. Summary of mixing table data F11.4.49

Table F11.4.13. Summary of plot data F11.4.50

Table F11.4.14. Summary of starting data F11.4.51

Table F11.5.1. KENO V Logical Unit Numbers F11.5.2

Table F11.5.2. Structure of RESTART file F11.5.111

Table F11.5.3. Key of RESTART file variables F11.5.115

Table F11.C.1. Summary of parameter data F11.C.2

Table F11.C.2. Summary of array data F11.C.3

Table F11.C.3. Summary of biasing data F11.C.4

Table F11.C.4. Summary of boundary condition data F11.C.5

Table F11.C.5. Summary of geometry data F11.C.6

Table F11.C.6. Summary of mixing table data F11.C.8

Table F11.C.7. Summary of plot data F11.C.9

Table F11.C.8. Summary of starting data F11.C.10

Table F11.D.1. KENO 16-group Hansen-Roach cross-section library F11.D.15

ACKNOWLEDGMENTS

Many individuals have contributed significantly to the development of KENO V. Special recognition is given to G. E. Whitesides, head of Technical Applications, who was responsible for the concept and development of the original KENO code. He has contributed significantly to some of the techniques utilized in KENO V. S. K. Fraley, formerly of the Nuclear Engineering Applications Department, is acknowledged for his assistance in implementing the mechanics of supergrouping. J. T. Thomas has contributed many ideas that have been implemented in the code. R. M. Westfall has provided consultation, encouragement and benchmarks for validating the code. The special abilities of J. R. Knight contributed substantially to debugging the code. Appreciation is expressed to C. V. Parks and J. C. Turner for their assistance in reviewing this document.

Special appreciation is expressed to J. C. White of Y-12 and W. R. Golliher of K-25 for providing funding for the development of the "array of arrays" and "holes" options which have significantly enhanced the geometric capabilities of the code.

The contributions of R. J. Patton and L. Norris in preparing this document are gratefully acknowledged.

ABSTRACT

KENO V.a is an extension of the KENO-V Monte Carlo criticality program and was developed for use in the SCALE system. In addition to the features available in KENO V, KENO V.a offers versatile new geometry capabilities and in-line printer plots of the geometry. The new geometry features include (1) the array of arrays option, (2) the holes option, and (3) variable chords for hemicylinders and hemispheres.

The array of arrays options allows arrays to be built of other arrays and nested to any depth, subject to the availability of sufficient computer memory. The holes option allows the placement of one or more geometry regions within other geometry regions. The depth of hole nesting is limited only by the availability of sufficient computer memory. The variable chords option allows a hemisphere or hemicylinder to be as small as non-existent or as large as a full sphere or cylinder, or any size in between.

The primary purpose of KENO V.a is to determine k-effective. Other calculated quantities include lifetime and generation time, energy-dependent leakages, energy- and region-dependent absorptions, fissions, fluxes, and fission densities.

KENO V.a retains the KENO V features such as flexible data input, the ability to specify origins for cylindrical and spherical geometry regions, the capability of supergrouping energy-dependent data, a P_s scattering model in the cross sections, a procedure for matching lethargy boundaries between albedos and cross sections to extend the usefulness of the albedo feature, and improved restart capabilities.

This advanced user-oriented program features simplified data input and efficient use of computer storage. This allows the user to readily solve large problems whose computer storage requirements and geometric complexity precluded solution when using older versions of KENO.

F11.1 INTRODUCTION TO KENO V.a

This report uses the terms KENO V.a and KENO V to refer to the KENO V.a program. KENO V.a, a functional module in the SCALE system, is a multigroup Monte Carlo criticality program used to calculate the k-effective of a three-dimensional system. Special features include simplified data input, supergrouping of energy-dependent data, the ability to specify origins for spherical and cylindrical geometry regions, a P_n scattering treatment, extended use of differential albedo reflection, and an improved restart capability.

The KENO V data input features flexibility in the order of input. The single restriction is that the title must be entered first and the parameter data, if any, must immediately follow. A large portion of the data has been assigned default values that have been found to be adequate for many problems. This enables the user to run a problem with a minimum of input data.

Blocks of input data are entered in the form:

READ XXXX input data END XXXX

where XXXX is the keyword for the type of data being entered. The types of data entered include parameters, geometry region data, array definition data, biasing or weighting data, albedo boundary conditions, starting distribution information, the cross-section mixing table, extra 1-D (reaction rate) cross-section ID's for special applications, and printer plot information.

A block of data can be omitted unless it is needed or desired for the problem. Within the blocks of data, most of the input is activated by using keywords to override the default values.

The geometry input is very similar to that of KENO IV, except the specification of the biasing data has been rearranged to minimize storage requirements. An important improvement is the ability to specify the origin for cylinders, hemicylinders, spheres, and hemispheres. This allows the use of nonconcentric cylindrical and spherical shapes and provides a great deal of freedom in positioning them. Another improvement allows for hemicylinders and hemispheres whose cut surface can be placed at any distance between the radius and the origin.

An additional geometry convenience is the availability of an alternative method for specifying the array definition (mixed box or unit orientation) data. This method utilizes FIDO-like options for filling the array.

The most outstanding KENO geometry advancement is the addition of the "array of arrays" and "holes" capabilities. The array of arrays option allows the construction of arrays from other arrays. The depth of nesting is limited only by computer space restrictions. This option greatly simplifies the setup for arrays involving different units at different spacings. The hole option allows placing a unit or an array at any desired location within a geometry region. The emplaced unit or array cannot intersect any geometry region and must be wholly contained within a region. As many holes as will snugly fit without intersecting can be placed in a region. This option is especially useful for describing shipping casks and reflectors that have gaps or other geometrical features. Any number of holes can be described in a problem and holes can be nested to any depth.

An important feature of KENO V is the capability of supergrouping the energy-dependent information such as cross sections and fluxes. This automatic feature is activated when the computer storage is insufficient to hold the entire problem at once. The energy-dependent data are then broken into supergroups that are written on a direct access device and moved in and out of memory as necessary. Thus larger problems can be run on smaller computers.

Anisotropic scattering is treated by using discrete scattering angles. The angles and associated probabilities are generated in a manner that preserves the moments of the angular scattering distribution for the selected group-to-group transfer. These moments can be derived from the coefficients of a P_n Legendre polynomial expansion. All moments through the 2n-1 moment are preserved for n discrete scattering angles. A one-to-one correspondence exists such that n Legendre coefficients yield n moments. The cases of zero and one scattering angle are treated in a special manner. KENO V can recognize that the distribution is isotropic even if the user specifies multiple scattering angles, and therefore selects from a continuous isotropic distribution. If the user specifies one scattering angle, the

code performs semi-continuous scattering by picking scattering angle cosines uniformly over some range between -1 and +1. The probability is zero over the rest of the range.

Differential albedos are available to simulate tracking in a reflector. These albedos were generated using the Hansen-Roach 16 energy group structure. KENO V can extend the use of these albedos to include other energy group structures. This is done by matching lethargy boundaries between the albedos and cross sections so the appropriate energy transfers can be made. Lethargy boundary tables are created for both the albedo and cross-section energy group structures, and the lethargy interval corresponding to the desired transfer is determined based on a uniform distribution over the lethargy interval. Approximations must be made when the energy group boundaries of the albedos and cross sections are different. Therefore the user should scrutinize the results to evaluate the effects of the approximations until an adequate information base is established.

The KENO V restart option is easy to activate. Certain changes can be made when a problem is restarted, including the use of a different random sequence and turning off certain print options such as fluxes or the fissions and absorptions by region.

F11.2 THEORY AND TECHNIQUES

F11.2.1. THE TRANSPORT EQUATION

The equation KENO V solves may be derived in the following manner, starting with the Boltzmann neutron transport equation which may be written as[1]

$$\frac{1}{v}\frac{\partial\Phi}{\partial t}(X,E,\Omega,t) + \Omega\cdot\nabla\Phi(X,E,\Omega,t) + \Sigma_t(X,E,\Omega,t)\Phi(X,E,\Omega,t) = S(X,E,\Omega,t)$$

$$+ \int_{E'}\int_{\Omega'}\Sigma_s(X,E'\rightarrow E,\Omega'\rightarrow\Omega,t)\Phi(X,E',\Omega',t)d\Omega' dE' \quad , \tag{F11.2.1}$$

where:

$\Phi(X,E,\Omega,t)$ = neutron flux (neutrons/cm^2/sec) per unit energy at energy E per steradian about direction Ω at position X at time t moving at speed v corresponding to E,

$\Sigma_t(X,E,\Omega,t)$ = macroscopic total cross section of the media (cm^{-1}) at position X, energy E, direction Ω and time t,

$\Sigma_s(X,E'\rightarrow E,\Omega'\rightarrow\Omega,t)$ = macroscropic differential cross section of the media (cm^{-1}) per unit energy at energy E' per steradian about direction Ω' at position X, and time t, for scattering to energy E and direction Ω,

$S(X,E,\Omega,t)$ = neutrons/cm^3/sec born at position X and time t per unit energy at energy E per steradian about direction Ω (excludes scatter source).

Defining $q(X,E,\Omega,t)$ as the total source resulting from the external source, scattering, fission and all other contributions, the following relationship can be written.

$$q(X,E,\Omega,t) = S(X,E,\Omega,t)$$

$$+ \int_{E'}\int_{\Omega'}\Sigma_s(X,E'\rightarrow E,\Omega'\rightarrow\Omega,t)\Phi(X,E',\Omega',t)d\Omega' dE' \quad . \tag{F11.2.2}$$

Combining Eqs. (F11.2.1) and (F11.2.2), assuming the media to be isotropic, ignoring the time dependence of the cross sections and converting the equation to multigroup form yields

$$\frac{1}{v_g}\frac{\partial\Phi_g}{\partial t}(X,\Omega,t) + \Omega\cdot\nabla\Phi_g(X,\Omega,t) + \Sigma_{t_g}(X)\Phi_g(X,\Omega,t) = q_g(X,\Omega,t) \quad , \tag{F11.2.3}$$

where:

g is the energy group of interest,

v_g is the average velocity of the neutrons in group g,

$\Phi_g(X,\Omega,t)$ is the angular flux of neutrons having their energies in group g, at position X and time t,

$\Sigma_{tg}(X)$ is the macroscopic total cross section of the media at position X for group g, corresponding to

$$\Sigma_{tg}(X) + \frac{\int_{\Delta Eg} \Sigma_t(X,E)\Phi(X,E,\Omega,t)dE}{\int_{\Delta Eg} \Phi(X,E,\Omega,t)dE} ,$$

where ΔEg defines group g, and

$q_g(X,\Omega,t)$ is the total source contributing to energy group g at position X, and time t in direction Ω.

Utilizing the relationship $X'=X-R\Omega$ defining the problem to be time-independent, using an integrating factor[2] on both sides of Eq. (F11.2.3), and defining

$$T(R) = \int_0^R \Sigma_t(X-R'\Omega)dR' ,$$

the following equation can be written.

$$\Phi_g(X,\Omega) = \int_0^\infty q_g(X-R\Omega,\Omega)e^{-T(R)}\, dR . \qquad \text{(F11.2.4)}$$

At this point, the problem becomes an eigenvalue problem. If there is no external source, the source may be defined as

$$q_g(X,\Omega) = \sum_{g'}\int d\Omega'\Phi_{g'}(X,\Omega')\Sigma_s(X,g'\rightarrow g,\Omega'\cdot\Omega) + \frac{1}{k}Q_g'(X,\Omega) . \qquad \text{(F11.2.5)}$$

where:

k is the largest eigenvalue of the integral equation,

$Q_g'(X,\Omega)$ is the fission source at position X for energy group g and direction Ω (all fission contributions to group g from all energy groups in the previous generation),

$\Sigma_s(X,g'\rightarrow g,\Omega'\cdot\Omega)$ is the scattering cross section for scattering at position X from group g' and direction Ω' to group g and direction Ω.

In terms of energy, the scatter can be defined as

$$\Sigma_s(X,g'\rightarrow g,\Omega'\cdot\Omega) = \frac{\int_{\Delta E_g}\int_{\Delta E_{g'}}\Sigma_s(X,E'\rightarrow E,\Omega'\cdot\Omega)\Phi(X,E',\Omega')dE'dE}{\int_{\Delta E_{g'}}\Phi(X,E',\Omega')dE'} \quad , \tag{F11.2.6}$$

where:

ΔE_g is the energy range defining energy group g and

$\Delta E_{g'}$ is the energy range defining energy group g'.

Assuming the fission neutrons to be isotropic, the fission source $Q_{g'}(X,\Omega)$ can be written as

$$Q_{g'}(X,\Omega) = \frac{1}{4\pi}\sum_{g'}\int_{\Omega'}d\Omega'\Phi_{g'}(X,\Omega')\chi(X,g'\rightarrow g)\upsilon_{g'}(X)\Sigma_{fg'}(X) \quad , \tag{F11.2.7}$$

where:

$\chi(X,g'\rightarrow g)$ is the fraction of neutrons born in energy group g from fission in energy group g' in the media at position X,

$\upsilon_{g'}(X)$ is the number of neutrons resulting from a fission in group g' at position X,

$\Sigma_{fg'}(X)$ is the macroscopic fission cross section of the material at position X for a neutron in energy group g'.

Substituting Eq. (F11.2.5) into Eq. (F11.2.4) yields the following equation:

$$\Phi_g(X,\Omega) = \int_0^\infty dR e^{-T(R)}\left\{\frac{1}{k}Q'_g(X-R\Omega,\Omega) + \sum_{g'}\left[\int_{\Omega'}d\Omega'\Phi_{g'}(X-R\Omega,\Omega')\Sigma_s(X-R\Omega,g'\rightarrow g,\Omega'\cdot\Omega)\right]\right\} . \tag{F11.2.8}$$

The definition of k may be given as the ratio of the number of neutrons in the (n+1)th generation to the number of neutrons in the nth generation or the largest eigenvalue of the integral equation. Using Eq. (F11.2.7), Eq. (F11.2.8) can be written as

$$\Phi_g(X,\Omega) = \int_0^\infty dR e^{-T(R)}\left\{\sum_{g'}\frac{1}{k}\int_{\Omega'}\upsilon_{g'}(X-R\Omega)\Sigma_{fg'}(X-R\Omega)\chi(X-R\Omega,g'\rightarrow g)\Phi_{g'}(X-R\Omega,\Omega')\frac{d\Omega'}{4\pi} + \sum_{g'}\int_{\Omega'}d\Omega'\Sigma_{tg'}(X-R\Omega,\Omega')\frac{\Sigma_s(X-R\Omega,g'\rightarrow g,\Omega'\cdot\Omega)}{\Sigma_{tg'}(X-R\Omega)}\right\} . \tag{F11.2.9}$$

Writing Eq. (F11.2.9) in "generation notation," multiplying and dividing certain terms by $\Sigma_{t_g}(X)$ and multiplying both sides of the equation by $\upsilon_g(X)\Sigma_{f_g}(X)$, yield the following equation, which is solved by KENO.

$$\frac{\upsilon_g(X)\Sigma_{fg}(X)}{\Sigma_{tg}(X)}\ \Sigma_{tg}(X)\Phi_{g,n}(X,\Omega) = \frac{\upsilon_g(X)\Sigma_{fg}(X)}{\Sigma_{tg}(X)}\Sigma_{tg}(X)\int_0^\infty dR e^{-T(R)}$$

$$\left\{\frac{1}{k}\sum_{g'}\int_{\Omega'}\frac{\upsilon_{g'}(X-R\Omega)\Sigma_{fg'}(X-R\Omega)}{\Sigma_{tg'}(X-R\Omega)}\ \chi(X-R\Omega,g'\rightarrow g)\Sigma_{tg'}(X-R\Omega)\Phi_{g',n-1}(X-R\Omega,\Omega')\frac{d\Omega'}{4\pi}\right.$$

$$\left. +\ \sum_{g'}\int_{\Omega'}\frac{\Sigma_s(X-R\Omega,g'\rightarrow g,\Omega'\cdot\Omega)}{\Sigma_{tg'}(X-R\Omega)}\Sigma_{tg'}(X-R\Omega)\Phi_{g',n}(X-R\Omega,\Omega')d\Omega'\right\}\ , \qquad \text{(F11.2.10)}$$

where n indicates the nth generation and n-1 is the (n-1)th generation. Note that the left-hand side of the equation, $\upsilon_g(X)\ \Sigma_{fg}(X)\Phi_{g,n}(X,\Omega)$, is the fission production for the nth generation.

The solution strategy utilized by KENO V solves Eq. (F11.2.10) by using an iterative procedure. The fission production at point X in energy group g due to neutrons in the (n-1)th generation, normalized to the system multiplication, is

$$\frac{1}{k}\sum_{g'}\int_{\Omega'}\frac{\upsilon_{g'}(X)\Sigma_{fg'}(X)}{\Sigma_{tg'}(X)}\chi(X,g'\rightarrow g)\Sigma_{tg'}(X)\Phi_{g',n-1}(X,\Omega')\frac{d\Omega'}{4\pi}\ .$$

The collision points used in KENO are chosen by selecting path lengths from the distribution

$$e^{-T(R)}$$

which is the probability of transport from any position $X-R\Omega$ to position X.

The first collision density of neutrons in group g per unit solid angle about Ω resulting from the fission source produced by the (n-1) generation, normalized to the system multiplication, is

$$\Sigma_{tg}(X)\int_0^\infty dR e^{-T(R)}\ \frac{1}{k}\int_{\Omega'}\sum_{g'}\frac{\upsilon_{g'}(X-R\Omega)\Sigma_{fg'}(X-R\Omega)}{\Sigma_{tg'}(X-R\Omega)}$$

$$\chi(X-R\Omega,g'\rightarrow g)\Sigma_{t_g}(X-R\Omega)\Phi_{g',n-1}(X-R\Omega,\Omega')\frac{d\Omega'}{4\pi}\ .$$

The scattering source at position X emerging in group g and direction Ω resulting from previous collisions in the same generation, is

$$\sum_{g'} \int_{\Omega'} \frac{\Sigma_s(X,g' \rightarrow g,\Omega' \cdot \Omega)}{\Sigma_{tg'}(X)} \Sigma_{t_{g'}}(X)\Phi_{g',n}(X,\Omega')d\Omega' \quad .$$

The collision density in group g, per solid angle about Ω is

$$\Sigma_{tg} \int_0^{\infty} dR e^{-T(R)} \sum_{g'} \int_{\Omega} \frac{\Sigma_s(X-R\Omega,g' \rightarrow g,\Omega' \cdot \Omega)}{\Sigma_{tg'}(X-R\Omega)} \Sigma_{tg'}(X-R\Omega)\Phi_{g',n}(X-R\Omega,\Omega')d\Omega' \quad .$$

The total collision density times $\frac{\upsilon_g(X)\Sigma_{fg}(X)}{\Sigma_{tg}(X)}$ is the relationship from which KENO V picks the source points for the next generation.

F11.2.2 COLLISION TREATMENT

A collision occurs in a geometrical region when a history exhausts its mean free path length within the boundaries of the region. For each collision, the absorbed weight and the fission weight are tabulated, then the weight is modified by the nonabsorption probability. This new weight is checked for splitting and Russian roulette, and if it survives, the history is scattered. A new energy group is selected from the cumulative transfer probability distribution. This group-to-group transfer determines an angular scattering distribution, usually expressed as a Legendre expansion of the cross section transfer array. A set of discrete angles and probabilities are generated by a generalized Gaussian quadrature procedure,[3] preserving the moments of the Legendre expansion of the angular scattering distribution. KENO V treats P_0 *and* P_1 Legendre expansions as special cases. If the scattering distribution is isotropic, a flag is set to randomly select new direction cosines from an isotropic distribution, instead of using discrete scattering angles. If the distribution is a P_1 expansion, KENO V randomly selects the cosine of the scattering angle according to

$$\mu = \zeta(1-|\bar{\mu}|) + \bar{\mu} \quad ,$$

where ζ is a uniform random variable between -1 and +1
and $\bar{\mu}$ is the mean cosine of the scattering angle.

Otherwise, KENO V randomly selects one of the discrete scattering angles (μ). New direction cosines are then calculated according to the following relationships where u, v, and w are the initial direction cosines and u',v', and w' are the direction cosines after the collision.

$$u' = u\cos\psi - \sqrt{v^2+w^2}\sin\psi\cos\eta$$

$$v' = v\cos\psi + \frac{uv}{\sqrt{v^2+w^2}}\cos\eta\sin\psi - \frac{w}{\sqrt{v^2+w^2}}\sin\psi\sin\eta \quad ,$$

$$w' = w\cos\psi + \frac{uw}{\sqrt{v^2+w^2}}\cos\eta\sin\psi + \frac{v}{\sqrt{v^2+w^2}}\sin\psi\sin\eta \quad ,$$

where:

$\sin\psi = \sqrt{1-\mu^2}$,

$\cos\psi = \mu =$ cosine of the scattering angle,

$\eta =$ a random azimuthal angle between 0 and 2π.

F11.2.3 BIASING OR WEIGHTING

In order to minimize the statistical deviation of k-effective per unit computer time, KENO V utilizes weighted tracking rather than analog tracking. Weighted tracking accounts for absorption by reducing the neutron weight, rather than allowing the neutron history to be terminated by absorption. To prevent expending excessive computer time tracking low weight neutrons, Russian roulette is played when the weight of the neutron drops below a preset weight, WTLOW. Neutrons which survive Russian roulette are assigned a weight, WTAVG. The value of WTLOW and WTAVG can be assigned as a function of position and energy. The values used by KENO V are:

DWTAV = 0.5	is the default value of WTAVG,
WTAVG = DWTAV	is the weight given a neutron that survives Russian roulette, and
WTLOW = WTAVG/3.0	is the value of weight at which Russian roulette is played.

A study[4] by Hoffman shows these default values to be reasonable for bare critical assemblies. Figure F11.2.1, from this study, shows the analytic relationship between the variance and WTLOW when WTAVG is 0.5. Note that the default value of 0.167 for WTLOW is very close to the minimum point on the curve. Experimental results of actual Monte Carlo calculations[5] provide further assurance that 0.167 is an optimum choice for WTLOW when WTAVG is 0.5.

Figure F11.2.2, also from the Hoffman study, shows the analytic relationship between the variance and the value chosen for WTAVG for a value of WTLOW=0.167. Although the KENO default value for WTAVG is not the optimum, a close examination of the data shows the variance to be changing relatively slow as a function of WTAVG. While this study shows a value near 0.26 to be optimum for this system, further studies of other systems are needed before changing the default value of WTAVG from the 0.5 that has been used in previous versions of KENO.

Inside a fissile core, the importance of a neutron is a slowly varying function in terms of energy and position. Hence, for many systems, the standard defaults for WTLOW and WTAVG are good values to use. For reflectors, however, the worth of a neutron varies both as a function of distance from the fissile material and as a function of energy. As a neutron in the reflector becomes less important relative to a neutron in the fissile region, it becomes desirable to spend less time tracking it. Therefore a space- and energy-dependent weighting or biasing function is used in KENO V to allow the user to minimize the variance in k-effective per unit tracking time. When a biasing function is used in a reflector, it becomes possible for a neutron to move from one importance region into another whose WTLOW is greater than the weight of the neutron. When this occurs, Russian roulette is played to reduce the number of neutrons tracked. When the reverse occurs, that is, the neutron moves to a region of higher importance, its weight may be much higher than WTAVG for that region. When the weight of the neutron is greater than a preset value, WTHI, the neutron is split into two neutrons, each having a

weight equal to one-half the weight of the original neutron. This procedure is repeated until the weight of the split neutron is less than WTHI. The default value for WTHI is WTAVG*3.0. WTHI is the weight at which splitting occurs.

The weighting or biasing function for a given core material and reflector material can be obtained by using the adjoint solution from S_n type programs for a similar (usually simplified) problem. This adjoint flux gives the relative contribution of a neutron at a given energy and position to the total fissions in the system. The weighting function for KENO is thus proportional to the reciprocal of the adjoint flux. Although such a function can be difficult to obtain, the savings gained makes the effort worthwhile for many of the materials that are frequently used as reflectors. Biasing functions have been prepared for several reflector materials commonly used in KENO calculations. The use of biasing to minimize the variance in k-effective per unit computer time will usually increase the variance in other parameters such as leakage or absorption in the reflector.

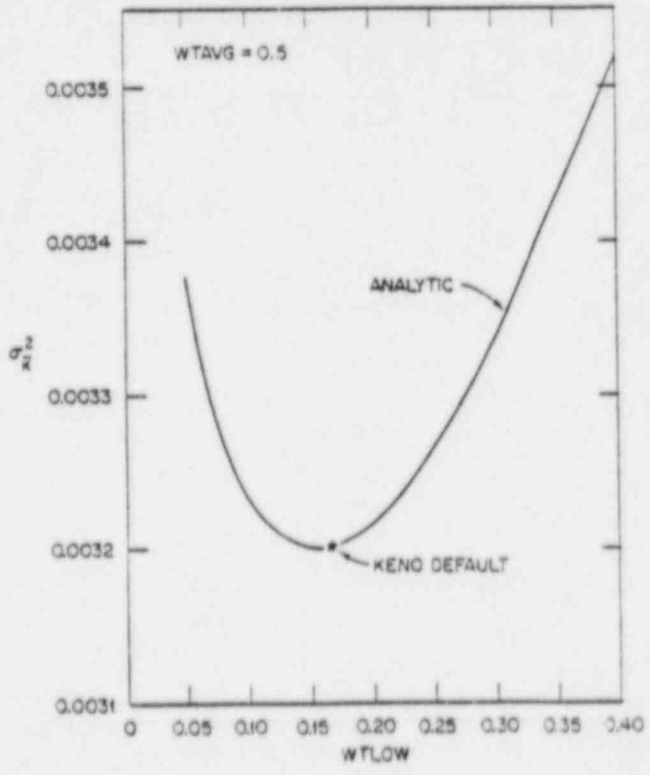


Fig. F11.2.1. Analytic estimate of the relationship between WTLOW and the variance, σ_k^2, when WTAVG is 0.5.

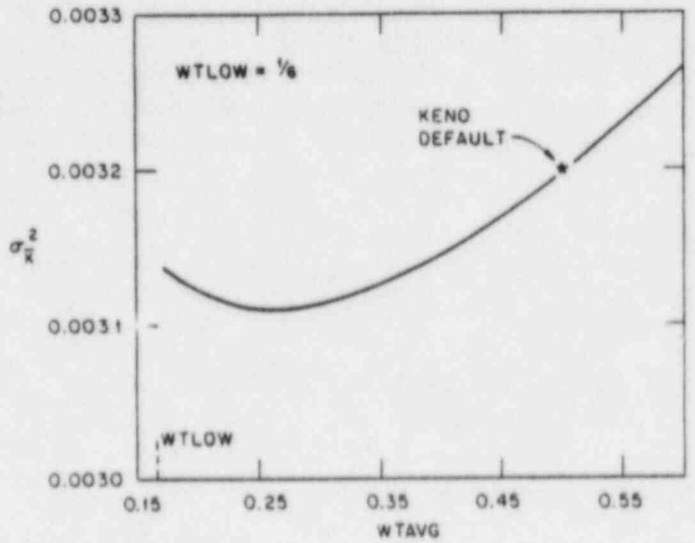


Fig. F11.2.2. Analytic estimate of the relationship between WTAVG and the variance, σ_x^2, when WTLOW is 1/6.

F11.2.4 DIFFERENTIAL ALBEDOS

Arrays reflected by thick layers of material having a small absorption to scattering ratio may require large amounts of computer time to determine k-effective because of the relatively long time a history may spend in the reflector. A differential albedo technique was developed for use with the KENO codes to eliminate tracking in the reflector. This involves returning a history at the point it impinges on the reflector and selecting an emergent energy and polar angle from a joint density function dependent upon the incident energy and polar angle. The weight of the history is adjusted by the functional return from the reflector, which is also based on the incident energy and angle.

The characteristics of a differential albedo emulate the attributes of the reflector material and are independent of the material or materials adjacent to the reflector. Thus, a differential albedo that is generated for a given reflector material can be used with any array, regardless of the type of fuel or fissile material contained within the array.

For many calculations involving reflected arrays of fissile material, the differential albedo treatment is a powerful tool that can significantly reduce the computing time required to determine k-effective. The savings will vary, depending on the importance of the reflector to the system. A substantial effort is required to generate a differential albedo, but the savings gained are well worth the effort for commonly used reflector materials.

To generate the differential albedo information for a material, a fixed source calculation must be made for each incident energy and angle. The data presently available for use with KENO were generated by one-dimensional discrete ordinates calculations for slab geometry, representing infinite slabs. Consequently, for a finite reflector, these data will not correctly treat histories that enter the reflector near an edge. Past experience with differential albedo reflectors indicates that k-effective appears to be conservative for small faces and will tend toward the correct result as the face becomes large relative to the area near the corners. Therefore care must be taken to ensure that any surface to which a differential albedo is applied is large enough that the errors at the edges can be ignored.

Because differential albedos are expensive and time-consuming to generate, those corresponding to the Hansen-Roach 16 energy group structure are the only differential albedos currently available for use with KENO. In the past, their use was limited to problems utilizing cross sections having the Hansen-Roach 16 energy group structure. KENO V extends the use of differential albedos to other energy group structures by allowing appropriate energy transfers. This is accomplished by creating lethargy boundary tables for the albedo group structure and the cross section group structure and determining the lethargy interval corresponding to the desired transfer (cross section group structure to albedo group structure or vice versa) based on a uniform lethargy distribution over the interval. When the energy group boundaries of the cross sections and albedos are different, the results should be scrutinized by the user to evaluate the effects of the approximations.

F11.2.5 SUPERGROUPING

An important feature of KENO V is the capability of supergrouping energy-dependent information. This includes the cross sections, albedos, pointer arrays, weights, leakages, absorptions, fissions and fluxes. If the available computer memory is too small to hold all the problem data at once, KENO V determines the number of supergroups necessary to allow execution of the problem. A problem cannot be supergrouped if the energy-dependent data associated with any individual energy group are too large to fit in the available memory. If enough memory is available to accommodate all the energy-dependent data at once, only one supergroup is created. Once the number of supergroups has been determined, the energy-dependent data are arranged in supergroups and are written on the direct access supergroup file. During execution of a problem, the supergrouped data are moved in and out of memory as necessary.

The advantage of supergrouping is that larger problems can be run on smaller computers. This capability is gained at the expense of running time and increased I/O's. The more supergroups, the more I/O's are used and the slower the problem will run because of the banking, sorting, and use of direct access devices in the solution of the problem.

In order to reduce the amount of data movement between memory and the direct access supergroup file, KENO V maintains a bank of histories (the neutron bank) and follows all those histories that fall within the current supergroup before going to the next supergroup. Histories that are scattered out of the current supergroup are placed back in the bank. When all the histories in the current supergroup have been processed, the bank is sorted, placing the histories for the most populous supergroup at the top of the bank. All other histories are placed at the bottom of the bank. The data for the most populous supergroup is then brought into memory and tracking proceeds.

F11.2.6 RESTART

KENO V incorporates a versatile and convenient restart capability. The decision to write a restart file requires the user to specify only the number of generations between writing restart data and the unit number where the restart file is to be written. A file definition card must be included in the job control language for the restart data file. The input data are the first data written on the restart data file. The group-dependent input data are written a group at a time. This includes the cross sections, albedos, pointer arrays and weights. The number of records of input data is automatically determined by the code and written on the restart data file. After the input data have been written on the restart data file, the calculated data are written at the end of each specified generation. These data include the generation number, random number, number of histories per generation, number of energy groups, bank lengths, common information, the k-effectives by generation, the neutron bank, the fission densities, matrix arrays, and the calculated group-dependent data. These group-dependent data are written a group at a time and include leakages, absorptions, fissions and fluxes.

The KENO V restart capability allows a problem to be restarted at the first generation with different input because all data input supersedes data from the restart data file.

If a problem is to be restarted at a generation greater than one, the only data that can be changed are certain parameter data. Changes in the parameter data that are not allowed include (1) requesting fissions and absorptions by region if they were not requested by the parent case, (2) requesting fission densities and fluxes if they were not requested by the parent case, (3) requesting matrix information that was not requested in the parent case, and (4) changing the configuration of the neutron bank to be different from that of the parent case.

Because restart data are written a group at a time, a problem may be restarted with an entirely different supergroup structure.

If a problem is to be restarted following a generation for which restart data were not written, the code will write a message and restart with the next available generation for which restart data exist. If no such generation is found, the problem is terminated.

F11.2.7 GEOMETRY

KENO geometry is restricted to the use of specific shapes. These shapes are called geometry regions or regions. Allowed shapes are cubes, cuboids (rectangular parallelepipeds), spheres, cylinders, hemispheres, and hemicylinders. These shapes must be oriented along orthogonal axes and cannot be rotated. They can be translated. Hemispheres and hemicylinders are not limited to half-spheres and half-cylinders; the definitive plane can be positioned by entering a chord. The value of this chord can range from the positive magnitude of the radius (giving a complete sphere or cylinder) to the negative magnitude of the radius (giving a zero volume, nonexistent sphere, or cylinder).

A major restriction applied to KENO geometry is that intersections are not allowed. Furthermore, each successive geometry region must completely enclose the preceding region. Tangency and shares faces are allowed. The volume of a region is the volume of the specified shape minus the volume of the preceding region shape. To alleviate the complete enclosure restriction, KENO allows multiple sets of geometry regions with each set independently governed by this restriction. Each set of these multiple geometry regions are called units or box types. Units can be stacked together in a three-dimensional rectangular parallelepiped called an array or lattice just as children's blocks can be stacked. Units that are to be stacked together in this manner must have a rectangular parallelepiped outer region and the adjacent faces of adjacent units must be the same size and shape. An array can be treated as a building block and be used as a unit within another array.

The use of holes in KENO V.a allows a unit to be emplaced within another unit, thereby alleviating the restriction that each region within a unit must completely enclose all preceding regions within that unit. However, a hole is not allowed to intersect other holes or regions. A unit that is to be used as a hole need not have a rectangular parallelepiped as its outer boundary.

Multiple arrays can be described in KENO V.a. The global array in an unreflected problem is the outermost array in the problem geometry description. The global array in a reflected problem is the array referenced by surrounding geometry regions following the last array placement description that does not immediately follow a unit number description. See Sect. F11.4.5.

Consistent with past versions of KENO, KENO V.a retains the capability of running a single unit problem. A single unit problem is one that has no array description.

F11.2.8 REFERENCES

1. Robert V. Meghreblian and David K. Holmes, *Reactor Analysis*, McGraw-Hill (1960).
2. J. E. Powell and C. P. Wells, *Differential Equations*, Ginn Company (1950).
3. E. A. Straker, P. N. Stevens, D. C. Irving, and V. R. Cain, *The Morse Code - A Multigroup Neutron and Gamma-Ray Monte Carlo Transport Code, Appendix B*, ORNL-4585 (1970).
4. T. J. Hoffman, *The Optimization of Russian Roulette Parameters for Keno*, ORNL/TM-7539 (1982).
5. L. M. Petrie and N. F. Cross, *KENO IV - An Improved Monte Carlo Criticality Program*, ORNL-4938 (1975). Also see Sect. F5 of the SCALE manual.
6. J. R. Knight and L. M. Petrie, *16 and 123 Group Weighting Functions for KENO*, ORNL/TM-4660 (1975).
7. G. E. Whitesides and J. T. Thomas, "The Use of Differential Current Albedos in Monte Carlo Criticality Calculations," *Trans. Am. Nucl. Soc. 12*, 889 (November 1969).

F11.3 LOGICAL PROGRAM FLOW

The general flow of the KENO V program during the solution of a problem is given in this section. A formal flow chart is not included because of the voluminous nature of the program. The logical program flow is broken up into small sections. The format of each section includes an abbreviated flow chart, a brief explanation of the purpose of that section of the program, and a brief description of each subroutine involved. The abbreviated flow charts are drawn with KENO V subroutine names contained in boxes and library routines as bare names. An arrow in the flow chart indicates that the subroutine associated with the arrow will be treated in detail later in the section.

F11.3.1 PROGRAM INITIATION

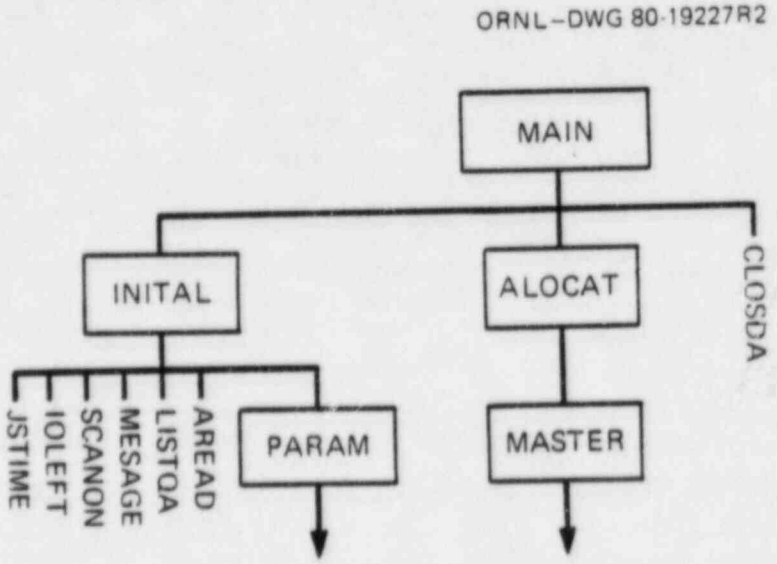


Fig. F11.3.1. Flow chart of program initiation

The function of this portion of the program is to initialize information, print a header page, call the parameter reading subroutine PARAM, access subroutine MASTER with the storage space allocated by subroutine ALOCAT, and close out the direct access files when the problem is completed or terminated.

MAIN - This subroutine sets flags to specify the proper mode of data reading for the stand-alone KENO V program. INITAL is called to perform some initialization. MASTER is then called from ALOCAT with the requested storage allocation. The direct access files are closed out by CLOSDA when a problem is completed or terminated.

INITAL - This subroutine calls library routines to perform initialization and print a header page. It then reads the problem's title card and parameter read flag and calls subroutine PARAM to do the actual reading of the parameter data.

JSTIME - This call to the library routine JSTIME is for the purpose of storing the initial time in COMMON/FINAL/ for timing purposes.

LISTQA - This library routine provides a program verification page for quality assurance purposes.

IOLEFT - This call to the library routine IOLEFT is for the purpose of storing the initial I/O count in COMMON/FINAL/ for future calculations that indicate the number of I/O's used in certain parts of the program.

SCANON - This library routine is called to activate the feature that allows scanning for the word END when reading data.

MESAGE - This library routine is called with two arguments, an eight-character hollerith argument and an output unit. Additional library routines are called from MESAGE to print a header page in block letters. The header page includes the eight-character hollerith argument (KENO−V), the date, the time execution was begun, and the job name.

AREAD - This library routine is used to read alphanumeric data. It is used here to read the title card and the parameter reading flag.

PARAM - This subroutine is called to set default values for the parameter input data and to read parameter data. See Sect. F11.3.2 for a more detailed description of PARAM.

ALOCAT - This assembly language library routine is called with two arguments. The first argument is a subroutine name and the second argument is the maximum number of words of storage to be allocated. ALOCAT calls subroutine MASTER with two arguments, an array name and the length of the array.

MASTER - This subroutine is called by subroutine ALOCAT. It is the controlling subroutine for the bulk of the KENO V program flow. See Sect. F11.3.3 for a more detailed description of subroutine MASTER.

CLOSDA - This library routine is called to close out each direct access file at the normal completion or normal termination of a problem.

F11.3.2 PARAMETER DATA

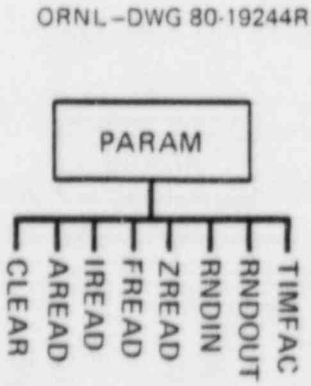


Fig. F11.3.2. Flow chart of parameter reading

The function of this section of the program *is* to set default values for the parameters and to read the parameter input data.

PARAM - This subroutine is responsible for reading the parameter input data and setting default values for the parameters. The library routine CLEAR is used to zero parameters that are defaulted to zero. The parameter data block is read using AREAD, IREAD, FREAD and ZREAD. Each entry in the parameter data block uses a keyword so the code can store the parameter data in the correct location. If the problem is a restart problem, restart information including the title of the original case, parameter data and some common information is read from the restart data file. The defaulted and input parameter tables are printed by PARAM. Some preliminary data checking is done and appropriate warning and error messages may be printed.

CLEAR - This library routine is called to zero the parameters that are defaulted to zero.

AREAD - This library routine can be called many times from subroutine PARAM. It is used to read parameter names and alphanumeric parameter data.

IREAD - This library routine can be called many times from subroutine PARAM. It is used to read integer parameter data.

FREAD - This library routine can be called many times from subroutine PARAM. It is used to read floating point data.

ZREAD - This library routine can be called from subroutine PARAM to read a hexadecimal random number that will be used as a kernel for the random number package.

RNDIN - This library routine is called from subroutine PARAM to transfer the random number read by ZREAD to the random number package. It is called only if a random number was entered as parameter data.

RNDOUT - This library routine is called from subroutine PARAM to preserve the current random number so it can be written on the restart data set and printed in the parameter data.

TIMFAC - This library routine is called from subroutine PARAM to provide the proper adjustment factor by which the allotted time is multiplied. This factor adjusts the execution time for execution on different computers.

F11.3.3 OVERALL PROGRAM FLOW

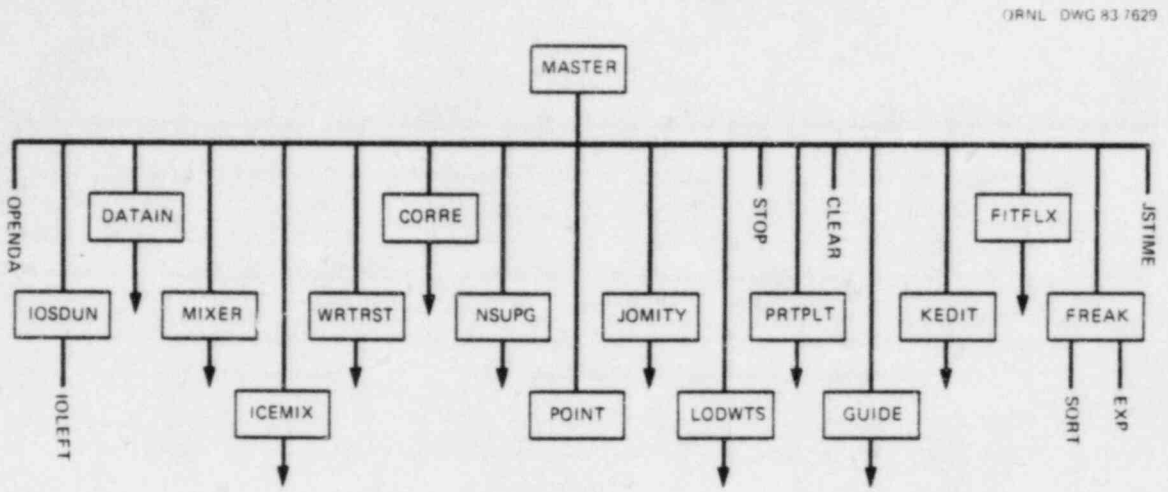


Fig. F11.3.3. Flow chart of overall program flow

The purpose of this section of the program is to direct the primary flow of KENO V. This covers the complete scope of the program from data reading to editing and printing the results.

MASTER - This subroutine controls the primary flow of KENO V. It initializes the direct access files and calls subroutines to read, check and print the input data, calls the tracking routines and prints the calculated results. The number of I/O's used during various operations is calculated and printed. The following subroutines are called from MASTER as indicated.

SUBROUTINE	FUNCTION	CONDITION
OPENDA	initialize direct access	always
IOSDUN	initialize I/O's	always
DATAIN	read input data	always
MIXER	mix cross sections	if mixing table is read
ICEMIX	read ICE mixed library	if cross sections are not to be used from the restart data file
WRTRST	write restart information	if a unit is defined for writing restart information
CORRE	generate albedo-cross-section information	if differential albedo data is used
NSUPG	create supergrouped data	always
POINT	calculate pointers	always
JOMITY	primarily controls geometry processing	always
PRTPLT	prints specified plots	if a plot data block is entered and the plot option is not turned off
CLEAR	initializes arrays where calculated data is stored	always
LODWTS	load biasing data from direct access into memory	always
GUIDE	control tracking	always
KEDIT	edit calculated results	always
FITFLX	load fluxes for printing	if fluxes are calculated
FREAK	print frequency distribution	always
JSTIME	timing	always

OPENDA - This library routine initializes the direct access files.

IOSDUN - This subroutine is called several times to indicate the number of I/O's used for various operations.

DATAIN - This subroutine controls the reading of all input data except the title card and parameters. It is explained in more detail in Sect. F11.3.4.

MIXER - This subroutine is called only if mixing table information is to be entered as data. It controls the mixing of cross-section information and writes a Monte Carlo formatted mixed cross-section library for use later in the program. More details are contained in Sect. F11.3.5.

ICEMIX - This subroutine reads the Monte Carlo formatted mixed cross-section library and manipulates the cross sections to obtain the cross-section information used by KENO V. This information is then written on the direct access data file. See Sect. F11.3.6 for additional information.

WRTRST - This subroutine is called if a unit has been defined on which to write restart information. The function of WRTRST is to write restart information as explained in Sect. F11.3.7.

CORRE - This subroutine is called only if the albedo data block specifies differential albedo data. It creates lethargy boundary tables and generates albedo cross-section energy group correspondence information. See Sect. F11.3.8 for additional details.

NSUPG - The purpose of this subroutine is to create supergrouped information and write it on the direct access file as described in Sect. F11.3.9.

POINT - This subroutine calculates pointers to access data in memory.

JOMITY - This subroutine is primarily responsible for generating additional geometry data, checking the geometry data, writing appropriate geometry error messages, and printing the geometry that is used in the problem. Section F11.3.10 contains additional details.

LODWTS - This subroutine reads biasing information data from the direct access data file, loads it into memory and prints it as described in Sect. F11.3.11.

PRTPLT - This subroutine is called to generate and print two-dimensional printer plots representing slices through the geometrical representation of the physical problem. See Sect. F11.3.12 for additional details.

CLEAR - This library routine initializes arrays where the calculated data are stored.

GUIDE - This subroutine guides the flow of the program through the actual tracking of each history. See Sect. F11.3.13 for a more detailed explanation.

KEDIT - This subroutine is responsible for editing the k-effectives and printing the various information calculated by KENO V. Section F11.3.14 contains additional details.

FITFLX - This subroutine is called only if fluxes are calculated. Its purpose is to determine the maximum number of regions for which fluxes can be contained in memory and to print the fluxes. A more detailed description is contained in Sect. F11.3.15.

FREAK - This subroutine is called to generate and print the frequency distribution of the k-effective calculated for each generation. The library routines SQRT and EXP are utilized in these calculations.

JSTIME - This library routine is called at the completion of a problem to compute the total amount of time used.

F11.3.4 PROBLEM DESCRIPTION

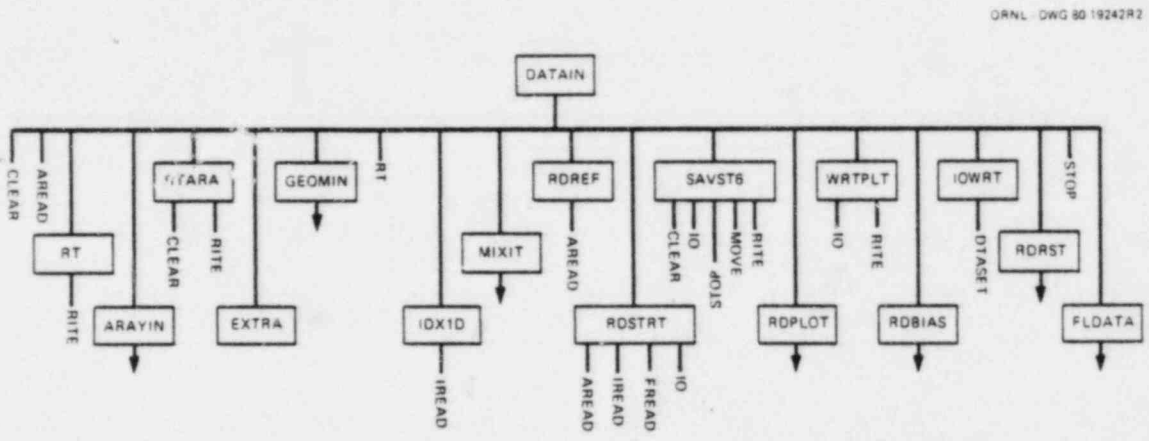


Fig. F11.3.4. Flow chart of input data reading

This section of the program controls the reading of the input data, excluding the title card and parameter data. After the data have been read, they are written on the direct access data file.

DATAIN - This subroutine controls all the data reading with the exception of the title card and parameter data. It initializes COMMON /STDATA/, the common that contains the start data, and initializes the MT array that contains the ID's of the 1-D cross sections that are to be utilized in the problem. The data reading is accomplished by reading blocks of data as described in the data guide. A keyword precedes the data, indicating the type of data to be read. After reading the keyword, the appropriate subroutine is called to read the accompanying data block. After the data block RDBOX has been read, it is written on the direct access data file. IOWRT is called several times to generate a table that lists the unit numbers used by KENO V, their names, data set names and the volume containing the data set. This table can be valuable for quality assurance applications. Subroutine FLDATA is then called to supply information that was not entered as data.

CLEAR - This library routine is called to initialize COMMON /STDATA/. If biasing data are entered, it is called to clear the space that will contain the biasing data. CLEAR is called with two arguments, a beginning location and a length. It initializes all included locations to zero.

AREAD - This library routine is used to read the READ flag, the keyword for the type of data to be read, the END flag and the keyword for the type of data just completed. It can be called many times from DATAIN.

RT - This subroutine is called to write data on the direct access data file. It passes information between DATAIN and RITE.

RITE - This library routine is called from RT to write an array of data on the direct access data file.

ARAYIN - This subroutine is called to read data defining the array size. It also reads the unit orientation data if any are entered as data. ARAYIN is not called for a single unit problem. See Sect. F11.3.4.1 for additional information. Data input for the array data block is described in detail in Sect. F11.4.5.

RTARA - This subroutine is called only if ARAYIN is called. It reads the array data from the scratch unit and writes them on the direct access device.

EXTRA - This dummy subroutine is provided to allow the user to input extra data that are not normally processed by KENO V. The user must provide the programming to read and utilize the data.

GEOMIN - This subroutine is called to read the geometry region data. See Sect. F11.3.4.2 for additional information. The geometry region data block is described in detail in Sect. F11.4.4.

IDX1D - This subroutine is called if the number of extra 1-D cross sections is greater than zero and an extra 1-D data block is entered. It reads the extra 1-D ID's and loads them into the MT array. The data reading is accomplished using the library routine IREAD. Section F11.4.9 describes the data input for defining extra 1-D data.

MIXIT - This subroutine is called to read the mixing table data block that defines the mixtures that are to be created. Section F11.1.4.3 explains the mixing procedure in more detail.

RDREF - This subroutine is called to read the boundary conditions (or albedo options) that are to be applied at the outer boundaries of the system described by the geometry data and the unit orientation data. The boundary condition data block is read using the library routine AREAD. Some preliminary data checks are made to detect invalid face code names and incompatible boundary conditions. Section F11.4.6 describes the data input for defining the boundary conditions.

RDSTRT - This subroutine is called to read the start data block that is used to define the spatial distribution of the initial generation. The library routine AREAD is used to read the keywords associated with the start data. IREAD and FREAD are used to read the integer and floating point start data, respectively. The library routine IO is used to write the start data associated with start type 6 on the scratch data file. Data input for defining the initial source distribution is described in Sect. F11.4.8.

SAVST6 - This subroutine is called to save the data associated with start type 6. SAVST6 is called only if start type 6 was specified in the start type data. See Sect. F11.4.8 for start type information. The library routine CLEAR is called to initialize the array that will contain the start data arrays. The library routine IO is used to read the start data array from the scratch data file and load it into memory. The library routine MOVE is used to move the start data array into the neutron bank. The library routine RITE is called to write the neutron bank data on the direct access data file.

RDPLOT - This subroutine is called to read the plot data block that is used to generate printer plots. Section F11.3.4.4 explains the processing of the plot data in more detail. Section F11.4.11 describes the plot input data in detail.

WRTPLT - This subroutine reads the plot data block from the scratch data file and loads it on direct access. IO reads the data from the scratch data file and loads it in memory. RITE writes the data on the direct access data file.

RDBIAS - This subroutine is called to read the biasing or weighting data to be used in the problem. See Sect. F11.3.4.4 for more information. The biasing input data block is described in detail in Sect. F11.4.7.

IOWRT - This subroutine is called with five arguments. They are, in order, (1) the output unit, (2) a four-character hollerith name representing a unit name, (3) the unit number represented by the second argument, (4) the number of words of hollerith information contained in the fifth argument, and (5) hollerith information to be printed. IOWRT is called several times to generate a table of the unit numbers, their names, the data set names, and the volumes on which each resides.

DTASET - This library routine is called from IOWRT to provide the data set name of the requested I/O unit and the volume on which it resides.

RDRST - After the data reading is complete, this subroutine is called if a unit containing data for restarting the problem has been defined. It loads data from the restart data file as described in Sect. F11.3.4.5.

FLDATA - This subroutine is called to supply default data for arrays that were not entered as input. Section F11.3.4.6 contains a more detailed account of the exact procedure.

F11.3.4.1 *Array Data*

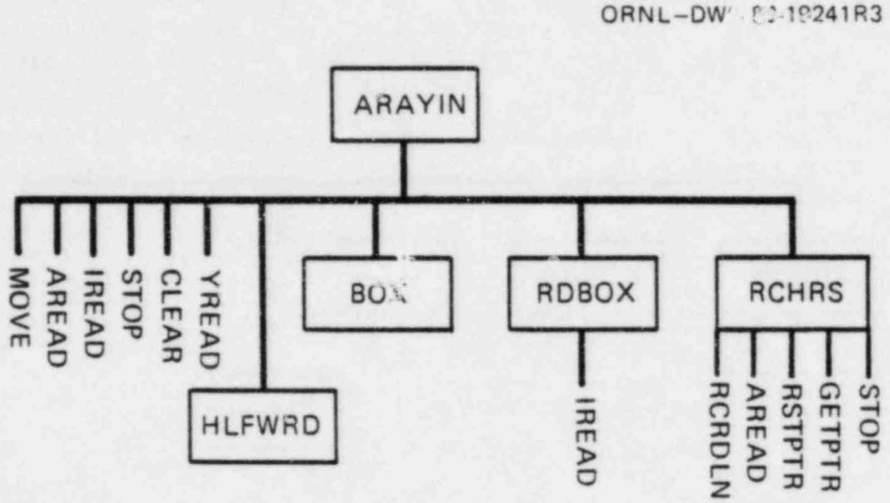


Fig. F11.3.5. Flow chart for reading array data

This section explains the procedure involved in reading the array data used in the problem.

ARAYIN - This subroutine is called from DATAIN when the words READ ARRA are encountered. It is responsible for reading the data parameters that define the size of each array. The unit orientation array data block for each array is read by HLFWRD for the FILL option and by RDBOX for the LOOP option. BOX then writes the array data on the scratch data file. The data reading is done using the library routines AREAD, IREAD, and YREAD. Section F11.4.5 describes the data read by this subroutine.

AREAD - This library routine is used to read the keywords associated with the array data.

IREAD - This library routine is used to read the integer data associated with the array data.

STOP - This library routine is called to write an error message and stop if insufficient memory is available to accommodate the unit orientation array.

YREAD - This library routine is called to read the unit orientation data for the FILL option.

HLFWRD - This subroutine is called only if the FILL option is used for entering the unit orientation data. The purpose of HLFWRD is to convert the unit orientation array data from full word integers to half-word integers.

BOX - This subroutine is called to write the array data on the scratch data file.

RDBOX - This subroutine is called only if the LOOP option is used for entering the unit orientation data. It uses the library routine IREAD to read the unit orientation data. Some data consistency checks are made and appropriate error messages are written if errors are encountered. If the input geometry is to be printed, RDBOX prints the unit orientation for each array.

RCHRS - This subroutine is used to read the comment associated with an array. GETPTR is used to return the current pointer in the input buffer. RSTPTR resets the pointer. AREAD is used to read the input data and RCRDLN sets the length of the input buffer. STOP is called to write an error message and stop if the array comment is too long (i.e., the ending delimiter is missing).

F11.3.4.2 *Geometry Data*

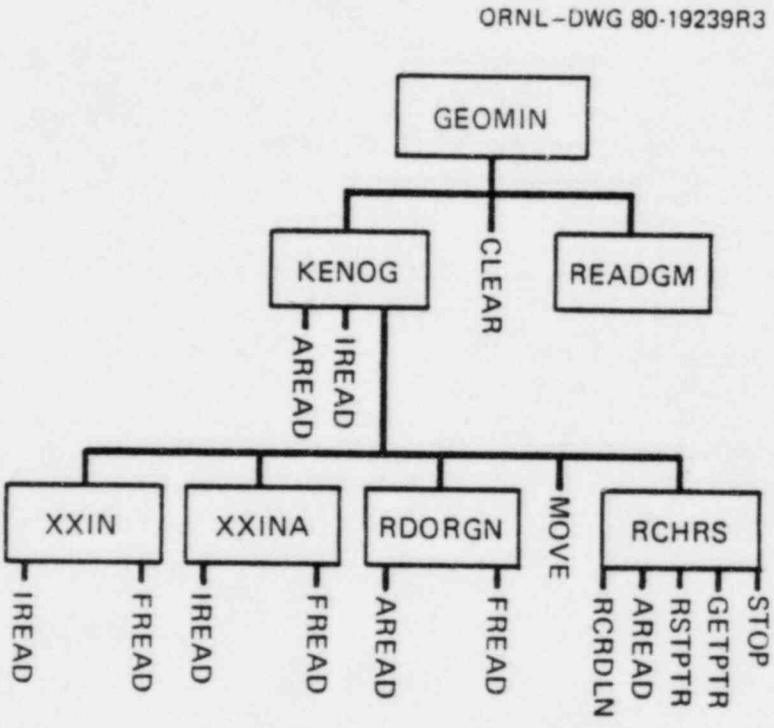


Fig. F11.3.6. Flow chart for reading geometry data

This section of the program reads the geometry data.

GEOMIN - This subroutine controls the reading of the geometry data. KENOG is called to read each geometry region specification and write it on the scratch data unit. Pointers for the data arrays are then calculated and CLEAR is called to initialize the data arrays. READGM is called to read the data from the scratch data unit and load them into the appropriate data arrays. The data block read in this portion of the program is described in Sect. F11.4.4.

KENOG - This subroutine uses AREAD to read the geometry word. IREAD is called to read the unit number for a unit or box type. It is also used to read the number of reflector regions on a reflector card. XXIN is called to read the mixture number, the bias ID number and the geometry dimensions. The origin specification, if any, associated with a sphere, hemisphere, cylinder or hemicylinder is read by RDORGN. The chord specification, if any, associated with a hemisphere or hemicylinder is also read by RDORGN. The necessary geometry data block is written on the scratch data unit. If the input geometry data block is to be printed, KENOG does this as the data are read.

XXIN - This subroutine is called to read the mixture number, the bias ID number and the geometry dimensions. IREAD is used to read the mixture number and bias ID number. FREAD is used to read the geometry region dimensions.

XXINA - XXINA is an entry point in XXIN. It is called when the geometry word ARRA or ARRAY is encountered. It reads the mixture number and the geometry dimensions. IREAD is used to read the mixture number. FREAD is used to read the geometry region dimensions.

RDORGN - This subroutine uses AREAD to read a word. If the word is ORIGIN, it uses FREAD to read the points defining the origin. If the word is CHORD, it uses FREAD to read the offset of the plane with respect to the origin. If the word is not ORIGIN or CHORD, a flag is set to prevent KENOG from attempting to read another geometry word.

RCHRS - This subroutine is used to read the comment associated with a unit in the geometry region data. GETPTR is used to return the current pointer in the input buffer. RSTPTR resets the pointer. AREAD is used to read the input data and RCRDLN sets the length of the input buffer. STOP is called to write a message and terminate the problem if the comment is too long.

READGM - This subroutine reads the geometry data from the scratch data unit and loads them into the proper arrays.

F11.3.4.3 *Mixing Table Data*

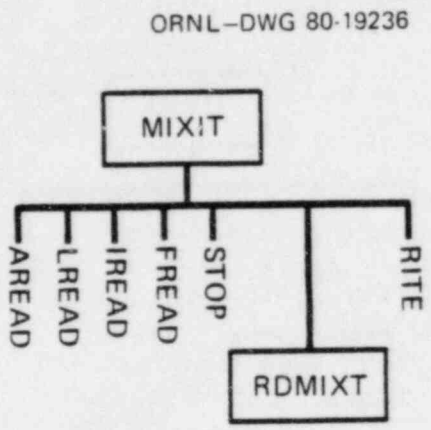


Fig. F11.3.7. Flow chart for reading mixing table data

This section deals with reading the mixing table data.

MIXIT - This subroutine uses AREAD to read the mixture keywords and the scattering keyword. IREAD is used to read the mixture numbers and the number of scattering angles as well as the nuclide ID's. FREAD reads the number densities. The necessary data arrays are written on the scratch data file. Pointers are calculated for the necessary storage arrays and RDMIXT is called to load the data from the scratch file into the storage arrays.

STOP - This library routine is called from MIXIT if the storage space is insufficient to hold the mixing table arrays.

RDMIXT - This subroutine reads the mixing table data arrays from the scratch data file and loads them into the storage arrays.

F11.3.4.4 *Plot Data*

ORNL-DWG 83-7630R

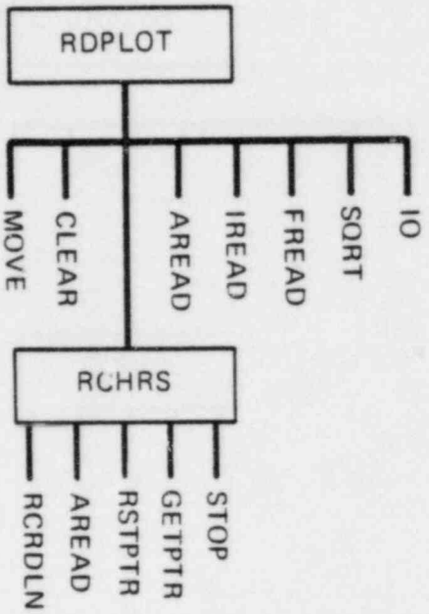


Fig. F11.3.8. Flow chart for reading plot data

This section of the program reads the plot or picture data used to generate printer plot maps of the mixtures, units and/or bias ID's used in the problem. The plot input data block is described in Sect. F11.4.11.

RDPLOT - This subroutine uses MOVE to initialize the plot title to the problem title and the table of characters to the default values. CLEAR is called to initialize the data arrays. MOVE is also used to save data to be used in multiple plots. RCHRS is used to read the plot title and the character string of symbols to be used in the plot. AREAD, IREAD and FREAD are used to read the plot or picture input data. SQRT is used to determine the normalization factor for the direction cosines and IO is called to load the plot data on the scratch data file.

RCHRS - This subroutine is used to read the plot title and the character string that defines the symbols to be used in the plot. GETPTR is used to return the current pointer in the input buffer. RSTPTR resets the pointer. AREAD is used to read the input data and RCRDLN sets the length of the input buffer. STOP is called to write an error message if the plot title is too long and terminate the problem.

F11.3.4.5 *Biasing Data*

ORNL–DWG 80-19240R3

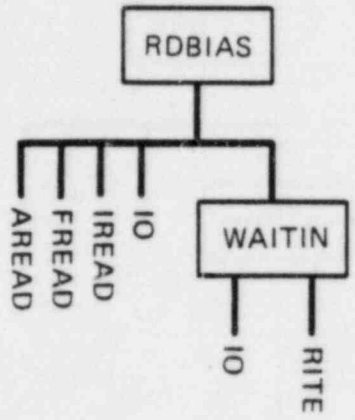


Fig. F11.3.9. Flow chart for reading biasing data

This section of the program reads the biasing data used in the problem. The biasing input data block is described in Sect. F11.4.7.

RDBIAS - This subroutine is responsible for reading the biasing data block and writing it on the scratch data file. AREAD is used to read the keywords used in the biasing data and a title for the biasing material if the energy and space dependent values of the biasing function are entered from cards. IREAD and FREAD are used to read the numerical data. Pointers for the storage arrays needed to process the biasing data are determined and WAITIN is called to load the data from the scratch data file into the storage arrays and write them on the direct access data file.

WAITIN - This subroutine reads the biasing data block from the scratch data file and loads it into the storage arrays. IO is used to load the energy and space dependent biasing function (*wtavg*) into the storage arrays. RITE is used to write the biasing data on the direct access data file.

F11.3.4.6 *Restart Data*

ORNL-DWG 80-19237R

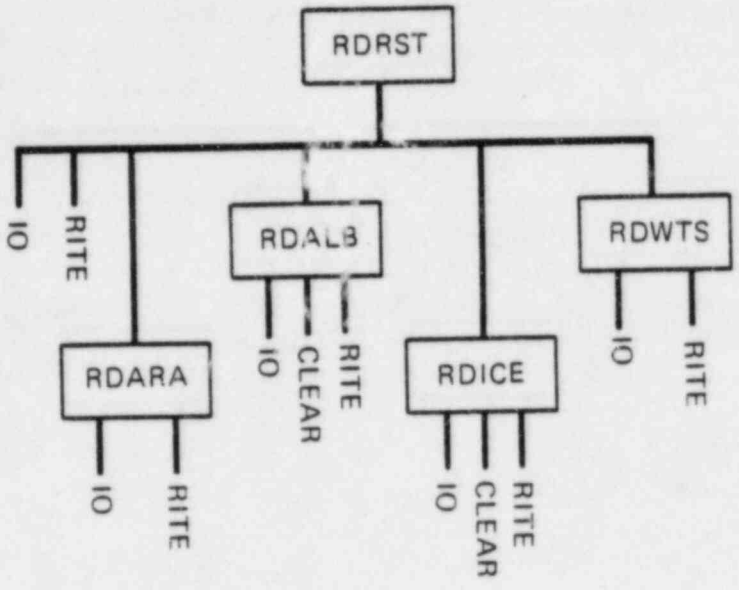


Fig. F11.3.10. Flow chart for reading restart data

This section of the program reads restart information from the restart data file.

RDRST - This subroutine is called only if the problem is to use data from the restart data file. The program recognizes that restart data will be read if the restart unit is defined as a number greater than zero. IO is used to load the array that contains the 1-D ID's from the restart data file. Each type of data is loaded from the restart data file using IO and is written on the direct access data file by RITE. All restart data except the mixed cross-section data, the differential albedo data, the array data, and the biasing data are processed directly in RDRST. RDICE is called to load the cross-section data on the direct access data file, RDALB is called to load the differential albedo data on the direct access data file, RDARA is called to load the array data on the direct access data file, and RDWTS is called to load the biasing data on the direct access data file.

RDALB - This subroutine is called from RDRST to read the albedo data block from the restart data file and write it on the direct access data file. IO is used to load the albedo pointer and length arrays into memory from the restart data file. CLEAR is called to zero the albedo pointer and length arrays and RITE writes them on the direct access data file. Each record of albedo data is read from the restart data file, loaded into memory using IO, and written on the direct access data file using RITE. When all the records of albedo data have been processed, the updated pointer and length arrays are rewritten over the initial ones using RITE.

RDICE - This subroutine is called from RDRST to read the cross-section data block from the restart data file and write it on the direct access data file. CLEAR is called to zero the pointer and length arrays. The length array is then read from the restart data file. RITE is used to write the pointer array and the length array on the direct access data file. Then IO and RITE are used to load the cross-section data from the restart data file and write them on the direct access data file. This procedure is repeated for every record of each mixture. The updated pointer and length arrays are then rewritten over the initial ones using RITE.

RDARA - This subroutine is called from RDRST to read the array data block from the restart data file and write it on the direct access data file. IO is used to load the data from the restart data file and RITE is used to write them on the direct access data file.

RDWTS - This subroutine is called from RDRST to read the biasing data block from the restart data file and write it on the direct access data file. IO is used to load the data from the restart data file and RITE is used to write them on the direct access data file.

F11.3.4.7 *Generate Remaining Data*

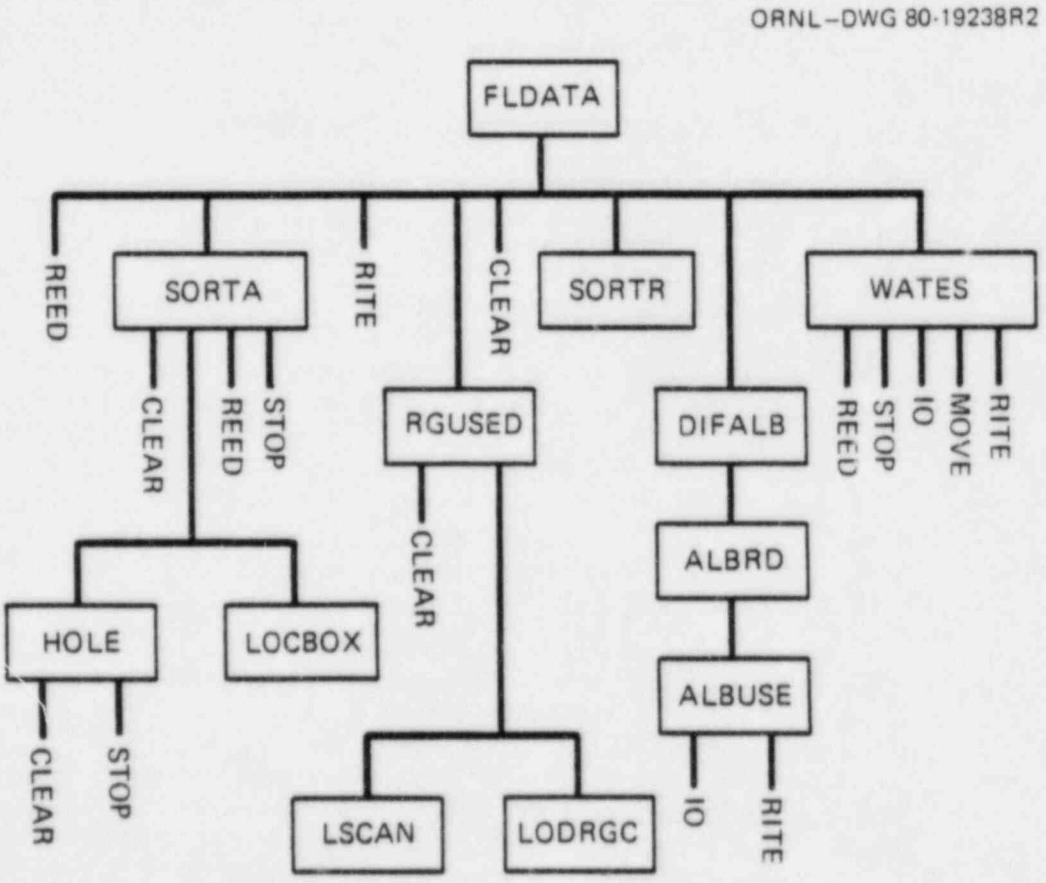


Fig. F11.3.11. Flow chart for generating remaining data

This section of the program is responsible for generating data blocks that are required to solve a problem but are not entered directly as data.

FLDATA - The library routine REED is used to load data arrays from the beginning of the mixtures used in the geometry through the geometry data. SORTA is called to determine which arrays and holes are used as well as the array and hole nesting levels. RITE is called to write the geometry data on the direct access data file. RGUSED is called to determine which geometry regions are used in the problem. SORTR is called to generate the mixture correspondence array. It is called again to generate the bias region correspondence array. These correspondence arrays are used to avoid storing mixture cross sections and biasing data that were entered as data but are not actually used in the problem. If boundary conditions specify differential albedo data and they are not available from the restart data file, DIFALB is called to read the albedo data block and load the requested data on the direct access file. If the requested biasing data were unavailable from the restart data file, WATES is called to load the energy and position biasing function (*wtavg* array) on the direct access data file.

SORTA - This subroutine checks to see that the global array is properly defined. It determines the array correspondence array and the nesting levels for holes and arrays. SORTA uses CLEAR to initialize arrays. REED is used to load the array data and STOP is used to write an error message and terminate if more computer storage is needed for the problem.

HOLE - This subroutine is called from SORTA to determine what holes occur at the next nesting level and to adjust the array nesting level for arrays that occur in holes. It also checks to be sure holes are not recursively nested.

LOCBOX - This function subprogram is called from SORTA to determine the unit or box type at a given position in the unit orientation array.

RGUSED - This subroutine determines which geometry regions are used in the problem. The library routine CLEAR is called to zero the space for the region correspondence array. LSCAN determines if a particular unit or box type has been used in the unit orientation array and if it has, LODRGC is called to load the region number into the region correspondence array.

LSCAN - This is a logical function that returns a value of true if the specified unit or box type is used in the unit orientation array. A value of false is returned if the unit or box type was not used in the unit orientation array.

LODRGC - This subroutine loads the region number in the region correspondence array.

SORTR - This subroutine is called twice from FLDATA to create a mixture correspondence array and a biasing correspondence array. These correspondence arrays are used to avoid storing mixture cross sections and biasing information that were defined in the input data but were not referenced in the geometry data utilized in the problem. They are also used throughout the code for accessing the proper mixture cross sections and biasing information.

DIFALB - This subroutine is called if differential albedos are specified as a boundary condition but are not available from the restart data file. It rewinds the albedo data file, reads the header record and calculates pointers. ALBRD is called to load the albedo data.

ALBRD - This subroutine searches through the albedo data file to locate the requested albedo name or boundary condition. If it is not found, an error message is written. If it is found, the number of different differential albedos that were requested are tabulated and ALBUSE is called.

ALBUSE - This subroutine writes the pointer array on the direct access data file. IO is used to load data from the albedo data file and RITE is used to write the data on the direct access data file. Then the pointer and length arrays are rewritten on the direct access data file.

WATES - This subroutine reads the biasing input data block from the direct access data file and reads the KENO V weights library. STOP is called if the computer storage space is too small to contain the energy and position dependent biasing function (*wtavg*). IO is used to load the biasing function into a temporary storage array. If a specific biasing function is to be used, MOVE is called to load it into the *wtavg* array. If biasing or weighting data are entered from cards, REED is used to load the data into a temporary storage array. If a specific biasing function that was loaded from cards is to be used in the problem, MOVE is called to load it into the *wtavg* array. When all the data have been processed, RITE is called to load the biasing input data on the direct access data file. RITE is called again to load the *wtavg* array on the direct access data file.

F11.3.5 CREATE A MIXED CROSS SECTION DATA FILE

ORNL-DWG 80-19233R

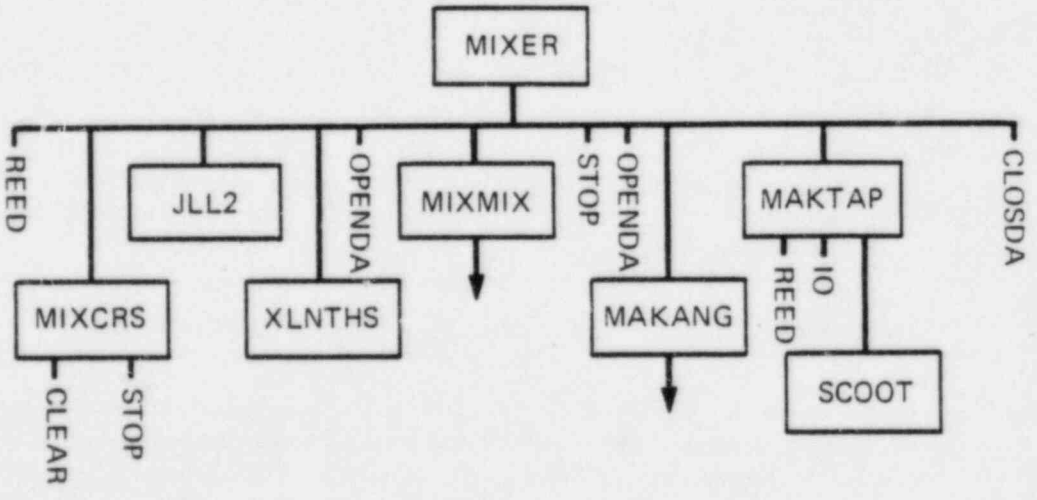


Fig. F11.3.12. Flow chart for mixing cross sections

The function of this portion of the program is to utilize the mixing table data and AMPX working format library from the cross-section data file to create a Monte Carlo formatted mixed cross-section data file. This data file has the same format as an ICE mixed cross-section MORSE/KENO library.[1]

MIXER - This subroutine controls the mixing of cross sections and the generation of the angles and probabilities to create a Monte Carlo formatted mixed cross-section library. First the mixing table is read from the direct access data file, MIXCRS is called to generate the mixture correspondence array, and JLL2 is called to generate the array that points to the beginning of each group in the triangularized mixture array. XLNTHS is called to calculate the number of direct access blocks required for the mixtures and MIXMIX is called to do the mixing. MAKANG is called to generate the angles and probabilities from the mixed cross sections, and MAKTAP is called to write the Monte Carlo formatted mixture tape.

MIXCRS - The mixture correspondence array is generated by this subroutine. This array relates the mixture number in the mixing table to the mixture index.

JLL2 - This routine generates an array of pointers that point to the beginning location of each energy group in a triangularized mixture transfer array.

XLNTHS - This routine computes the number of direct access blocks necessary to hold the mixtures.

MIXMIX - This subroutine controls the actual mixing of the cross sections. See Sect. F11.3.12 for a more detailed description of subroutine MIXMIX.

MAKANG - This subroutine controls the generation of angles and probabilities. See Sect. F11.3.13 for a more detailed description of subroutine MAKANG.

MAKTAP - Subroutine MAKTAP is used to write a mixed cross-section library in a format similar to the ICE-II[1] Monte Carlo library format. MAKTAP calls SCOOT to compress out zero 1-D cross sections.

SCOOT - This subroutine eliminates 1-D cross sections that are zero in all groups.

F11.3.5.1 *Cross-Section Mixing*

ORNL–DWG 80-19234

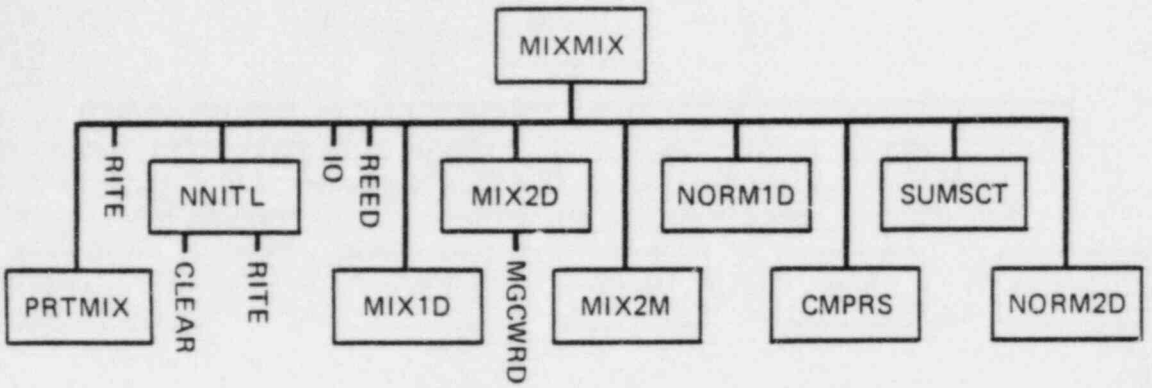


Fig. F11.3.13. Flow chart of mixing operations

MIXMIX - This subroutine controls the actual mixing of the cross sections. It calls PRTMIX to print the mixing table and calls NNITL to initialize the mixtures. It then reads the input AMPX working library, calls MIX1D to mix the 1-D cross sections, and calls MIX2D to mix the 2-D arrays. After all the nuclides selected from the working library have been mixed, the already mixed cross sections may be used as input for further mixing operations using MIX1D to mix the 1-D cross sections and MIX2M to mix the 2-D cross sections. NORM1D is then called to normalize the fission spectrum and to prepare the adjoint production cross sections if necessary. CMPRS is called to generate the magic word array and to compress the 2-D arrays. SUMSCT is called to sum the transfer array by group and NORM2D is called to convert the transfer array to a probability density function.

PRTMIX - This subroutine prints the requested number of scattering angles and the mixing table.

NNITL - This subroutine initializes the mixture cross sections to zero on the direct access storage.

MIX1D - This subroutine mixes the 1-D cross sections. It mixes most 1-D cross sections using number densities, but the fission spectrum is mixed based on $\nu\Sigma_f\phi dE$ for the nuclide being mixed. The flux is the flux used to generate the multigroup cross sections, if it is available. Otherwise a flat flux is used. If the problem is an adjoint case, MIX1D prepares the cross sections in adjoint form as they are mixed.

MIX2D - This subroutine mixes the 2-D cross sections from a working library. The mixture cross sections are stored in a triangularized array. If the problem is an adjoint case, MIX2D prepares the cross sections in adjoint form as they are mixed.

MIX2M - This subroutine mixes previously mixed 2-D cross sections into new mixtures using new number densities or volume fractions.

NORM1D - This subroutine is used to normalize the fission spectrum vector. If the problem is an adjoint problem, it also interchanges the fission spectrum and $\nu\Sigma_f$.

CMPRS - This subroutine compresses out zeros in the 2-D mixture arrays and generates the pointer array used to access data in these arrays.

SUMSCT - This subroutine is used to sum the transfers for each group.

NORM2D - This subroutine normalizes the transfer arrays and divides the P_ℓ arrays by $(2_\ell + 1)$.

F11.3.5.2 *Generate Angles and Probabilities*

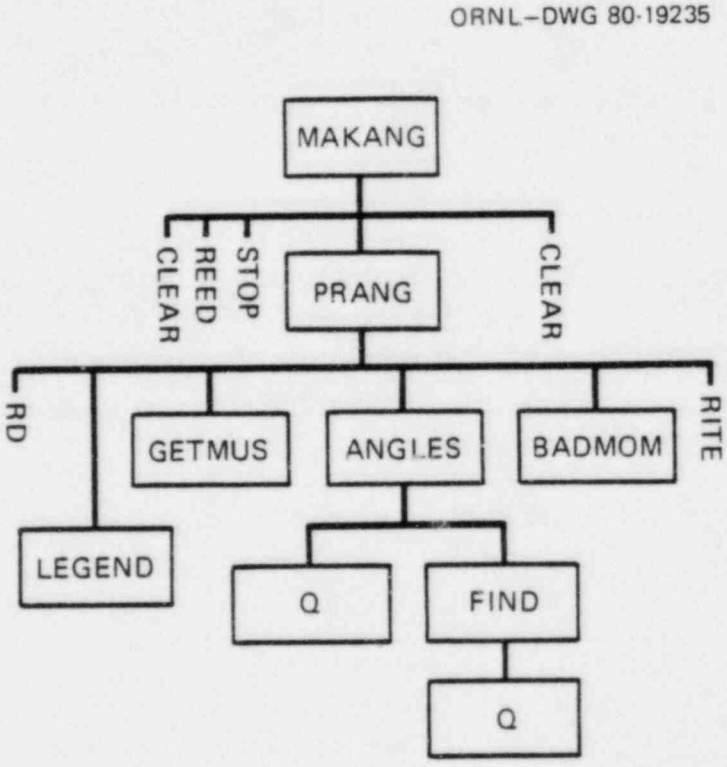


Fig. F11.3.14. Flow chart of angle and probability generation

MAKANG - This subroutine controls the generation of angles and probabilities. If the available storage is insufficient to generate all the angles and probabilities for one mixture at a time, MAKANG supergroups the data. Subroutine PRANG is called to actually generate the angles and probabilities for a supergroup.

PRANG - This subroutine reads the mixture data for a supergroup and loops through each transfer in the supergroup. It calls LEGEND to convert the Legendre expansion to moments. GETMUS is called to generate the orthogonal polynomials and ANGLES is called to generate the angles and probabilities. If there is an error, BADMOM is called to print the data input and the data corresponding to the angles and probabilities actually used.

LEGEND - This subroutine determines the moments of a function from a Legendre expansion of the function.

GETMUS - This subroutine calculates the μ_i's and σ_i's that determine the orthogonal polynomials, the Q_i's.[2]

ANGLES - This subroutine determines the angles and probabilities from the Q_i polynomials.[2] It calls FIND to determine the roots of Q_i, and calls the function Q to evaluate Q_i.

FIND - This subroutine finds the roots of a polynomial Q_i by using an interval-halving technique.

Q - This function evaluates a polynomial Q_i by using the recursion formulas.
$Q_{i+1}(x)=(x-\mu_{i+1})Q_i(x)-\sigma_i^2 Q_{i-1}(x)$,
$Q_0(x)=1.0$, and
$Q_1(x)=x-\mu_1$.

BADMOM - This routine is called when an error is detected in generating the angles and probabilities. BADMOM prints the Legendre coefficients that were input to the calculation and the moments corresponding to these coefficients. It then computes the moments corresponding to the angles and probabilities actually generated and the Legendre coefficients corresponding to these moments. BADMOM then prints these moments and coefficients so they may be compared with the original moments and coefficients.

F11.3.6 WRITE CROSS SECTIONS ON DIRECT ACCESS FILE

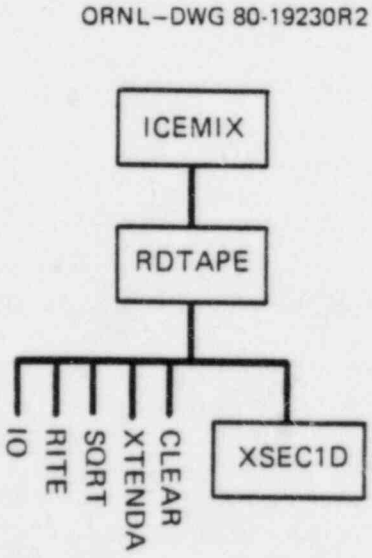


Fig. F11.3.15. Flow chart of mixed cross-section processing

This section of the program is responsible for reading the Monte Carlo mixed cross-section library, selecting the desired information, and writing it on the direct access data file.

ICEMIX - This subroutine reads information from the Monte Carlo mixed cross-section library, performs preliminary checks for data consistency, calculates pointers for the desired data arrays and calls RDTAPE to actually read the library and write the desired information on the direct access data file.

RDTAPE - This subroutine reads the Monte Carlo formatted mixed cross-section library, sorts out the data needed by KENO V, and writes the data on the direct access data file. The library routine SQRT is used in calculating the inverse velocities used by KENO V. The library routine IO is used repeatedly to read and load data from the Monte Carlo formatted mixed cross-section library. RITE is used to write data required by KENO V on the direct access data file. XTENDA is a library routine that is called to extend the number of blocks for a direct access device if too few blocks were initialized. XSEC1D is called to process the 1-D cross sections. RDTAPE manipulates the pointer array and writes it on the direct access data file. The P_0 cross sections, angles and probabilities are also processed by RDTAPE.

XSEC1D - This subroutine sorts the 1-D cross sections and loads them in the following order: total cross section, scattering cross section, production cross section, absorption cross section, extra cross sections for special purposes, and the fission spectrum. The extra cross sections require the user to provide programming to utilize them. If one of the required cross sections is not found, it is padded with zeros. The cross sections are normalized by the total cross section and the fission spectrum is summed and normalized to 1.0. The data are transferred back to RDTAPE where they are written on the direct access data file.

F11.3.7 WRITE INPUT DATA ON RESTART FILE

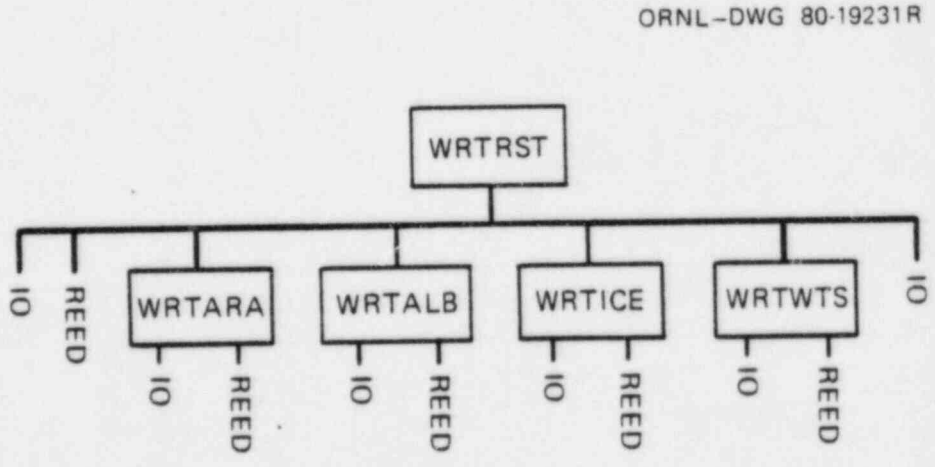


Fig. F11.3.16. Flow chart for writing data on the restart data file

This section of the program is responsible for writing all data except the calculated results on the restart data file. This section of the program is omitted if a unit number has not been assigned for the restart data file. This information is entered as parameter data, WRS=, as described in Sect. F11.5.3. The calculated results are written on the restart data file later in the program. The restart data file is used for restarting a problem.

WRTRST - This subroutine writes the input data on the restart data file. The array that contains the ID's of the 1-D cross sections is written first. The geometry data, mixing table data, start data, and energy and inverse velocities are also written. WRTARA is called to write the array data and unit orientation data, WRTALB is called to write the albedo data, WRTICE is called to write the cross-section data, and WRTWTS is called to write the biasing data.

WRTARA - This subroutine is called from WRTRST to write the array number, array size and corresponding unit orientation array on the restart data file for each array that is entered in the problem. The library routine REED is used to read the data from the direct access data file and IO is used to write it on the restart data file.

WRTALB - This subroutine is called from WRTRST to write the albedo data on the restart data file. The library routine REED is used to read the albedo data from the direct access data file and IO is used to write it on the restart data file.

WRTICE - This subroutine is called from WRTRST to write the cross-section data block on the restart data file. The library routine REED is used to read the cross-section information from the direct access data file and IO is used to write it on the restart data file.

WRTWTS - This subroutine is called from WRTRST to write the biasing input data block on the restart data file. The library routine REED is used to read the data block from the direct access data file and IO is used to write it on the restart data file.

F11.3.8 GENERATE ALBEDO CROSS SECTION CORRESPONDENCE TABLES

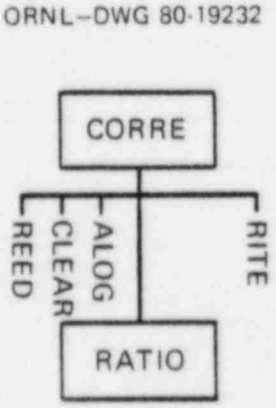


Fig. F11.3.17. Flow chart for generating correspondence arrays

This section of the program generates the correspondence tables that are necessary to correlate the energy group structures of the cross sections and albedos. It is invoked only if differential albedos are used in the problem.

CORRE - This subroutine is called only if differential albedos are used. The library routine CLEAR is used to initialize the arrays CORRE reads the energy bounds for the albedos and the cross sections and converts them to lethargies by using the library routine ALOG. RATIO is called to create a probability array relating the cross-section energy bounds to those of the albedos. A second call is made to RATIO to create and print the probability array relating the albedo energy bounds to those of the cross sections. RITE is used to write the correspondence tables on the direct access data file.

RATIO - This subroutine generates the pointer array that is used to access the albedo data. It also generates probability arrays that are used to correlate the cross-section energy group structure with the albedo energy group structure.

F11.3.9 GENERATE SUPERGROUPED DATA

ORNL-DWG 80-19243R

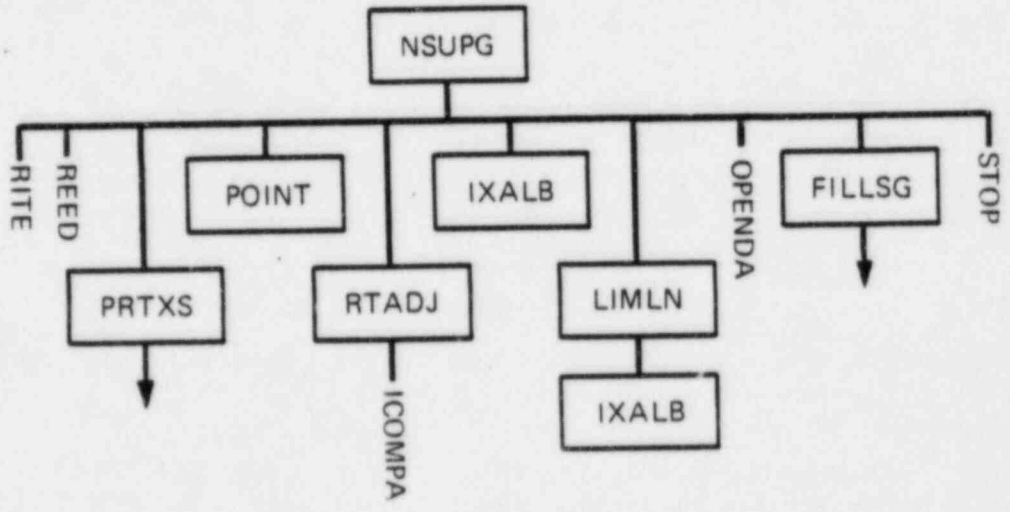


Fig. F11.3.18. Flow chart of super group creation

The function of this section of the program is to create supergrouped data from the energy-dependent input data. It determines which energy group has the largest amount of data associated with it, and determines how many supergroups must be created to be able to fit the data into the available computer memory. The energy groups associated with each supergroup are tabulated and supergrouped data are written on the direct access supergroup data file, one supergroup at a time.

NSUPG - This subroutine controls the creation of the supergrouped data and writes them on the direct access supergroup data file. The library routines RD and REED are used to load the cross-section data and albedo data from the direct access data file. PRTXS is called to print the cross-section information as described in Sect. F11.3.9.1. POINT is called to determine the storage requirements of the nonsupergrouped data and create pointers to access these data. The first portion of the additional information table is printed in the computer output. RTADJ is then called to right adjust the albedo boundary condition names. These albedo boundary condition names are then printed in the additional information table, thus completing the table. If insufficient space is available, STOP is called to write a message and terminate the problem. A rough estimate of the number of supergroups is made and an implied loop iterative procedure is used to determine the number of supergroups that must be created to fit the problem in the available space. IXALB is called to determine the amount of space required for the albedos corresponding to a cross-section energy group. A check is made to be sure the largest amount of data for a single energy group will fit in the available memory. If they will, supergrouping is possible. Otherwise a message is written and a stop is executed. Once the number of supergroups is determined, LIMLN is called to calculate and print information in the space and supergroup information table. The printed information includes the supergroup and corresponding energy groups, the length of the cross section and albedo data for the supergroups and the total length of each supergroup. FILLSG is then called to construct the supergroups as described in Sect. F11.3.9.2.

PRTXS - This subroutine prints the cross-section information as described in Sect. F11.3.9.1.

POINT - This subroutine determines the storage requirements of the nonsupergrouped data and creates pointers to access data within the nonsupergrouped storage array.

RTADJ - This subroutine right adjusts the albedo boundary condition names which are read in as left-adjusted data. RTADJ utilizes the library routine ICOMPA to determine if a character is blank.

IXALB - This function determines the amount of space necessary to contain the albedo data corresponding to a cross-section energy group.

LIMLN - This subroutine is called to calculate and print the supergroup number, the energy groups contained in the supergroup, the length of the cross section and albedo data in the supergroup, and the total length of the supergroup. IXALB is utilized to obtain the amount of space required to contain the albedo data corresponding to a cross-section energy group.

FILLSG - This subroutine constructs supergroups as described in Sect. F11.3.9.2.

F11.3.9.1 *Print Macroscopic Cross Sections*

ORNL–DWG 80-19228

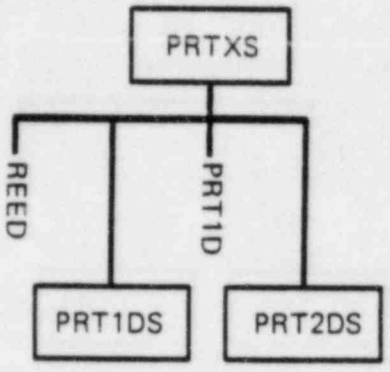


Fig. F11.3.19. Flow chart for printing macroscopic cross sections

This portion of the program prints the macroscopic cross-section data for materials used in the problem description.

PRTXS - If macroscopic cross sections are not to be printed, a return is executed. PRTXS loops over the number of mixtures that are used in the problem. REED is used to load the data, and pointers into the data arrays are calculated. If the space is insufficient to contain the data, a message is printed. If 1-D mixture cross sections are to be printed, PRT1DS is called to print them. If the extra 1-D cross sections are to be printed, REED is used to load the data and PRT2DS is called to print them. If the number of scattering angles is greater than zero and the mixture probabilities and angles are to be printed, REED is used to load probability data and PRT2DS prints them. Then REED loads angle data and PRT2DS prints them.

PRT1DS - This subroutine prints the macroscopic 1-D cross sections, one energy group at a time.

PRT2DS - This subroutine prints a two-dimensional variable length array in a compact manner.

F11.3.9.2 *Write Supergroup Data File*

ORNL–DWG 80-19229

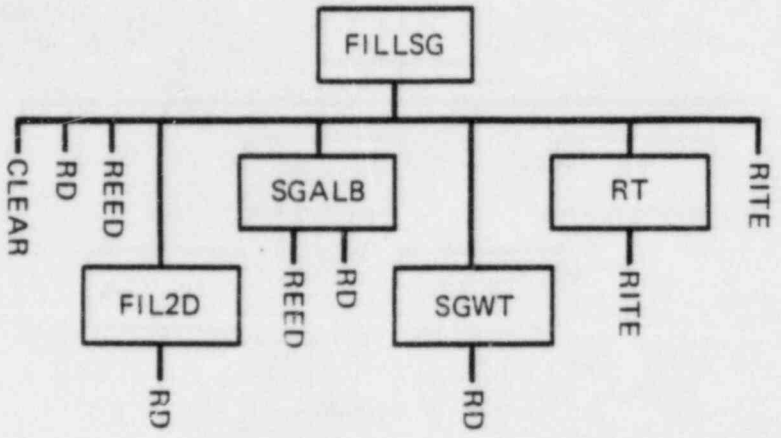


Fig. F11.3.20. Flow chart for loading supergrouped data

This portion of the program collects the group-dependent data by supergroup and writes them on the direct access supergroup data file.

FILLSG - This subroutine is responsible for filling the supergroups and writing them on the direct access supergroup data file. For each supergroup, CLEAR is called to initialize the working space. Pointers are calculated for the data arrays and space is allotted for the calculated data such as fluxes that will be supergrouped. The library routines REED and RD are used to load data from the direct access data file into the proper arrays as follows: For each mixture the inverse velocities are loaded, followed by the pointer array and all the 1-D cross-section arrays. RD and REED are used to load the 2-D arrays from the direct access data file. FIL2D is called to load them into the supergroup. Pointers are calculated for the albedo data and SGALB is called to load the albedo data into the supergroup. SGWT loads the average weight array (biasing data) into the supergroup. FILLSG then calls RT to write the calculated data for the supergroup on the direct access supergroup data file. RITE writes the group-dependent data on the direct access supergroup data file.

FIL2D - This subroutine loops over the number of mixtures. It uses RD to load the 2-D cross-section data for the supergroup from the direct access data file; first the P_0 data, then for each scattering angle, the angles and probabilities. The pointer array is redefined so the supergrouped data can be accessed.

SGALB - This subroutine uses REED and RD to load the albedo data for the supergroup from the direct access data file. It loops over the number of differential albedos used in the problem. Then the pointer array is redefined so the supergrouped data can be accessed.

SGWT - This subroutine loops over the number of biasing regions used in the problem. RD is used to load the average weight array from the direct access data file. If any average weight entry remains undefined or zero, it is set to the default value of weight average.

RT - This subroutine is called to write the calculated data (fluxes, etc.) on the direct access supergroup data file.

RITE - This library routine is used to write the group-dependent data for the supergroup on the direct access supergroup data file.

F11.3.10 PROCESS GEOMETRY

ORNL–DWG 83-7631R

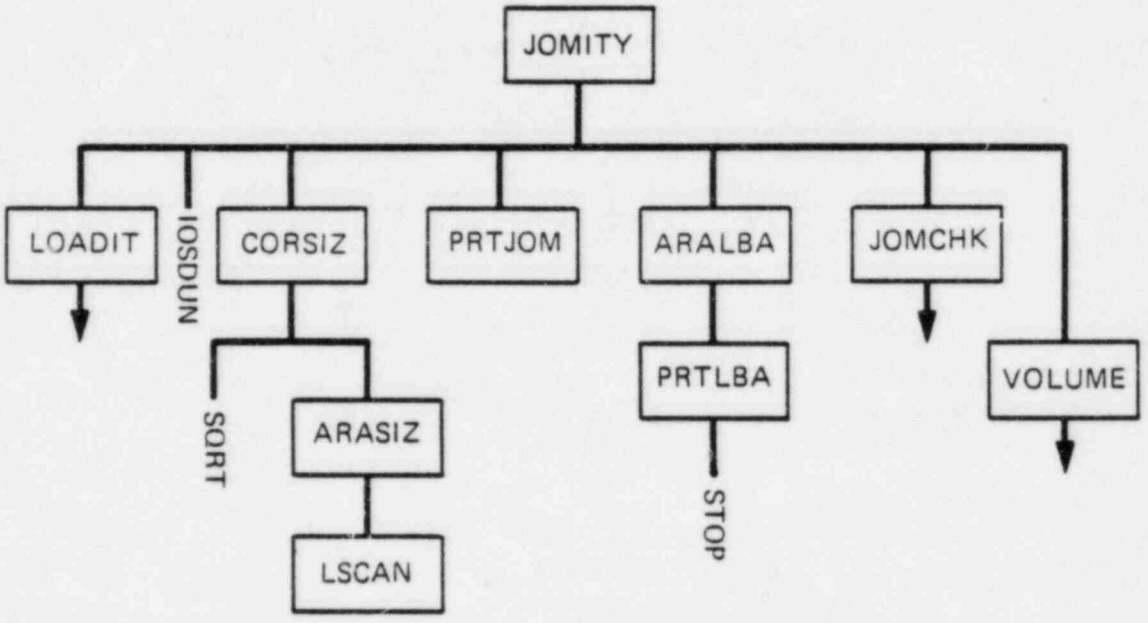


Fig. F11.3.21. Flow chart for processing geometry

This portion of the program is primarily responsible for loading the geometry data, generating additional geometry data, checking the geometry for consistency, writing error messages related to the geometry, and printing the geometry that is used in the problem.

JOMITY - This subroutine is responsible for generating additional geometry data, checking the geometry data, writing geometry error messages, and printing the geometry.

LOADIT - This subroutine loads the geometry data and the non-supergrouped portion of the albedo data. Section F11.3.10.1 contains a more detailed description of the procedure.

CORSIZ - This subroutine sends the appropriate lattice or array information to ARASIZ for each lattice that is used in the problem. Using this information, it calculates the overall positive dimensions of the global array. The library routine, SQRT, is utilized to calculate the maximum chord length of an unreflected array, a reflected array or a single unit problem.

ARASIZ - This subroutine uses the array unit orientation data to calculate the positive dimensions of the core boundary for that array or lattice. The function LSCAN is called to determine if a specified unit or box type has been used in the array. ARASIZE also checks to assure that the faces of adjacent units are the same size and shape. Several error messages are written if errors are encountered.

LSCAN - This is a logical function that returns a value of true if the specified unit or box type is used in the unit orientation array. A value of false is returned if the unit or box type was not used in the unit orientation array.

PRTJOM - This subroutine prints the geometry data that are used in the problem.

ARALBA - This subroutine cails PRTLBA for each array that is used in the problem.

PRTLBA - This subroutine is called to print the unit orientation data for each lattice or array that is used in the problem. It prints the error and/or warning messages associated with each lattice. STOP is called if a unit or box type number exceeds 10,000.

JOMCHK - The purpose of this subroutine is to perform consistency checks on the geometry data and write the appropriate error messages. See Sect. F11.3.10.2 for additional details.

VOLUME - This subroutine is responsible for calculating the volume of each geometry region and the cumulative volumes for each unit that is used in the problem. See Sect. F11.3.10.3 for additional details.

F11.3.10.1 *Load Data From the Direct Access File*

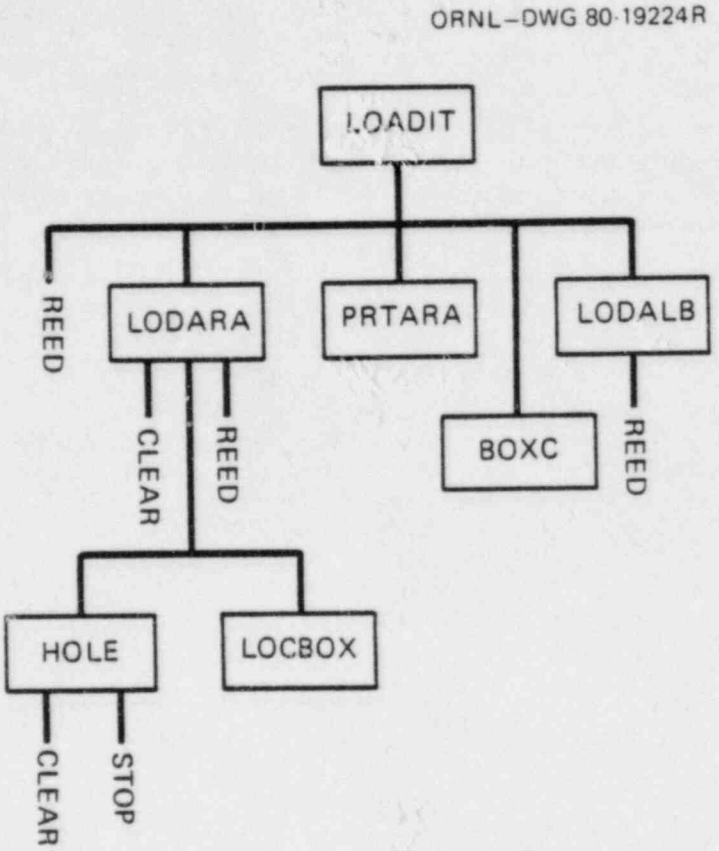


Fig. F11.3.22. Flow chart for loading data from direct access

This portion of the program loads data from the direct access data file into permanent memory.

LOADIT - This subroutine calls the library routine REED to load the geometry data. If the problem is an array problem (lattice geometry), LODARA is called to load the lattices that are used in the problem and recompute and readjust the array nesting level array and hole nesting level array. If multiple boxes are used in the problem, PRTARA is called to print the unit orientation array for each lattice used in the problem. BOXC is called to load the box correspondence array and LODALB is called to load the non-supergrouped portion of the albedo data.

LODARA - This subroutine is responsible for loading the lattices (unit orientation arrays) that are used in the problem, computing the hole nesting level array, and computing and adjusting the array nesting level array. CLEAR is used to initialize the arrays and REED is used to load the unit orientation arrays. HOLE and LOCBOX are both called by LODARA.

HOLE - This subroutine is called from LODARA to determine which holes occur at the next nesting level and to adjust the array nesting level for arrays that occur in holes. It also checks to assure that holes are not recursively nested. CLEAR is used for initialization purposes and STOP is called if holes are recursively nested.

LOCBOX - This function is called from LODARA to return the unit or box type at a given position in the unit orientation array.

PRTARA - This subroutine prints the unit orientation array for each lattice that is used in the problem.

BOXC - This subroutine uses the number of units or box types and the geometry region number corresponding to the first and last geometry region of each unit to generate the box correspondence array which contains the unit or box type number for each geometry region. This is loaded in the appropriate position as it is generated.

LODALB - This subroutine calls the library routine REED to load the pointer and length arrays for the albedo data from the direct access data file. REED is used to load the nonsupergrouped albedo data, for each albedo that is used, into a temporary array. A loop over the number of angles is then used to load these data into the appropriate arrays. When all of the albedos used in the problem have been processed, REED is called to load the pointer arrays for the cross sections and albedos as well as the arrays defining the albedo to cross-section energy group correlation.

F11.3.10.2 *Check the Geometry Data*

ORNL–DWG 80-19226R2

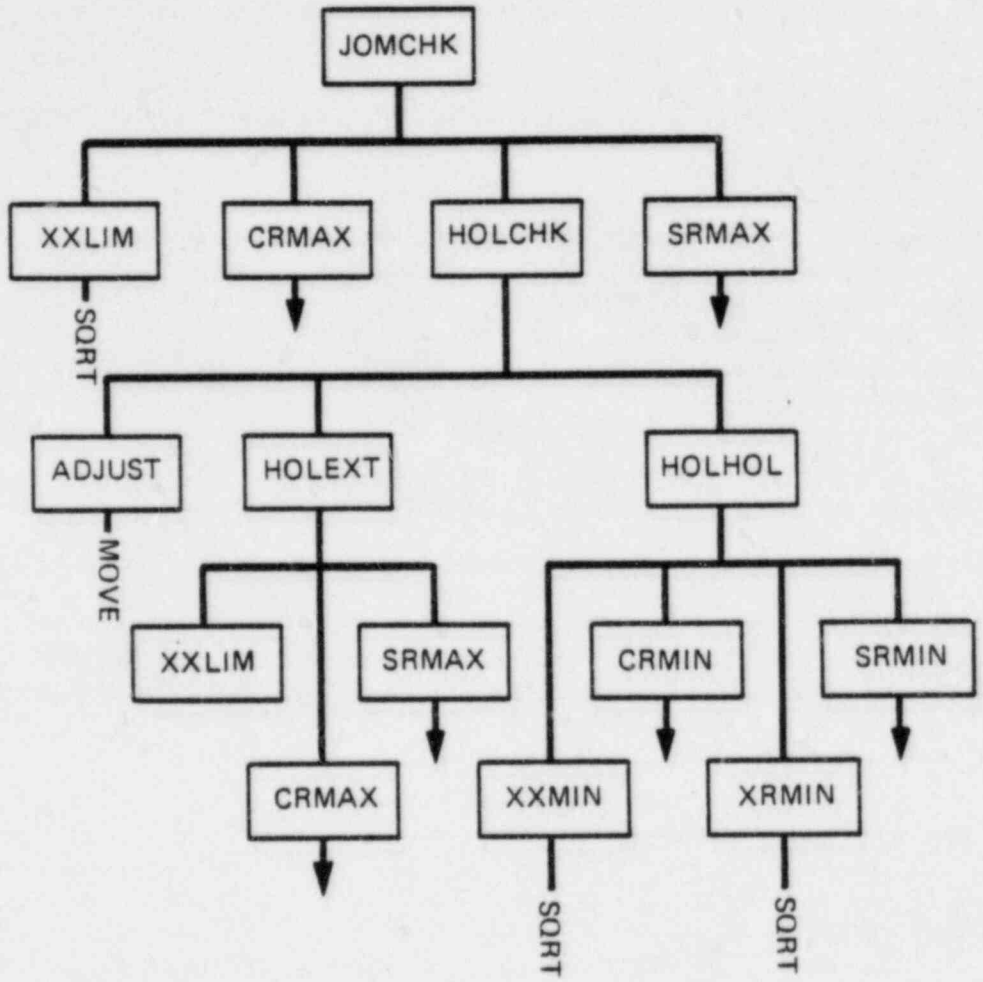


Fig. F11.3.23. Flow chart of geometry checking procedure

This portion of the program checks the geometry for inconsistencies, surface intersections, and other errors.

JOMCHK - This subroutine checks each geometry region to assure that it does not intersect the next outer region. If an intersection occurs, an error flag is set and an error message is written. JOMCHK checks each surface of the outer region for an intersection. If the surface is a planar surface, XXLIM is used to return the farthermost point of the inner region in the direction of the plane specified in the call (i.e., for the -x face, XXLIM returns the most negative x value of the inner region). For a spherical surface, SRMAX is used to return the length of the maximum radius vector of the inner region with respect to the origin of the outer region. For a cylindrical surface, CRMAX is used to return the length of the maximum radius vector of the inner region with respect to the axis of the outer region.

CRMAX - This function determines the maximum cylindrical radius vector of a geometry region with respect to a given axis, and then returns the magnitude of that vector. See Sect. F11.3.10.2.1 for additional details.

XXLIM - This routine returns the maximum coordinate of an interior region corresponding to a particular face direction (for negative face directions maximum means most negative).

SRMAX - This function determines the maximum radius vector of a geometry region with respect to a given origin, and then returns the magnitude of that vector. See Sect. F11.3.10.2.1 for additional details.

HOLCHK - This subroutine is responsible for assuring that a hole doesn't intersect any other geometry region. ADJUST is called to adjust the dimensions of the hole with respect to the origin of the unit that contains the hole. HOLEXT is called to check for a hole intersecting the region external to it. HOLHOL is used to check for a hole intersecting the region internal to the region that contains the hole. It is also used for checking for the intersection of two holes.

ADJUST - This subroutine corrects the dimensions of a region represented as a hole, with respect to the origin of the unit that contains the hole.

HOLEXT - This subroutine checks for a hole intersecting the region external to it. XXLIM is used to return the farthermost point of the hole in the direction of the plane specified in the call. CRMAX is used for a cylindrical surface to return the length of the maximum radius vector of the hole with respect to the axis of the outer region. SRMAX is used for a spherical surface to return the length of the maximum radius vector of the hole with respect to the outer region.

HOLHOL - This subroutine is responsible for assuring that a hole doesn't intersect the geometry region internal to the region that contains the hole. It also checks for holes intersecting each other. XXMIN is used to return the nearest point of the hole in the direction of the plane specified in the call. CRMIN is used for a cylindrical surface to return the length of the minimum radius vector of the hole with respect to the axis of the outer region. XRMIN determines the minimum coordinate of a hole with respect to a flat circular face. SRMIN is used for a spherical surface to return the length of the minimum radius vector of the hole with respect to the region being checked for an intersection.

XXMIN - This function returns the minimum coordinate of the hole with respect to the face of the cuboid specified in the call. SQRT is utilized in determining the minimum coordinate of spherical or cylindrical holes.

CRMIN - This function returns the magnitude of the minimum cylindrical radius vector of a hole with respect to a given axis. See Sect. F11.3.10.2.1 for additional details.

XRMIN - This function determines the minimum coordinate of a hole with respect to a flat circular face. SQRT is used in processing spherical and cylindrical holes.

SRMIN - This function returns the magnitude of the minimum radius vector from the center of a sphere to another geometry region. See Sect. F11.3.10.2.1 for additional details.

F11.3.10.2.1 *Determine Distances*

ORNL - DWG 83 7632

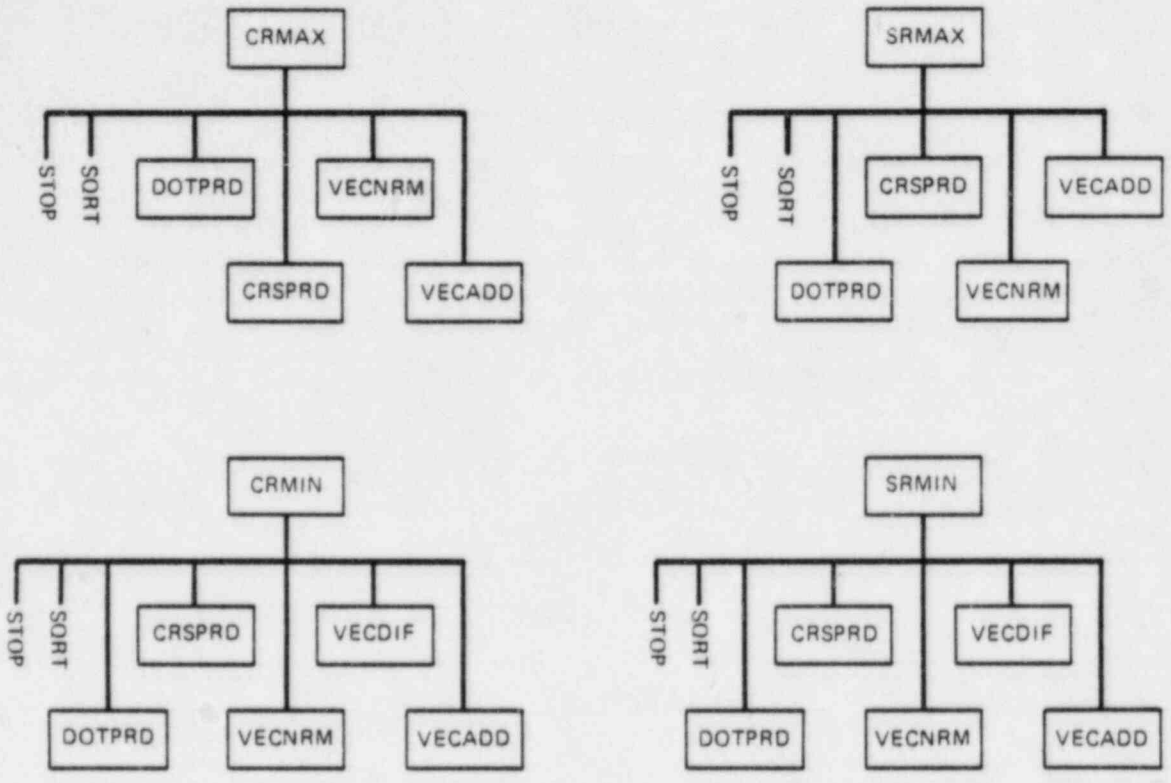


Fig. F11.3.24. Flow chart for distance to intersection

This section of the geometry checking procedure is utilized in checking for geometry surface intersections.

CRMAX - This function determines the maximum cylindrical radius vector of a geometry region with respect to a given axis, and then returns the magnitude of that vector. Depending on the geometry type, CRMAX may call CRSPRD to generate the cross product of two vectors, DOTPRD to generate the dot product of two vectors, VECADD to add two vectors together and VECNRM to multiply a vector by a scalar. STOP is called if geometry inconsistencies are encountered.

SRMAX - This function determines the maximum radius vector of a geometry region with respect to a given origin, and then returns the magnitude of that vector. Depending on the geometry type, SRMAX may call CRSPRD to take the cross product of two vectors, DOTPRD to take the dot product of two vectors, VECADD to add two vectors together, and VECNRM to multiply a vector by a scalar. STOP is called if geometry inconsistencies are encountered.

CRMIN - This function determines the minimum cylindrical radius vector of a geometry region with respect to a given axis and returns the magnitude of that vector. The geometry type determines which functions will be used to calculate the minimum radius vector. DOTPRD is used to generate the dot product of two vectors, CRSPRD generates the cross product of two vectors, VECNRM multiplies a vector by a scalar, VECDIF subtracts two vectors, and VECADD adds two vectors together. STOP is called if geometry inconsistencies are encountered.

SRMIN - This function determines the minimum radius vector of a geometry region with respect to a given origin (center of a sphere) and returns the magnitude of that vector. Depending on the geometry type, SRMIN may call DOTPRD to take the dot product of two vectors, CRSPRD to take the cross product of two vectors, VECNRM to multiply a vector by a scalar, VECDIF to subtract two vectors, and VECADD to add two vectors. STOP is called if geometry inconsistencies are encountered.

CRSPRD - This is a utility routine to generate the cross product of two vectors.

DOTPRD - This function returns the dot product of two vectors.

VECADD - This subroutine returns the result of adding two vectors.

VECNRM - This routine scales a vector by a constant.

VECDIF - This subroutine returns the difference of two vectors.

F11.3.10.3 *Calculate Volumes*

ORNL–DWG 83-7633R

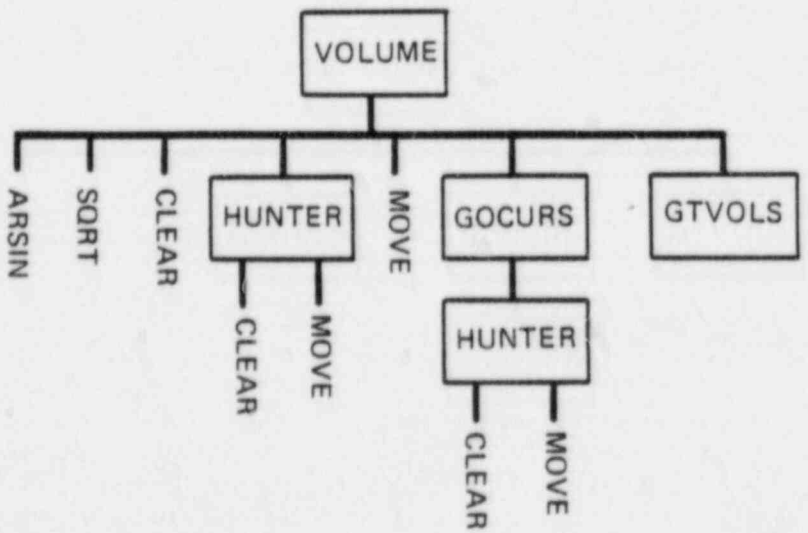


Fig. F11.3.25. Flow chart for calculating volumes

This portion of the program is responsible for calculating the volume of each region used in the problem, the cumulative volumes for each unit used in the problem, the number of times each unit was used in the problem, and the total volume of each region summed over all occurrences.

VOLUME - This subroutine calculates the volume of each region for every unit that is used in the problem. It then calculates the cumulative volumes for each unit. CLEAR is used to initialize arrays. If an external reflector is present, HUNTER is called to determine the number of times each array and/or hole is used in the reflector. GOCURS is used to determine the number of times each unit, array and hole is used in the problem. GTVOLS is called to calculate and print the number of occurrences for each unit and the corresponding total volumes for the entire system.

HUNTER - This subroutine determines the number of times each unit, array and hole is used. CLEAR is used to initialize storage arrays for the present hole level and the next hole level. MOVE is used to move the storage arrays.

GOCURS - This subroutine loops over the array size and calls HUNTER to determine the number of times each unit, lattice or array, and hole is used in the problem.

GTVOLS - This subroutine calculates the total volume of each region for the entire problem by multiplying the volume of the region by the number of times the region is used in the problem.

F11.3.11 LOAD BIASING OR WEIGHTING DATA

ORNL-DWG 80-19225R2

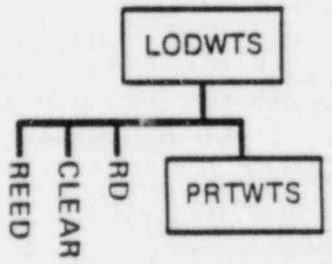


Fig. F11.3.26. Flow chart for loading biasing data

This portion of the program is executed only if the input parameter data contain PWT=YES, as described in Sect. F11.4.3. If biasing data are to be printed, the program loads and prints the average weight array.

LODWTS - This subroutine is responsible for printing the bias ID's versus the material ID's used in the problem and for loading and printing the biasing or weighting data. The number of sets of data that will fit in the available memory is calculated. Then the library routine CLEAR is used to initialize that space. RD is used to load the data that will fit and PRTWTS is called to print it. Then the entire process is repeated until all the biasing or weighting data used in the problem have been loaded and printed.

PRTWTS - This subroutine prints the group-dependent weight average array for each biasing region in a compact fashion.

F11.3.12 GENERATE PRINTER PLOT

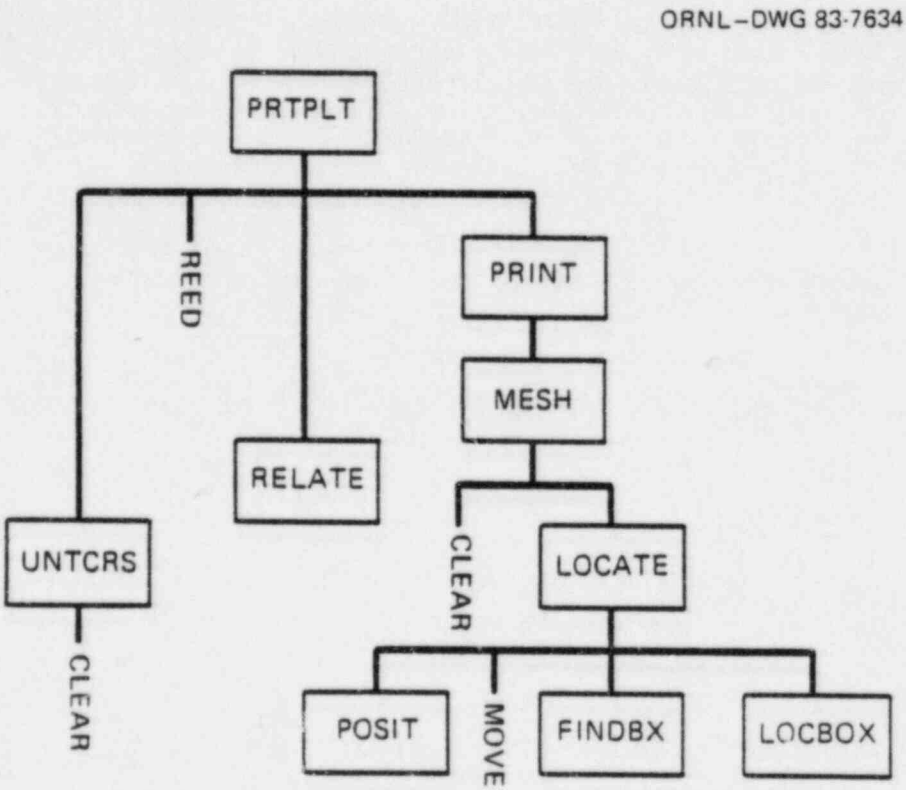


Fig. F11.3.27. Flow chart for pictures or plots

This portion of the program generates printer plots of a two-dimensional slice through the geometry. As many plots as are desired can be printed.

PRTPLT - This subroutine controls the generation of the printer plots. If the plot is specified to print by unit numbers, subroutine UNTCRS is called to generate a unit correspondence array. REED is used to load the picture data from direct access. The picture title is printed and subroutine RELATE is called to print a heading for the symbol map and print the symbol map. The picture coordinates, direction cosines and number of symbols across and down the page as well as the step intervals are printed. Then subroutine PRINT is called to generate the actual picture.

PRINT - This subroutine determines the number of pages that will be needed to print the picture. Subroutine MESH is called to load the appropriate mixture numbers, unit numbers or bias ID numbers for each line of print for the picture. Then PRINT prints the line of symbols corresponding to them.

MESH - This subroutine loads an array that contains the appropriate mixture number, unit number or bias ID number for each character in a line. CLEAR is called to initialize arrays if nested holes or nested arrays are present in the problem. LOCATE is called to determine the geometry region and unit or box type that contains each mesh point. MESH then loads the mixture number, unit number, or bias ID number.

LOCATE - This subroutine is responsible for determining the geometry region for each mesh point in the picture. If the mesh point is within an array, FINDBX is called to determine the position within the lattice or array that contains the mesh point. LOCBOX is then called to determine the unit or box type that contains the mesh point. POSIT is used to determine the geometry region that contains the mesh point.

FINDBX - This subroutine locates the position in an array that contains a specified point.

LOCBOX - This function returns the unit or box type at a given position in an array.

POSIT - This subroutine determines the region within a unit that contains a specified point.

F11.3.13 PROCESS HISTORIES BY SUPERGROUP

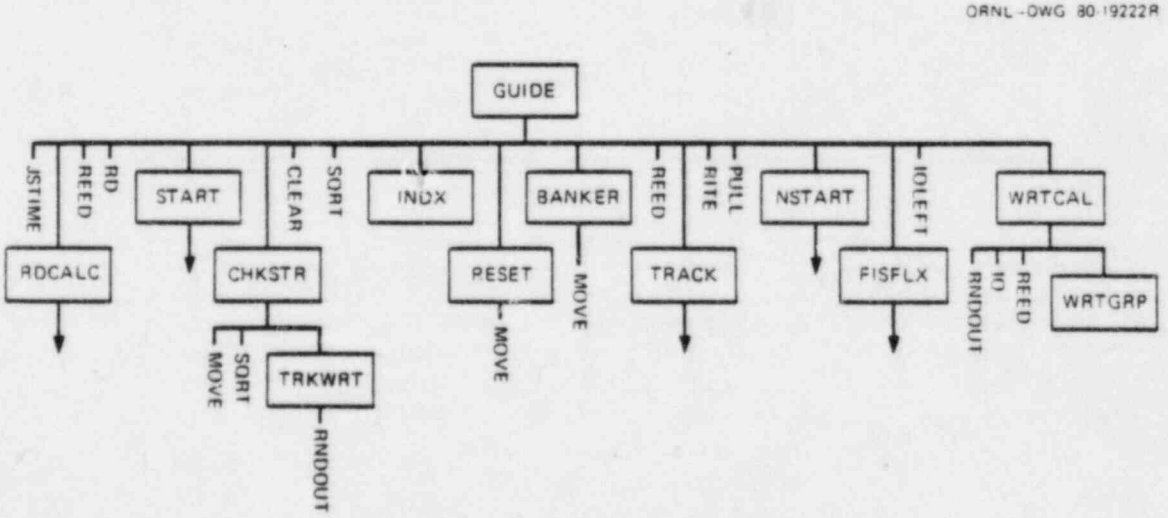


Fig. F11.3.28. Flow chart of tracking routines

This section of the program is where the tracking of the individual histories is done, one supergroup at a time.

GUIDE - This subroutine controls the tracking procedure. It loads the calculated data for a restarted problem, calls subroutine START to obtain the initial source distribution, calls CHKSTR to load the initial starting distribution in COMMON /NUTRON/ and prints the starting points as requested by the data. The library routine CLEAR is called several times to initialize arrays. IOLEFT and JSTIME are called to initialize the I/O's and time for the tracking procedure. SQRT is used in estimating the lower limit of the 99% confidence interval of the sample distribution of k-effective.

The heart of subroutine GUIDE is a loop over generations, from the starting generation to the number of generations requested. Subroutine RESET is called to accumulate the fission source and the source vectors for matrix k-effective. It also counts the number of histories in each supergroup and determines the supergroup with the largest number of histories. Then subroutine BANKER is called to sort the histories by supergroup, loading the largest supergroup at the top of the bank. GUIDE calculates pointers for the supergrouped data and loads the supergrouped data using the library routine REED. TRACK is called to do the actual tracking. The library routine RITE writes the calculated supergrouped data on the direct access supergroup data file. When all the supergroups have been processed, NSTART is called to provide the fission source for the next generation. Then FISFLX is called to calculate matrix information and statistics for the calculated data. GUIDE then checks to be sure sufficient time and I/O's remain to assure that another generation can be processed. If another generation cannot be processed and restart data are to be written, WRTCAL is called to write the calculated data on the restart data file. A message is printed by GUIDE stating the reason for terminating the calculation.

GUIDE is a very important subroutine in KENO V. The following table is provided to assist in understanding the functions performed by the subroutines called by GUIDE.

SUBROUTINE	FUNCTION	CONDITION
JSTIME	monitor time usage	always
RDCALC	load calculated data for restarting a problem	if the problem is restarted
REED	load data from direct access supergroup data file	always
RD	load data from direct access	always
START	provide initial source	if the problem is started with the first generation
CHKSTR	print starting points	if requested
INDX	locate cross-section index	always
RESET	count histories/supergroup	always
BANKER	sort the neutron bank	always
TRACK	track individual histories	always
RITE	write data on direct access	always
NSTART	provide fission source for the next generation	always
FISFLX	calculate statistics	always
IOLEFT	monitor I/O usage	always
PULL	terminate problem	if excessive time is used
WRTCAL	write data on restart file	if a restart data file is to be created

JSTIME - This library routine is called several times from GUIDE for timing purposes.

RDCALC - This subroutine is called only if the problem is being restarted at a generation greater than 1. Its purpose is to read data from the restart data file and write the supergrouped data on the direct access data file. See Sect. F11.3.13.1 for additional information.

REED - This library routine is called early in GUIDE if a problem is not being restarted in order to load data that are needed to create the initial source distribution. REED is called to load supergrouped data for the supergroup being processed from the direct access supergroup file.

RD - This library routine is called to load the fission spectrum for use in creating the initial source distribution.

START - This subroutine is responsible for creating the initial source distribution. See Sect. F11.3.13.2 for additional details.

CHKSTR - This subroutine calls the library routine MOVE to load the initial source distribution into COMMON /NUTRON/. If the starting points are to be printed, TRKWRT is called to print them using the debug tracking format.

TRKWRT - This subroutine is called from CHKSTR to print information about the current status of the current neutron in the debug tracking format. When TRKWRT is called from CHKSTR, the information of interest is the position at which the neutron was started. RNDOUT is called from TRKWRT to make the current random number available for printing.

CLEAR - This library routine is called to initialize the fission density array and the arrays that hold the matrix k-effective and associated statistics.

INDX - This subroutine is called only if the average number of neutrons per fission and the average energy at which fission occurs are to be calculated. (See Sect. F11.4.3, NUB=.) INDX determines the position of the fission cross section in the extra 1-D cross-section array. The fission cross section is required for calculating the average energy at which fission occurs.

RESET - This subroutine is called to accumulate the fission source and the source vectors for the matrix k-effective and to count the number of histories in a supergroup.

BANKER - This subroutine is responsible for sorting all the particles in the current supergroup into the top of the neutron bank and all other particles into the bottom of the bank.

TRACK - This subroutine is responsible for the actual tracking of each individual history. See Sect. F11.3.13.3 for a detailed description.

RITE - When the program returns from TRACK, RITE is called to write the calculated supergrouped data on the direct access supergroup data file.

PULL - This library routine is called from GUIDE to set a time interval that results in a nonstandard return if that time interval is exceeded. This is for the purpose of preventing the program from looping indefinitely. PULL is called later in GUIDE to reset the time interval as appropriate.

NSTART - This subroutine is called from GUIDE to provide the fission source for the next generation. See Sect. F11.3.13.4 for specific details.

FISFLX - This subroutine calculates statistics for k-effective, the matrix k-effecive, fissions, absorptions, leakages, and fluxes. Section F11.3.13.5 contains additional details.

IOLEFT - This library routine is called to determine if sufficient I/O's remain to allow processing another generation.

WRTCAL - If a restart data file is to be created, this subroutine writes the calculated information on the restart data file with a frequency specified by the parameter data (Sect. F11.4.3, RES=). RNDOUT is called to preserve the random number. Then the generation number, random number, number of histories per generation, number of energy groups, banked information, some common information and all the k-effectives calculated to this point are written on the restart data file. The neutron bank and, if requested, the fission densities are written. If matrix k-effective information is requested, IO is used to write it on the restart data file. Then WRTCAL loops over the number of supergroups, calculating pointers, using REED to load data from the direct access supergroup file and calling WRTGRP to write the group-dependent calculated information on the restart data file. When all the data have been written, a message is printed.

WRTGRP - This subroutine writes calculated data (leakages, absorptions, fissions, and if requested, fluxes) for each energy group on the restart data file.

F11.3.13.1 *Load Calculated Restart Data*

ORNL–DWG 80-19223

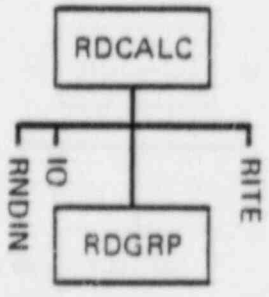


Fig. F11.3.29. Flow chart for loading calculated restart data

This section of the program loads the calculated data such as k-effectives and fluxes from the restart data file if the starting generation number is greater than 1.

RDCALC - This subroutine is called from GUIDE if the starting generation number is greater than 1. This indicates that a calculation is to be restarted using the starting generation number as the first generation to be processed. Thus all the results that were calculated in a previous run must be loaded from the restart data file to continue the calculation. RDCALC reads the previously calculated data from the restart data file and checks for consistency against parameters that were entered as input data. Appropriate messages are printed if inconsistencies are encountered. RNDIN may be called to load a new random number. IO is used to load data from the restart data file. Pointers are calculated for the supergrouped restart data and RDGRP is called to read the group-dependent data from the restart data file. RITE is used to write the restart data on the supergroup data file.

RNDIN - This library routine is called to load a new random number for use with the restarted problem if a random number was entered in the parameter input data.

IO - This library routine is used to load matrix k-effective information if it is to be calculated.

RDGRP - This subroutine is used to load the supergrouped data from the restart data file. This includes leakages, fissions, absorptions and fluxes.

RITE - This library routine is used to write previously calculated results on the direct access supergroup data file so the problem can be restarted properly.

F11.3.13.2 *Generate Initial Source Distribution*

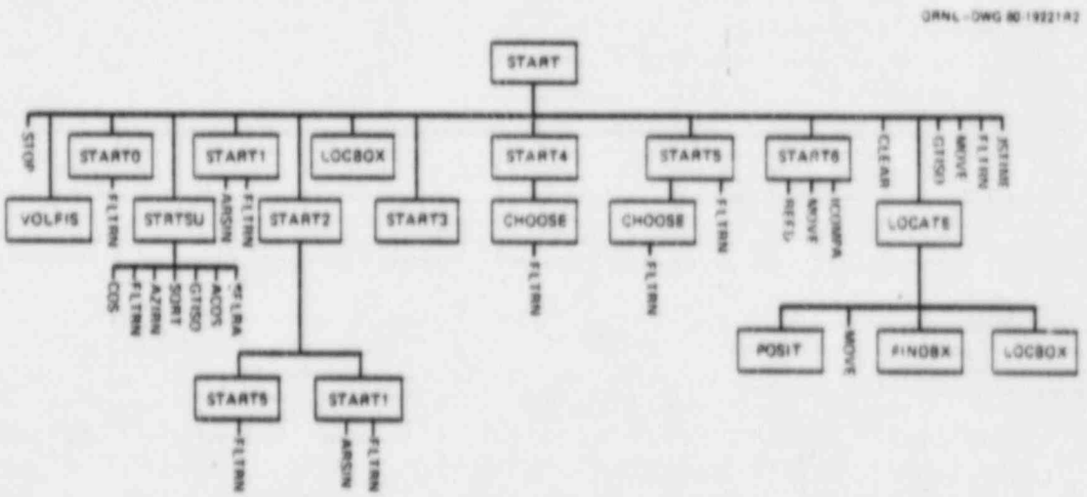


Fig. F11.3.30. Flow chart for providing initial source distribution

This portion of the program is responsible for generating the initial source distribution in accordance with information specified in the start data (see Sect. F11.4.8).

START - This subroutine is responsible for generating the initial source distribution. If fissile material is not used in the problem, STOP is called to write a message and terminate the problem. Appropriate messages are printed if the specified start type is incompatible with the geometry configuration. Subroutine VOLFIS is called to calculate the volume fraction of fissile material. Then START0, STRTSU, START1, START2, START3, START4, START5, or START6 is called to generate starting points having the characteristics specified in the start data (see Sect. F11.4.8). Subroutine LOCBOX is called after both START2 and START3 to determine the unit number at the specified position in the array. Subroutine LOCATE then determines the geometry region that contains the specified point by utilizing POSIT, FINDBX and LOCBOX. FINDBX is called to locate the position within the array that contains the starting point. LOCBOX is called to determine the unit number located at that position in the array. The starting point is translated to the coordinate system of the unit and POSIT is called to determine which region within the unit contains the starting point. A check is then made to be sure the region contains fissile material. If it does not, the point is discarded. If the region contains fissile material, GTISO is called to provide the initial direction cosines. START uses FLTRN to set the initial energy group and MOVE is called to load the initial data for the history into the neutron bank. JSTIME is called to be sure the allowed time is not exceeded. If it is, or if the required number of source neutrons have been generated, the starting is terminated. If too few initial source neutrons exist, FLTRN and MOVE are used to fill the remaining starting positions from those that were generated. A message to that effect is then printed.

STOP - This library routine is called from START to write an error message if fissile material is not utilized in the problem description. At least one mixture having a nonzero fission spectrum must be used in the problem description.

VOLFIS - This subroutine determines the volume fraction of fissile material in an unreflected array or an array whose reflector material is not fissile. It determines the volume fraction of fissile material in the system for single unit problems and reflected problems having fissile reflector material. If the volume fraction of fissile material is found to be zero, an error message is written and execution is terminated.

STARTO - This subroutine is called from START to generate a uniform initial source distribution in a cuboidal volume. This is accomplished by choosing points uniformly throughout the volume by using the library routine FLTRN, and discarding points that do not occur in fissile material. See Sect. F11.4.8 and Table F11.4.6 for assistance in modifying the default boundaries over which the starting points are chosen.

STRTSU - This subroutine is called to generate a uniform initial source distribution for a single unit problem or a reflected problem for which the user has specified that the reflector be included in the starting distribution. FLTRN is used to choose points uniformly throughout a cuboidal volume. AZIRN and FLTRN are used to choose points uniformly throughout a cylindrical or hemicylindrical volume. GTISO and FLTRN are used to choose points uniformly throughout a spherical or hemispherical volume.

AZIRN - This library routine provides the sine and cosine of a random azimuthal angle.

GTISO - This library routine provides the direction cosines of an isotropically distributed random direction. It is used to generate an isotropic source distribution.

START1 - This subroutine is used to provide starting points chosen from a cosine distribution. The library routines ARSIN and FLTRN are used to provide the cosine distribution. See Table F11.4.6 for details of the various initial starting distributions.

ARSIN - This library routine is the arcsine function, y=arcsine(x), and is used in generating a cosine source distribution.

START2 - This subroutine starts a specified fraction of the initial source distribution uniformly in fissile material in the unit located at a specified position in the global array. The remainder of the initial source is chosen from a cosine distribution as described in Table F11.4.6. START5 is called to determine the source points located in the unit at the specified location, and START1 is called to choose points from a cosine distribution.

LOCBOX - This function is called to return the unit or box type at a given position in a given array.

START3 - This subroutine starts all the initial source neutrons at a specified point within the unit located at a specified position in the global array.

START4 - In this subroutine, a unit type is specified for starting the initial source distribution. A uniform sampling is made over the global unit orientation array to locate units of the specified type, and start the initial source neutrons at a specified positions within these units. CHOOSE is called to determine the positions within the global array that are occupied by the specified unit type.

CHOOSE - This subroutine locates the positions of a specified unit within the global unit orientation array by randomly choosing positions in the array and discarding them if the specified unit was not at that position. The library routine FLTRN is used to randomly choose positions in the unit orientation array.

START5 - This subroutine starts the initial source neutrons uniformly in fissile material within a specified unit type. A uniform sampling is made over the global unit orientation array to locate units of this type by utilizing subroutine CHOOSE.

START6 - This subroutine starts the initial source neutrons at points specified by the user. These points must be specified relative to the origin of the global array. See Table F11.4.6 for exact specifications.

LOCATE - This subroutine determines the geometry region that contains the specified starting point. FINDBX is used to determine the point's location in an array. LOCBOX determines the unit number at that location and POSIT determines the region within the unit.

POSIT - This subroutine determines the region within the unit that contains the specified position.

FINDBX - This subroutine locates the position in an array that contains a specified point.

LOCBOX - This function returns the unit or box type for a specified position in an array.

MOVE - This library routine is called from START to load data from COMMON /NUTRON/ into the neutron bank. It is also used to pad the neutron bank to provide enough starting positions if too few were initially created.

FLTRN - This library routine is called from various subroutines during creation of the initial source distribution, to return a random number between zero and 1. START calls FLTRN to aid in choosing the initial source neutrons. It is also called if too few starting positions were generated. It is used to randomly choose from the initially created starting positions to pad the neutron bank until sufficient starting points exist.

JSTIME - This library routine is used for timing purposes.

F11.3.13.3 *Track Individual Histories*

ORNL-DWG 80-19218R

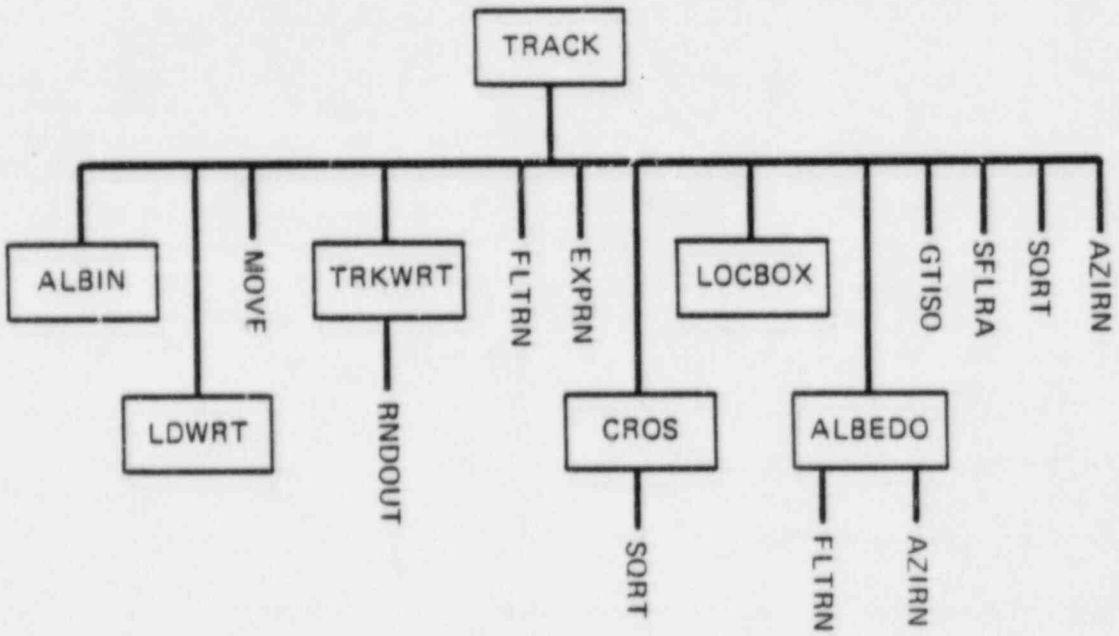


Fig. F11.3.31. Flow chart for tracking individual histories

This section of the program does the actual tracking of the individual histories.

TRACK - This subroutine is called by GUIDE to accomplish the actual tracking of the individual histories. Each history is tracked, and its contributions to the various calculated results are tabulated, until it escapes from the system or is killed via Russian roulette. If a history changes supergroups as a result of a collision or an albedo reflection, it is stored in the neutron bank. In the course of tracking a history, an initialization call is made to subroutine ALBIN if differential albedo boundary conditions are utilized in the problem. LDWRT may be called to print debug information and, if history tracks are to be printed, TRKWRT is called from strategic locations throughout TRACK to provide pertinent information about the history as it moves through the tracking process. The library routine MOVE is utilized throughout the tracking procedure to move data in and out of storage arrays and commons. FLTRN is used to provide random numbers used in playing Russian roulette, processing downscatters, picking fission points, picking scattering angles and determining the fission energy group. EXPRN provides a random number, selected from an exponential distribution, to be used as the number of mean free paths a history can traverse. CROS is called to determine if a boundary crossing has occurred. LOCBOX is called to determine the unit at a specified location in an array. ALBEDO is called to process differential albedo boundary conditions. The library routine GTISO is used to provide direction cosines from an isotropic distribution. SFLRA provides a random number between -1.0 and 1.0 for use in processing anisotropic scattering. SQRT is used in calculating the direction cosines of a history after it has a collision. AZIRN provides the sine and cosine of a random azimuthal angle for use in the anisotropic scattering treatment of a collision.

ALBIN - This subroutine contains the entry point ALBEDO. An initialization call is made to ALBIN by TRACK.

LDWRT - This subroutine prints debug information that is useful only for a programmer. It is called if BUG=YES is specified in the parameter data (see Sect. F11.4.3). In normal operation, this subroutine should never be called.

MOVE - This library routine is utilized frequently in TRACK to move data in and out of storage arrays and commons.

TRKWRT - This subroutine is called from various locations in TRACK to print information about the current neutron as it is being processed. RNDOUT is called from TRKWRT to make the current random number available for printing.

FLTRN - This library routine provides a random number between zero and 1. TRACK utilizes these random numbers for playing Russian roulette, processing downscatters, picking fission points, determining scattering angles and determining the fission energy group.

CROS - This important subroutine is responsible for processing both inward and outward crossings (i.e., it determines when a history has moved out of one geometry region into another). It determines if a crossing has actually occurred, and if it has, the coordinates of the crossing are upgraded to give the crossing point. The fraction of the path length used is also determined.

CROS consists of eight major sections. Four geometrical packages exist; one each for cuboids, cylinders, spheres and hemispheres. Hemicylinders are special adaptations of the cylinder package. Each geometrical package contains programming for inward crossings and for outward crossings. The variable N indicates the type of crossing being processed; $N \leq 0$ checks for an inward crossing and $N > 0$ checks for an outward crossing. The variable M is the crossing indicator, indicating whether or not a real crossing occurs. M=0 means a real crossing does not occur. M=1 indicates a successful crossing. SQRT is the only library routine utilized in CROS. It is used in the cylinder and sphere packages to solve the quadratic equation for the fraction of the path length that is used.

LOCBOX - This function returns the unit or box type for a specified position in an array.

ALBEDO - ALBEDO is an entry point in subroutine ALBIN. It is responsible for processing a differential albedo reflection. The direction cosines for the face where the albedo reflection occurs are loaded, and the incident angle and the albedo energy group corresponding to the incident energy group are determined. The position of the albedo energy group within the supergroup, and the first cross-section energy group and the number of cross-section energy groups corresponding to it are determined. This is used to calculate the new cross-section energy group of the history. Then the returning angle and direction cosines are calculated. The history's weight is then corrected for the weight lost in the albedo portion of the problem. The weight lost in the albedo reflection is summed.

FLTRN - This library routine is called from ALBEDO to return a random number between zero and one that is used to select (1) the albedo energy group corresponding to the input energy group, (2) the returning albedo energy group, (3) the returning energy group corresponding to the returning albedo energy group and (4) the returning angle.

EXPRN - This library routine is called from TRACK to provide the number of mean free paths to the next collision. This random number is picked from an exponential distribution.

GTISO - This library routine is called from TRACK to provide direction cosines from an isotropic distribution. These direction cosines are utilized in processing isotropic scattering.

SFLRA - This library routine is called from TRACK to provide a random number between -1.0 and 1.0 for use in processing anisotropic scattering.

SQRT - This library routine is utilized by TRACK to calculate the direction cosines of a history that has experienced a collision.

AZIRN - This library routine is called from ALBEDO to return the sine and cosine of a random azimuthal angle which are used in determining the direction cosines of the returning history.

The remainder of this section is devoted to the logical flow of subroutine TRACK as illustrated by Fig. F11.3.27. The portions of TRACK performing specialized functions are denoted in the text by a descriptive name enclosed in quotes to distinguish them from subroutine names. These descriptive names correspond to the functions depicted in the flow chart.

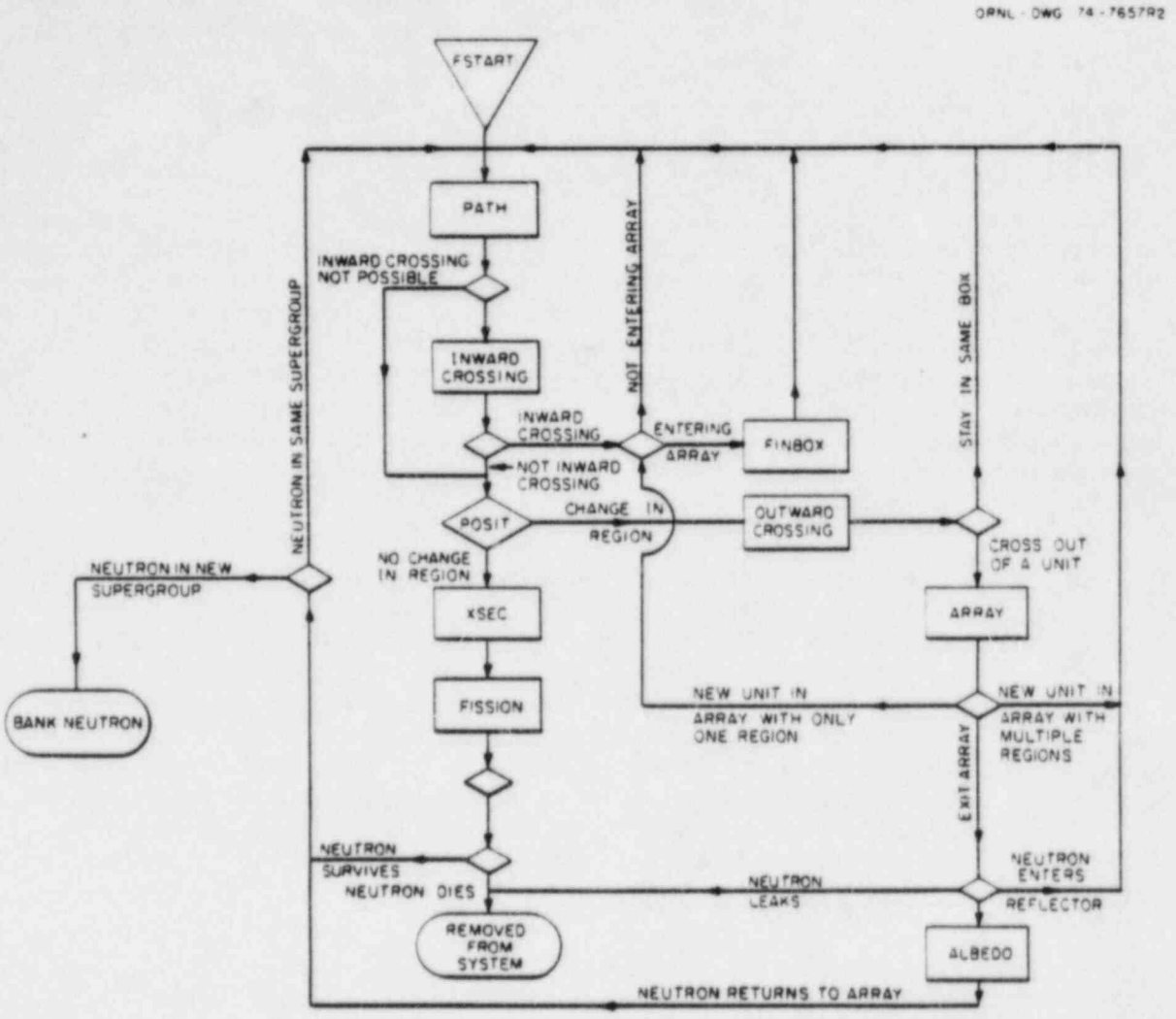


Fig. F11.3.32. Logic flow chart for subroutine TRACK

A brief discussion of the logical program flow through subroutine TRACK follows. If differential albedos are used, an initialization call is made to ALBIN. If debug print was specified in the parameter data, LDWRT is called. The library routine MOVE is used to load data from the neutron bank into COMMON /NUTRON/ and various flags are set. Then the fission source portion of TRACK, denoted "FSTART," initializes and sets information necessary for processing the history. The "PATH" portion of TRACK sets the path length and the end point of the path. The "INWARD" portion of TRACK determines if an inward crossing is possible and, if it occurs, the point of the boundary crossing. If the history is entering an array from an external region, the "FINBOX" portion of the code determines the history's position relative to the origin of the unit it is entering. If an inward crossing was not possible or did not occur, the "POSIT" portion of TRACK determines if the end point of the path is in the same region. If it is, a collision occurs and is processed in the portion of TRACK denoted as "XSEC." Based on the weight of the history, it is split and banked and/or Russian roulette is played. If the history survives Russian roulette, it is scattered and fission points may be generated in the FISSION portion of TRACK, and if it now falls in a different supergroup, it is banked. Otherwise it is retained in the system and processing returns to "PATH."

If "POSIT" determines that the end point of the path is in a different region, the "OUTWARD" portion of TRACK processes an outward crossing. If the history remains within the same unit, processing returns to "PATH" to continue as before.

If, after an outward crossing, the new region indicates that the history has crossed out of the unit it was in, the "ARRAY" portion of TRACK determines if the history is exiting from the array or entering a new path. If the history enters a new unit, processing returns to "PATH" and continues as before.

If the history is existing from the array, it can (1) enter the reflector, in which case processing returns to "PATH," (2) leak from the system, thus terminating that history or (3) undergo albedo reflection. The "ALBEDO" portion of TRACK can process a specular or mirror image reflection, a periodic reflection, or a differential albedo. For a mirror image or specular reflection, the history is returned at the point it exited with the weight and energy unchanged. The sign of the direction cosine perpendicular to the reflecting face is reversed. For a periodic reflection, the history is moved to the opposing face of the system and the weight, energy and direction cosines remain unchanged. For a differential albedo reflection the history returns at the point it exited, and subroutine ALBEDO determines the returning weight, energy and angle. The history is then returned to the PATH portion of TRACK to continue processing.

"FSTART" - This portion of TRACK calls the library routine MOVE to load information pertaining to the history to be tracked. Then logical flags are set to indicate if the history is in an array and whether or not the history is a split neutron. Variables are initialized and if the history tracks are to be printed, TRKWRT is called to print information about the history. If the history is the result of an albedo reflection that resulted in the history moving to a new supergroup, and the weight is large enough, the history proceeds to the "PATH" portion of TRACK. Otherwise Russian roulette is played. If it survives, the weight is set to the average weight and the history proceeds to the "PATH" portion of TRACK. If the history is a split neutron, variables are initialized and the history proceeds to the "XSEC" portion of TRACK to undergo the collision process.

"PATH" - This portion of TRACK determines the path length. If all the path length has been exhausted, the library routine EXPRN is used to define a new number of mean free paths from an exponential distribution. If the region contains a void, the distance traveled is set to the maximum chord length of the system. Otherwise the distance traveled is equal to the remaining path length, divided by the macroscopic total cross section of the mixture contained in the region. The end point of the path is determined from the starting coordinates, the distance traveled and the direction cosines. If the starting and ending coordinates are identical in any given direction, the end point is changed by a very small amount in the proper direction.

"INWARD" - This portion of TRACK decides whether or not an inward boundary crossing is possible, and if it is, calls CROS to determine if a crossing actually occurs and the coordinates of the crossing. Each time CROS is called, the library routine MOVE is used to load the geometry region dimensions of the possible new region into COMMON/ NUTRON/ to be used by subroutine CROS. The crossing indicator is set to instruct CROS to check for an inward crossing and the possible new region type is set. If the history tracks are to be printed, TRKWRT is called to print information pertinent to the crossing. Subroutine CROS sets a crossing indicator that informs TRACK whether a crossing occurs as well as determining the coordinates of the crossing. If a crossing occurs, the appropriate contributions are summed into the flux and neutron age. Then the number of mean free paths remaining for the history is determined.

When checking for an inward crossing, a check is made to determine if the history is in the innermost region of a unit and whether the innermost region contains holes. An inward crossing is not possible if the history is in the innermost region and that region does not contain holes. If holes are present in the innermost region, each hole in the region must be checked for a crossing.

If the history is not in the innermost region, an inward crossing is possible if holes are present, even if the last crossing was outward. CROS determines if an inward crossing actually occurs. If a crossing does not occur and no holes are in that region, tracking proceeds to the "POSIT" portion of TRACK. If a crossing did occur and holes are not present in that region, the flux and neutron age are updated and tracking proceeds to the "FINBOX" portion of TRACK. Tracking proceeds to the "POSIT" portion of TRACK if an inward crossing is not possible and no holes are in the region.

If holes are present in the region, after checking for an inward crossing, each of the holes is checked for an inward crossing. This involves transforming the coordinates of the history to the coordinate system of the hole and calling CROS to determine if the history crosses into the outer region of the hole. After repeating this procedure for each hole in the region, the crossing with the shortest distance is selected as the actual crossing. If the crossing is into a hole, the coordinates of the history are transformed to the coordinate system of the hole and tracking proceeds to the "FINBOX" portion of TRACK. The correct new region number is set whenever an inward crossing occurs.

"FINBOX" - This portion of subroutine TRACK is responsible for determining when a history enters a new array and for performing the transformation of coordinates whenever a history travels from a surrounding region into an array. It also determines the location within the unit orientation array of the unit containing the history. This information is used to determine the unit type, and the first and last regions of the unit.

"POSIT" - This portion of TRACK determines if the path ends in the same region in which it originated.

"OUTWARD" - An outward crossing occurs when the "POSIT" portion of TRACK determines that the history is entering a different region. The library routine MOVE loads the dimensions of the new geometry region into COMMON /NUTRON/ for use by subroutine CROS. The crossing-type indicator is then set to instruct CROS to process an outward crossing. If history tracks are to be printed, TRKWRT is called to print information pertinent to the outward crossing. CROS determines the coordinates of the boundary crossing. The contribution of the history is summed into the flux and neutron age and the number of mean free paths remaining for the history is calculated. The region number is incremented. If the old region was not the last region in the unit, a logical flag is set to avoid some of the checking for an inward crossing for the next boundary crossing. Tracking then proceeds to the "PATH" portion of TRACK. The region number is set to the last region in the unit, if the old region was the last region in the unit. If the problem is a single unit problem or the history is exiting the external reflector, the history proceeds to the "ALBEDO" portion of TRACK to leak from the system or process an albedo reflection. If the history has exited a unit in an array, tracking proceeds to the "ARRAY" portion of TRACK to continue the tracking process.

"COLLISION" - When a history has a collision, the processing is done in this portion of TRACK. If fluxes are to be calculated, the new contribution is summed in. The age of the history is summed, the remaining path length is set to zero, the absorption weight, fission weight and the contribution to the average number of neutrons produced per fission and the self-multiplication of the unit are calculated, based on the macroscopic cross-section data and the weight of the history. The weight of the history is then redefined to be the weight times the macroscopic nonabsorption probability. If the history tracks are to be printed, TRKWRT is called to print information pertinent to the collision process. If matrix k-effectives are to be calculated, the fission weight is summed into the proper arrays. If the weight of the history exceeds the weight at which splitting occurs, a check is made to assure the neutron bank has adequate space for another history. If it is full, a message is written and Russian roulette is played. If the bank can hold another history, the weight of the history is halved and the neutron counter is incremented. If the history tracks are to be printed, TRKWRT is called to print information pertinent to the split neutron. The library routine MOVE is used to store the split neutron in the neutron bank. The history cycles through the checking and splitting process until its weight is less than the weight at which splitting occurs. Then the weight is checked to see if Russian roulette should be played. If it is played and the history survives, the weight is set to the average weight. If Russian roulette is not played, the weight remains unchanged. In both cases, the new energy group is computed and a check is made to determine if the history undergoes anisotropic scattering. If it does, the azimuthal angle is chosen using AZIRN and the sine and cosine of that angle are returned to be used for calculating new direction cosines. If the history does not undergo anisotropic scattering, the new direction cosines are chosen from an isotropic distribution using GTISO. This completes the "COLLISION" portion of TRACK.

"FISSION" - This portion of TRACK is responsible for generating and storing the fission source resulting from a collision. In the "COLLISION" portion of TRACK, the fission weight is defined as the weight of a history times the macroscopic production probability. If the fission weight of a history is greater than zero, the "FISSION" portion of TRACK is executed. To assure generating enough fission source points to maintain an adequate representation of the true distribution, a minimum production factor is defined at the beginning of each generation to be

$$\frac{3.0\bar{k}}{\sqrt{FG}}$$

where $\bar{k}$ is the running average value of the k-effective through the current generation and FG is the number of histories per generation. This represents an estimate of the lower limit of the 99% confidence interval for the distribution of the generation k-effective. Experience indicates that using this factor to generate fission source points leads to enough new fission points to fill the neutron bank for most generations.

When the "FISSION" portion of TRACK is entered, the library routine FLTRN is used to provide a random number that is saved. A pseudo fission weight is defined as the fission weight divided by the random number. If the result is less than the production factor, the history proceeds to "PATH" or is stored in the neutron bank,

depending on whether or not it remains in the same supergroup. If the history remained in the "FISSION" section, and its fission weight is greater than the production factor, the pseudo fission weight is redefined to be the production factor divided by the random number. If the fission bank is not full, the fission energy group is determined using FLTRN, the fission point is stored in the bank, and the number of fission points is incremented. The library routine MOVE is used to load information pertaining to the fission point from COMMON /NUTRON/ into the fission bank and the pseudo fission weight is loaded directly into the fission bank. If the fission bank is full when a new fission source point is generated, the bank is searched for the smallest pseudo fission weight. This is compared with the pseudo fission weight of the new fission point and the point having the larger pseudo fission weight is stored in the bank. After the fission point has been banked, the fission weight of the history is decremented by the production factor. If the remaining fission weight is greater than zero, the history returns to the beginning of "FISSION" to continue processing.

"ARRAY" - This portion of TRACK is responsible for processing a history that crosses a unit or box type boundary. This occurs only when the history is moving in an outward direction. A check is made to determine if the history is exiting from a unit in an array or a unit used as a hole. If the exited unit was used as a hole, the coordinates are transformed to the coordinate system of the region containing the hole. The history then proceeds to the "PATH" portion of TRACK. However, if the unit exited by the history was part of an array, a check is made to see if tracks are to be printed. If so, TRKWRT is called to print pertinent information about the history. Each face of the unit is checked in sequence to force proper positioning for face, edge and corner crossings. As each face is processed, the position of the unit within the unit orientation array is updated as appropriate (this is done by incrementing the X, Y and/or Z indices of the unit position). A logical flag is set for each face to indicate if the history is exiting from the array. After all faces have been processed or bypassed, a check is made to determine if the history remains within the array. If it does, the new unit type is determined from the position indices of the location within the unit orientation array, and a transformation of coordinates is done to correct for crossing into the new unit. If the new unit consists of only one region, the history proceeds to "FINBOX" to determine if the history is entering a new array. The history then proceeds to "PATH" to continue processing. If the history exited the global array, a check is made to determine if a reflector is external to it. If not, the history leaks from the system or proceeds through the "ALBEDO" portion of TRACK as appropriate. If surrounding geometry exists where the history exits an array, a transformation of coordinates provides the history's position relative to the surrounding geometry. The history then proceeds to the "PATH" portion of TRACK and tracking continues.

"ALBEDO" - If an albedo boundary condition is specified on any face of the problem, this portion of TRACK is utilized to provide the proper treatment. Each face is checked in sequence to determine if the history (1) leaks from the system, (2) undergoes specular or mirror image reflection, (3) undergoes periodic reflection, or (4) proceeds through the differential albedo treatment.

If the history undergoes specular or mirror image reflection, it is returned at the point it exited the face with its energy unchanged and the sign of its direction cosines reversed. If a periodic reflection occurs, the history is moved to the opposing face with its energy and direction cosines unchanged. If the history enters a differential

albedo reflector, subroutine ALBEDO is called to determine the new weight of the history and its returning angle and energy. If the history enters a new supergroup, TRKWRT may be called to print the history track information and MOVE is called to bank the history in the neutron bank. When a history remains in the same supergroup, it is returned at the point it exited and Russian roulette is played if the returning weight is low enough to warrant that action. If the weight is sufficiently high to avoid playing Russian roulette or if the history survives Russian roulette, the history proceeds to the beginning of "PATH" and tracking continues.

F11.3.13.4 *Provide the Next Generation Source*

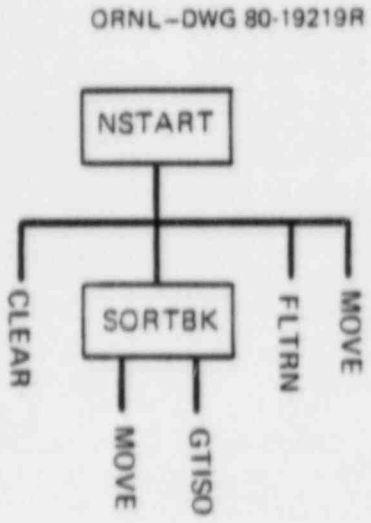


Fig. F11.3.33. Flow chart for providing the next generation source

This section of the program is responsible for providing the source for the next generation from the fission source generated during the tracking procedure.

NSTART - This subroutine calls CLEAR to initialize the neutron bank and writes an error message if no fission points were generated. Subroutine SORTBK is called to move information from the fission bank containing the fission source generated by the last generation into the neutron bank to be used as the source for the next generation. NSTART then checks to be sure enough source points exist to start the next generation. If too few starting points exist, the library routines MOVE and FLTRN are used to fill the required number of starting positions from the existing fission points.

SORTBK - This subroutine sorts the fission bank so the fission points are loaded in the order of their probability of being picked in a random selection process. For each source history, the library routine MOVE is used to move the fission source generated by the last generation into the neutron bank, and the library routine GTISO is used to provide direction cosines from an isotropic distribution. Then the neutron number, weight and age are initialized.

F11.3.13.5 *End of Generation Processing*

ORNL–DWG 80-19220R

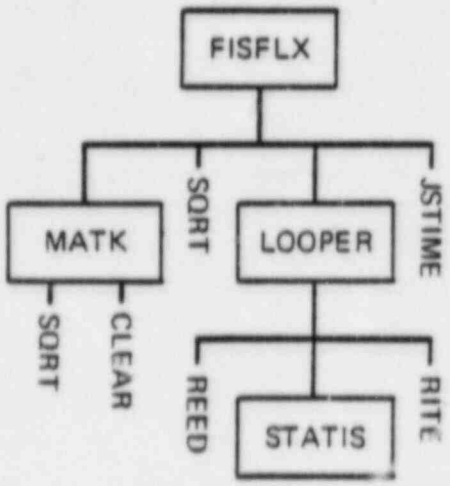


Fig. F11.3.34. Flow chart for end of generation processing

This portion of the program is responsible for processing data at the end of each generation.

FISFLX - This subroutine is called at the end of each generation to process data collected for that generation. The generation k-effective, the running average value of k-effective and its deviation, and the matrix k-effective and its deviation are processed and printed. MATK is called to process the matrix k-effective(s) and associated information. LOOPER is called and, in turn, calls STATIS to collect and process the contribution and statistics for the flux, fissions, and absorptions, as well as the contribution to the leakage.

MATK - This subroutine is called to calculate the matrix k-effective by solving for the principal eigenvalue and eigenvector of a matrix using an iterative technique. The library routine CLEAR is used to initialize arrays and SQRT is used in calculating the deviation of the eigenvalue. MATK may be called to calculate the matrix k-effective by array position, unit type, array number, and/or hole number.

LOOPER - This subroutine is called from FISFLX to load arrays in preparation for calling STATIS. A loop is made over the number of supergroups, within which pointers are calculated, REED is called to load the leakage, absorption, fission and flux arrays from the direct access supergroup file, STATIS is called to process the data and RITE writes the processed data on the direct access supergroup file. This procedure is repeated until all supergroups have been processed.

STATIS - This subroutine collects the sum of the contributions and the sum of the square of the contributions for the fluxes, fissions, absorptions and leakages to be used at the end of the problem in calculating their deviations.

F11.3.14 END OF PROBLEM PROCESSING

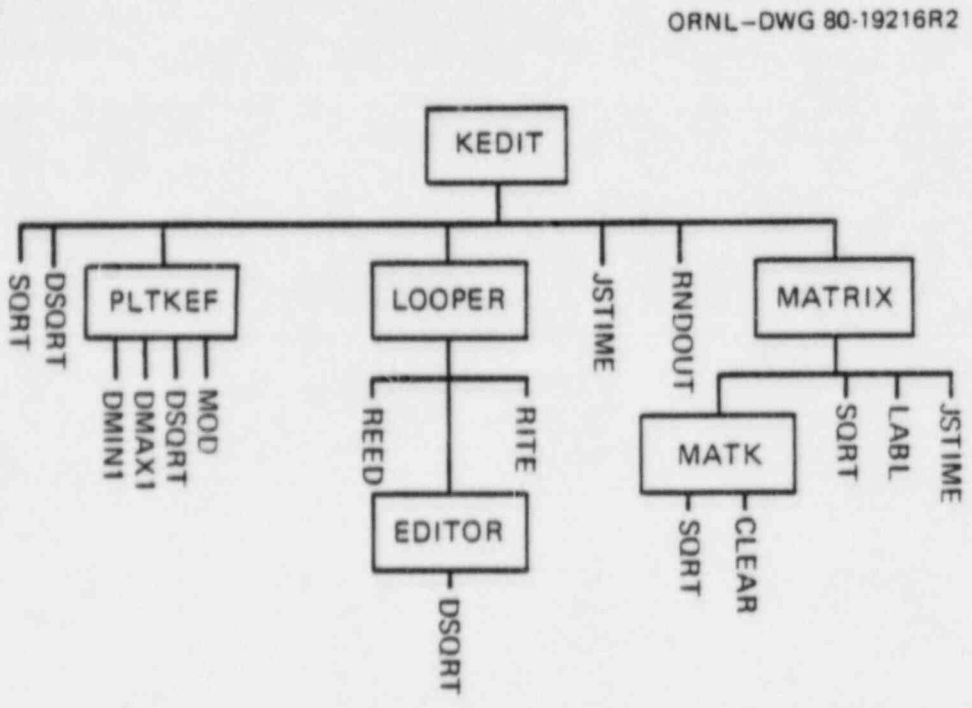


Fig. F11.3.35. Flow chart for end of problem processing

This portion of the program is responsible for processing data and printing all the results except the fluxes at the completion or termination of a problem.

KEDIT - This subroutine controls the processing and printing of results at the end of a problem. The life time and generation time are printed and, if the average number of neutrons per fission was calculated, it and the average fission group are printed. KEDIT then calculates and prints the average k-effective and its associated deviation for the 67, 95, and 99% confidence intervals and the number of histories involved. This is done repeatedly, skipping more generations each time. PLTKEF is called to print a plot of the average value of k-effective as a function of the number of generations. It also prints a plot of the average value of k-effective as a function of the number of generations skipped. LOOPER is called to prepare data for EDITOR which in turn calculates and prints the group-dependent fissions, absorptions and leakages and their deviations. If requested, the fissions and absorptions may also be printed by region. KEDIT then prints the totai fissions, absorptions and leakages for the system, and their associated deviations. RNDOUT is called to provide the current random number to be printed. If matrix k-effective data were requested, MATRIX is called to calculate and print matrix information. KEDIT then processes and prints the fission densities.

PLTKEF - This subroutine is called from KEDIT to print a plot of the average value of k-effective versus generation and a plot of the average value of k-effective versus generations skipped. The library routines DMIN1, DMAX1, and DSQRT are used to generate the k-effective axis of the plots. The MOD function is used in labeling the generation axis.

LOOPER - This subroutine is called from KEDIT to load arrays in preparation for calling EDITOR. A loop is made over the number of supergroups. Within the loop, pointers are calculated, REED is called to load the leakage, absorption, fission and flux arrays from the direct access supergroup file and EDITOR is called to process the data. RITE then writes the processed data for the supergroup on the direct access supergroup file.

EDITOR - This subroutine is called from LOOPER to calculate and print the energy-dependent fissions, absorptions and leakages and their deviations. The fissions and absorptions may be region dependent as well as energy dependent.

MATRIX - This subroutine is called from KEDIT to calculate and print various information related to the matrix k-effective. It is called if one or more matrix options were specified in the parameter data (see Sect. F11.4.3). If matrix information was collected, MATK is called to calculate cofactor k-effectives. MATRIX then prints them as they are calculated. MATRIX also prints the fission production matrix if that was specified in the parameter data (see Sect. F11.4.3). The library routine LABL is called to print the source vector. MATRIX then prints the average self-multiplication calculated on the basis of collected data. This procedure is repeated for each type of matrix information specified in the parameter data. JSTIME is called to determine the amount of time used in the problem, which is then printed in MATRIX.

MATK - This subroutine calculates the principal eigenvalue and eigenvector of a matrix using an iterative technique. It also calculates the deviation associated with the eigenvalue, using CLEAR to initialize arrays and SQRT in calculating the deviation. MATK may be called from MATRIX to calculate cofactor k-effectives.

F11.3.15 PRINT FLUXES

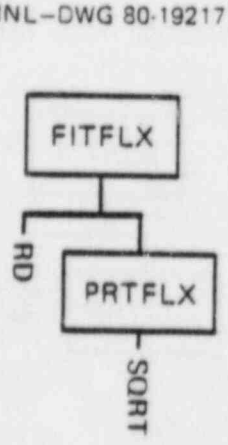


Fig. F11.3.36. Flow chart for printing fluxes

This portion of the program is responsible for printing the fluxes at the completion of a problem.

FITFLX - This subroutine determines the maximum number of regions for which flux data will fit in memory and loads and prints them as they will fit. Pointers are calculated and the library routine RD is used to load the fluxes for those regions from the direct access supergroup file. PRTFLX is called to calculate the deviations and print the region- and energy-dependent fluxes and their associated deviations. If more fluxes remain to be printed, the process is repeated until all have been printed.

PRTFLX - This subroutine normalizes the fluxes, calculates their deviations and prints them, one supergroup at a time. The library routine SQRT is used when calculating the deviations.

F11.3.16 REFERENCES

1. S. K. Fraley, *Users Guide for ICE-II*, ORNL/CSD/TM-9/R1 (July 1977). Also see Sect. F8 of the SCALE manual.

2. "Appendix B, Generalized Gaussian Quadrature," *The Morse Code - A Multigroup Neutron and Gamma-Ray Monte Carlo Transport Code*, ORNL-4585 (1970). Also see Sect. F9.D.2 of the SCALE manual.

F11.4 KENO V DATA GUIDE

KENO V may be run "stand alone" or as a part of a SCALE criticality safety sequence. If KENO V is run "stand alone," cross-section data can be utilized from an AMPX[1] working format library or from an ICE (Sect. F8) mixed cross-section MORSE/KENO format library, also called a Monte Carlo formatted cross-section library. If KENO V uses an AMPX working format library, a mixing table data block must be entered. If an ICE mixed cross-section MORSE/KENO library is used, a mixing table data block is not entered and the mixtures specified in the KENO V geometry description must be consistent with the mixtures created in ICE. These are the entries in the ITS[MUD] array in the ICE input data.

If KENO V is run as part of a SCALE criticality safety sequence, the mixtures are defined in the CSAS4 data (Sect. C4.4) and a mixing table data block cannot be entered in KENO V. Furthermore, the mixture numbers used in the KENO V geometry description must correspond to those defined in the standard composition cards of the CSAS input. A mixture number of 500 **must** be used in the KENO V geometry description in order to use a cell weighted mixture. A cell weighted mixture is available only in SCALE sequences that do a cell weighting calculation.

F11.4.1 KENO V INPUT OUTLINE

The data input for KENO V is outlined below. Defaulted data for KENO V have been found to be adequate for many problems. These values should be carefully considered when entering data. The information in **BOLD TYPE** is entered as data.

Blocks of input data are entered in the form:

READ XXXX input data END XXXX

where XXXX is the keyword for the type of data being entered. The keywords that can be used are listed in Table F11.4.1. A minimum of four characters are required for a keyword. However, the keywords can be up to twelve characters long, the first four of which **must be input exactly as listed in the table**. Data input is activated by entering the words **READ XXXX** followed by one or more blanks. All input data pertinent to **XXXX** are then entered. Data for **XXXX** are terminated by entering **END XXXX** followed by two or more blanks.

Table F11.4.1. Types of input data

Type of Data	First 4 Characters
parameters	PARA or PARM
geometry	GEOM
biasing	BIAS
boundary conditions	BOUN or BNDS
start	STAR or STRT
array (unit orientation)	ARRA
extra 1-D cross sections	X1DS
cross section Mixing Table*	MIXT or MIX
plot or picture *	PLOT or PLT or PICT

*MIX and PLT must include a trailing blank which is considered part of the keyword.

There are two data cards that **must** be entered for every problem. The first is the problem title. The second is the **END DATA** to terminate the problem.

(1) **problem title**

Enter a problem title (limit, one card, i.e., 80 characters including blanks). A title **must be entered.**
See Sect. F11.4.3.

(2) **READ PARA parameter data END PARA**

Enter parameter input as needed to describe a problem. Default values are assigned to all parameters. A problem **can** be run without entering any parameter data if the default values are acceptable.

Parameter data must begin with the words **READ PARA**. Parameter data may be entered in any order. If multiple entries are made for a parameter, the last value is used. The words **END PARA** terminate the parameter data.
See Sect. F11.4.3.

(3)...(9) The following data may be entered in any order. Data not needed to describe the problem may be omitted.

(n_1) **READ GEOM all geometry region data END GEOM**

Geometry region data **must be entered** for every problem that is not a restart problem. Geometry data must begin with the words **READ GEOM**. The words **END GEOM** terminate the geometry region data.
See Sect. F11.4.4.

(n_2) **READ ARRA array definition data END ARRA**

Enter array definition data as needed to describe the problem. Array definition data define the array size and position units (defined in the geometry data) in a three-dimensional lattice that represents the physical problem being analyzed. Array data must begin with the words **READ ARRA**. The words **END ARRA** terminate the array data.
See Sect. F11.4.5.

(n_3) **READ BIAS biasing information END BIAS**

Biasing information is used to define the weight that is given a neutron surviving Russian roulette. Enter biasing information as needed to describe the problem. Biasing data must begin with the words **READ BIAS**. The words **END BIAS** terminate the biasing data.
See Sect. F11.4.7.

(n_4) **READ BOUN albedo boundary conditions END BOUN**

Enter albedo boundary conditions as needed to describe the problem. Albedo data must begin with the words **READ BOUN** and terminate with the words **END BOUN**.
See Sect. F11.4.6.

(n_5) **READ STAR** **starting distribution information** **END STAR**

Enter starting information data for starting the initial source neutrons only if a uniform starting distribution is undesirable. Start data must begin with the words **READ STAR** and terminate with the words **END STAR.**
See Sect. F11.4.8.

(n_6) **READ MIXT** **cross section mixing table** **END MIXT**

Enter a mixing table to define all the mixtures to be used in the problem. The mixing table must begin with the words **READ MIXT** and end with the words **END MIXT.** Do not enter mixing table data if KENO V is run as a part of a SCALE criticality safety sequence.
See Sect. F11.4.10.

(n_7) **READ X1DS** **extra 1-D cross-section ID's** **END X1DS**

Enter the ID's of any extra one-dimensional cross sections that are to be used in the problem. These must be available on the mixture cross section library. Extra 1-D cross-section data must begin with the words **READ X1DS** and terminate with the words **END X1DS.**
See Sect. F11.4.9.

(n_8) **READ PLOT** **plot data** **END PLOT**

Enter the data needed to provide a two-dimensional printer plot of a slice through a specified portion of the three-dimensional geometrical representation of the problem. Plot or picture data must begin with the words **READ PLOT** and terminate with the words **END PLOT.**
See Sect. F11.4.11.

(n_9) **END DATA** **must be entered**

Terminate the data for the problem.

F11.4.2 PROCEDURE FOR DATA INPUT

This section is a brief list of the input data for KENO V. Additional information concerning KENO V data input may be found in Sect. F11.5. The first card of data **must** be the title. The next block of data **must** be the parameters if they are to be entered. A problem can be run without entering the parameters. The remaining blocks of data can be entered in any order.

BOLD TYPE specifies keywords. A keyword is used to identify the data that follow it. When a keyword is used, it must be entered exactly as shown in the data guide. All keywords, except those ending with an equal sign, must be followed by at least one blank.

small italics correlate data with a program variable name. The actual values are entered in place of the program variable name and are terminated by a blank or a comma.

CAPITAL ITALICS identify general data items. General data items are general classes of data including

(1) geometry data such as *UNIT INITIALIZATION* and *UNIT NUMBER DEFINITION, GEOMETRY REGION DESCRIPTION, GEOMETRY WORD, MIXTURE NUMBER, BIAS ID* and *REGION DIMENSIONS*,

(2) albedo data such as *FACE CODES* and *ALBEDO NAMES*,

(3) weighting data such as *BIAS ID NUMBERS*, etc.

F11.4.3 TITLE AND PARAMETER DATA

TITLE . . . A title must be entered.

title one card, 80 characters including blanks

PARAMETER DATA . . . Enter only those whose values you wish to change. The commonly changed parameters are *TME, GEN*, and *NPG*. Seldom changed parameters are *NBK, NFB, LBK, LFB, WTH, WTL, TBA, BUG, TRK*, and *LNG*.

READ PARAM

Floating point parameters

RND = *rndnum* input hexadecimal random number, a default value is provided.

TME = *tmax* execution time (in minutes) for the problem, default = 30 minutes.

TBA = *tbtch* time allotted for each generation (in minutes), default = 0.5 minutes. If *tbtch* is exceeded in any generation, the problem is terminated and final edits are performed.

WTA = *dwtav* the default average weight given a neutron that survives Russian roulette, *dwtav* default = 0.5.

WTH = *wthigh* the default value of *wthigh* is 3.0 and should be changed only if the user has a valid reason to do so. The weight at which splitting occurs is defined to be *wthigh* x *wtavg*, where *wtavg* is the weight given to a neutron that survives Russian roulette.

WTL = *wtlow* Russian roulette is played when the weight of a neutron is less than *wtlow* x *wtavg*. The *wtlow* default = 1.0/*wthigh*.

NOTE: The default values of *wthigh* and *wtlow* have been determined to minimize the deviation per unit running time for many problems.

Integer parameters

GEN = *nba* number of generations to be run, default = 103

NPG = *npb* number of neutrons per generation, default = 300

NSK = *nskip* number of generations (1 through *nskip*) to be omitted when collecting results, default = 3

RES = *nrstrt* number of generations between writing restart data, default = 0.
If **RES** is zero, restart data are not written. When restarting a problem, **RES** is defaulted to the value that was used when the restart data block was written. Thus, it must be entered as zero to terminate writing restart data for a restarted problem. (**WRS** is the logical unit number for writing restart data. See logical unit numbers in the parameter data.)

NBK = *nbank* number of positions in the neutron bank, default = *npb* + 25

XNB = *nxnbk* number of extra entries in the neutron bank, default = 0

NFB = *nfbnk* number of positions in the fission bank, default = *npb*

XFB = *nxfbk* number of extra entries in the fission bank, default = 0

X1D = *numx1d* number of extra 1-D cross sections, default = 0

LNG = *lng* number of words of storage to be requested by subroutine *ALOCAT*, default = 4000000. (This is reduced to fit in the space allotted to the job when the problem is run.)

BEG = *nbas* beginning generation number, default = 1
If *BEG* is greater than 1, restart data must be available.

NB8 = *nb8* number of blocks allocated for the first direct access unit, default = 200

NL8 = *nl8* length of blocks allocated for the first direct access unit, default = 789

Alphanumeric parameter data . . . enter *YES* or *NO*

RUN = *lrun* key for determining if problem is to be executed when data checking is complete, default = *YES*

Note: The value of *RUN* set here will be overridden by a value entered in the *PLOT* data. However, if the problem is restarted, the default value or the value entered here is the value that will be used unless the *PLOT* data block from the restart unit is overridden by new *PLOT* data. See Sect. F11.4.9.

FLX = *nflx* — key for collecting and printing fluxes, default = NO

FDN = *nfden* — key for collecting and printing fission densities, default = NO

ADJ = *nadj* — key for running adjoint calculation, default = NO. Adjoint cross sections must be available to run an adjoint problem. If LIB= is specified, the cross sections will be adjointed by the code. If XSC= is specified, the cross sections must already be in adjoint order.

AMX = *amx* — key for printing all mixture cross-section data. This is the same as activating *XAP, XS1, XS2, PKI,* and *P1D.* If any of these are entered in addition to *AMX,* that portion of *AMX* will be overridden, default = NO.

XAP = *prtap* — key for printing discrete scattering angles and probabilities for the mixture cross sections, default = NO

XS1 = *prtp0* — key for printing mixture 1-D cross sections, default = NO

XS2 = *prt1* — key for printing mixture two-dimensional cross sections, default = NO

PKI = *prtchi* — print input fission spectrum, default = NO

P1D = *prtex* — print extra 1-D cross sections, default = NO

FAR = *lfa* — key for printing fissions and absorptions by region, default = NO

MKP = *larpos* — calculate and print matrix k-effective by unit location, default = NO. Unit location may also be referred to as array position or position index.

CKP = *lckp* — calculate and print cofactor k-effective by unit location, default is the value of *MKP.* Unit location may also be referred to as array position or position index.

FMP = *pmapos* — print fission production matrix by array position, default = NO

MKU = *lunit* — calculate and print matrix k-effective by unit type, default = NO

CKU = *lcku* — calculate and print cofactor k-effective by unit type, default is the value of *MKU*

FMU = *pmunit* — print fission production matrix by unit type, default = NO

MKH = *lmhole* — calculate and print matrix k-effective by hole number, default = NO

CKH = *lckh* — calculate and print cofactor k-effective by hole number, default is the value of **MKH**.

FMH = *pmhole* — print fission production matrix by hole number, default = NO

HHL = *lhhgh* — collect matrix information by hole number at the highest hole nesting level, default = NO

MKA = *lmarry* — calculate and print matrix k-effective by array number, default = NO

CKA = *lcka* — calculate and print cofactor k-effective by array number, default is the value of **MKA**

FMA = *pmarry* — print fission production matrix by array number, default = NO

HAL = *langh* — collect matrix information by array number at the highest array nesting level, default = NO

PLT = *lplot* — key for drawing specified pictures of the problem geometry, default = YES

Note: To draw a plot, appropriate plot data must be entered. The value of *PLT* set here will be overridden by a value entered in the *PLOT* data. However, if the problem is restarted, the default value or the value entered here is the value that will be used unless the *PLOT* data block from the restart unit is overridden by new *PLOT* data. See Sect. F11.4.9. The value of *lplot* set here is written on the restart unit and the value of *lplot* set in the *PLOT* data is not.

BUG = *ldbug* — print debug information, default = NO
Enter *YES* for code debug purposes only.

TRK = *ltrk* — print tracing information, default = NO
Enter *YES* for code debug purposes only.

PWT = *lpwt* — print weight average array, default = NO

PGM = *lgeom* — print unprocessed geometry as it is read, default = NO

SMU = *lmult* — calculate the average self-multiplication of a unit, default = NO

NUB = *newbar* — calculate the average number of neutrons per fission and the average energy group at which fission occurred, default = NO

PAX = *lcorsp* — print the arrays defining the correspondence between the cross-section energy group structure and the albedo energy group structure, default = NO

Logical Unit Numbers

XSC = *xsecs*	logical unit number for mixed cross sections, default = 14
ALB = *albdo*	logical unit number for albedo data, default = 79
WTS = *wts*	logical unit number for weights, default = 80
LIB = *lib*	logical unit number for *AMPX* working library, default = 0
SKT = *skrt*	logical unit number for scratch space, default = 16
RST = *rstrt*	logical unit number for reading restart data, default = 0 Enter a logical unit number to restart with *BEG* = 1
WRS = *wstrt*	logical unit number for writing restart data, default = 0 A non-zero value must be entered if *RES* > 0.

F11.4.4 GEOMETRY DATA

GEOMETRY REGION DATA . . . geometric arrangements in KENO are achieved in a manner similar to using a child's building blocks. Each building block is called a UNIT or BOXTYPE. An array or lattice is constructed by stacking these units or box types. Once an array or lattice has been constructed, it can be placed in a unit by using an ARRAY or CORE specification.

Each UNIT in an array or lattice has its own coordinate system; however, all coordinate systems in all units must have the same orientation. All geometry data used in a problem are correlated to the absolute coordinate system by specifying a global unit or a global array. UNITS are constructed of combinations from several allowed shapes or geometric regions; i.e., cubes, rectangular parallelepipeds, cylinders, spheres, hemispheres, and hemicylinders. These regions can be placed anywhere within a UNIT as long as they are oriented along the coordinate system of the unit and do not intersect other regions. This means, for example, that a cylinder must have its axis parallel to one of the coordinate axes, while a rectangular parallelepiped must have its faces perpendicular to a coordinate axis. The most stringent KENO geometry restriction is that none of the options allow geometry regions to intersect. Figure F11.4.1 shows some situations that aren't allowed.

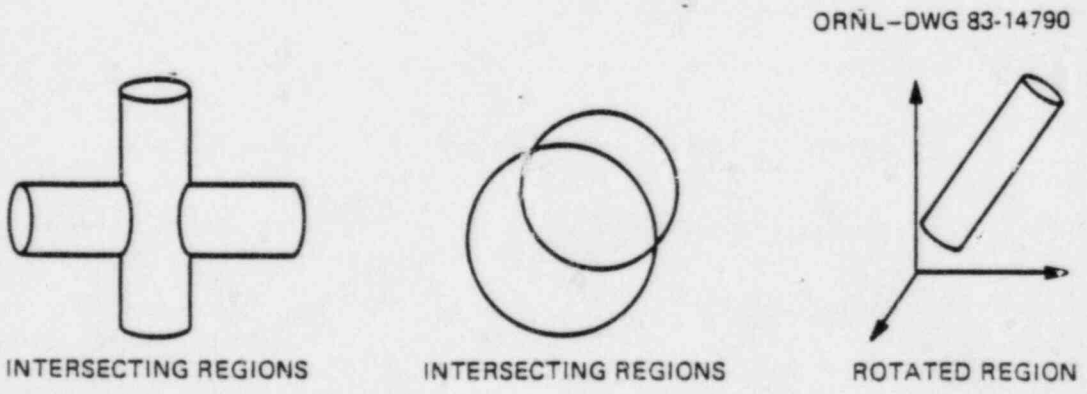


Fig. F11.4.1. Examples of geometry not allowed in KENO

Unless special options are invoked, each geometric region in a UNIT must completely enclose each interior region. However, regions can touch at points of tangency and share faces. See Fig. F11.4.2 for examples of allowable situations.

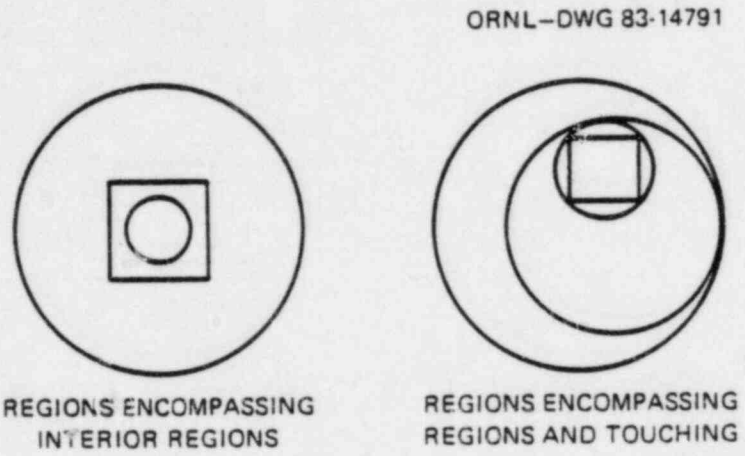


Fig. F11.4.2. Examples of correct KENO units

Special options are provided to circumvent the complete enclosure restriction. These fall under the heading EXTENDED GEOMETRY DESCRIPTIONS and include ARRAY and HOLE descriptions. The HOLE option is the simplest of these and allows placing a unit anywhere within a region, as long as intersections do not occur. See Fig. F11.4.3.

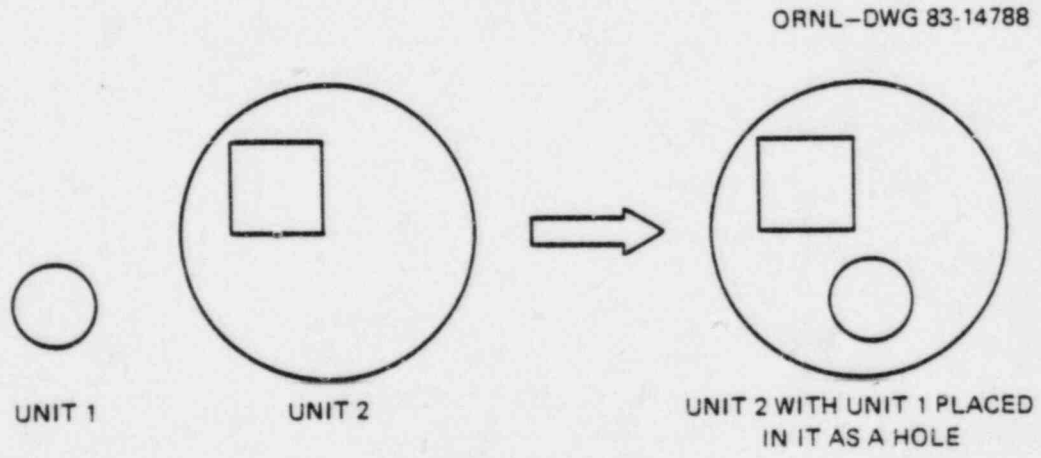


Fig. F11.4.3. Example demonstrating HOLE capability in KENO

An arbitrary number of HOLES can be placed in a region in combination with a series of surrounding regions.

Lattices or arrays are created by stacking UNITS that have a rectangular parallelepiped outer region. The adjacent faces of adjacent units stacked in this manner must match exactly. See Sect. F11.5.6.4 for additional clarification and Fig. F11.4.4 for a typical example.

ORNL–DWG 83-14789

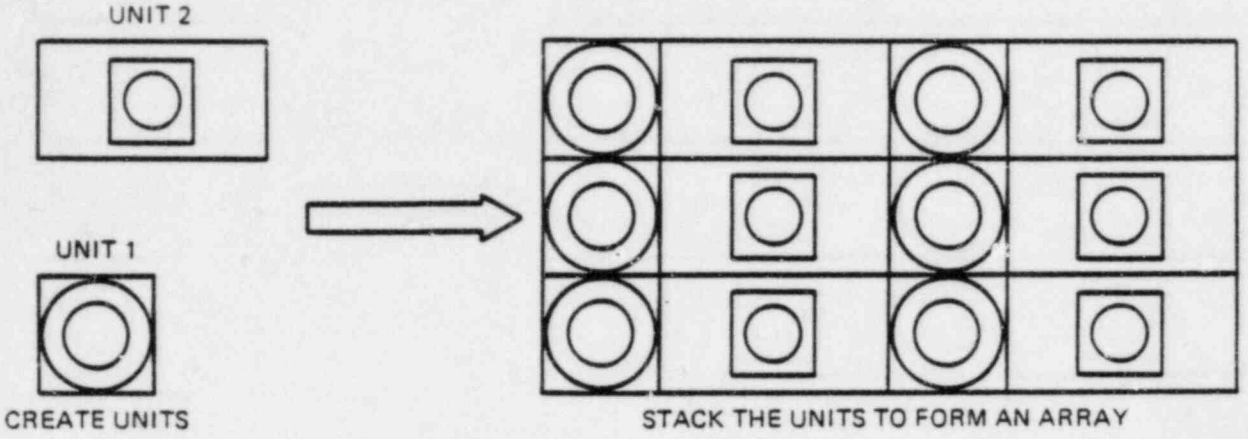


Fig. F11.4.4. Example of array construction

The ARRAY option is provided to allow placing an array or lattice within a unit. Only one array can be placed directly in a UNIT. However, multiple arrays can be placed within a unit by using HOLES. When an array is placed in a unit via a HOLE, the unit that contains the array, rather than the array itself, is placed in the unit. Arrays of dissimilar arrays can be created by stacking units that contain arrays. See Fig. F11.4.5 for an example of an array composed of units containing holes and arrays.

ORNL–DWG 83-14787

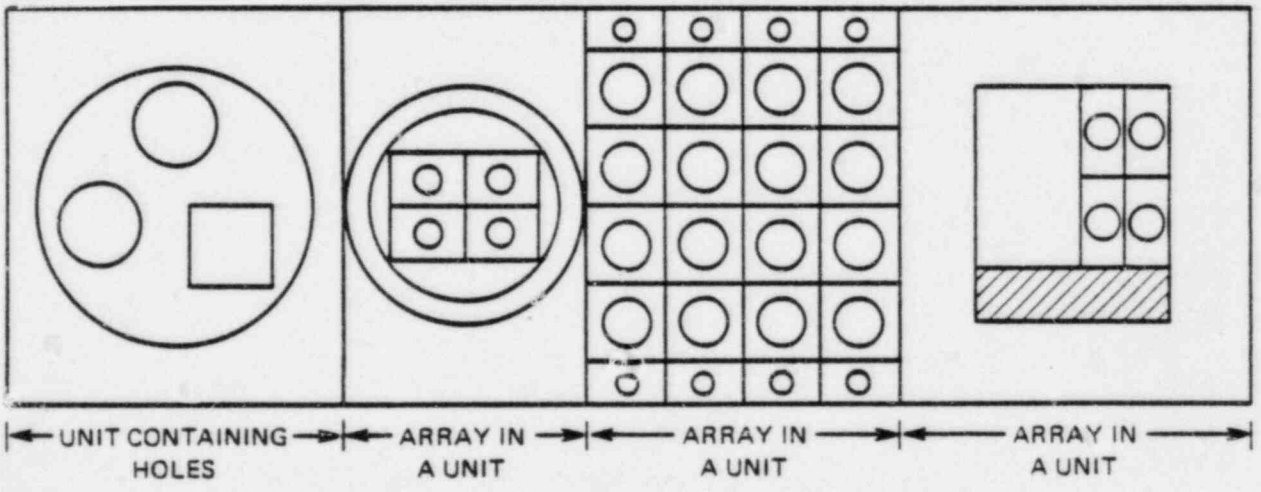


Fig. F11.4.5. Example of an array composed of units containing arrays and holes

The method of entering GEOMETRY REGION DATA follows:

READ GEOM GEOMETRY REGION DATA END GEOM

GEOMETRY REGION DATA must be entered unless the problem is being restarted. A description of all units or box types the user wishes to define must be entered. See Sect. F11.5.6 for detailed examples.

The description of a unit includes all geometry data following a UNIT INITIALIZATION. A unit is terminated by encountering another UNIT INITIALIZATION or an END GEOM.

A GEOMETRY REGION DATA description consists of one or more of the following:

(a) UNIT INITIALIZATION
(b) SIMPLE GEOMETRY REGION DESCRIPTION
(c) EXTENDED GEOMETRY DESCRIPTION
(d) OPTIONAL GEOMETRY COMMENTS

(a) *UNIT INITIALIZATION* . . . This data sequence signals the beginning of a new geometric coordinate system and assigns the unit number to the geometry regions comprising the unit.

A UNIT INITIALIZATION is invoked when one or more of the following data items are encountered:

(a1) OPTIONAL GLOBAL SPECIFICATION
fgeom
GLOBAL

The word GLOBAL, if entered, must be followed by one or more blanks and a UNIT NUMBER DEFINITION and/or an ARRAY PLACEMENT DESCRIPTION. Enter the geometry word GLOBAL immediately prior to the UNIT or ARRAY that defines the overall geometric boundaries of the problem. The code defaults the global array to the array referenced by the last ARRAY PLACEMENT DESCRIPTION that is not immediately preceded by a UNIT NUMBER DEFINITION. Otherwise, it is the largest array number specified in the array data (Sect. F11.4.5).

(a2) UNIT NUMBER DEFINITION
KEYWORD UNIT NUMBER

fgeom	nbox
UNIT	enter unit ID number
or	
BOX TYPE	enter unit ID number
or	
BOXTYPE	enter unit ID number

The UNIT NUMBER definition assigns the specified unit number to the geometry data that define the unit.

(a3) ARRAY PLACEMENT DESCRIPTION
fgeom(**)
ARRAY
or
CORE
or
COREBDY
or
COREBNDS
or
COREBOUN

**Additional data are required as described in the EXTENDED GEOMETRY DESCRIPTION.

The array placement description is used to place a specified array in a UNIT. It positions the most negative point of the array relative to the origin of the UNIT.

(b) SIMPLE GEOMETRY REGION DESCRIPTION . . . This type of geometry data consists of simple shapes whose specification is independent of other geometry regions. Free form input is used to enter the data. Options R, *, $, and P from Table F11.4.2 can be used. Each SIMPLE GEOMETRY REGION DESCRIPTION is entered in the form:

GEOMETRY WORD
fgeom

MIXTURE ID
mat

BIAS ID
imp

APPROPRIATE REGION DIMENSIONS
xx

OPTIONAL REGION DATA
origin data
and/or
chord data

The *GEOMETRY WORD fgeom* is followed by one or more blanks and must be one of the keywords below.

CUBE, CUBOID, SPHERE, CYLINDER, ZCYLINDER, XCYLINDER, YCYLINDER, HEMISPHERE, HEMISPHE+X, HEMISPHE-X, HEMISPHE+Y, HEMISPHE-Y, HEMISPHE+Z, HEMISPHE-Z, XHEMICYL+Y, XHEMICYL-Y, XHEMICYL+Z, XHEMICYL-Z, YHEMICYL+X, YHEMICYL-X, YHEMICYL+Z, YHEMICYL-Z, ZHEMICYL+X, ZHEMICYL-X, ZHEMICYL+Y, ZHEMICYL-Y

Note: *fgeom* may be no more than 12 characters long.

CUBE has +X = +Y = +Z and -X = -Y = -Z. Note that the +X dimension need not equal the -X dimension of the cube; i.e., the origin need not be at the center of the cube.

CUBOID is a rectangular parallelepiped and may be described anywhere relative to the origin.

SPHERE must be centered about the origin, unless otherwise specified by the optional region origin data.

CYLINDER has its length described along the Z axis and its center line must lie on the Z axis, unless otherwise specified by the optional region origin data.

ZCYLINDER has its length described along the Z axis and its center line must lie on the Z axis, unless otherwise specified by the optional region origin data.

XCYLINDER has its length described along the X axis and its center line must lie on the X axis, unless otherwise specified by the optional region origin data.

YCYLINDER has its length described along the Y axis and its center line must lie on the Y axis, unless otherwise specified by the optional region origin data.

HEMISPHERE is used to define a spherical segment of one base whose spherical surface exists in the positive z direction. The base or flat portion of the spherical segment is centered about a point that may be specified in the optional region data. By default, the center of the spherical surface is the origin and the distance to the base from the center of the spherical surface is zero.

HEMISPHE*bc* is used to define a spherical segment of one base whose spherical surface exists in the *bc* direction (b = + or -, c = x, y or z). The base or flat portion of the spherical segment is located a distance ρ from the center of the spherical surface, and the center may be specified in the optional region data. **HEMISPHE+Z** is the same as the previously described **HEMISPHERE** and **HEMISPHE-Z** is the mirror image of **HEMISPHE+Z**, therefore existing only in the negative Z direction. By default the center of the spherical surface is the origin and the distance of the base from the center of the spherical surface is zero.

*b***HEMICYL***cd* is used to define a cylindrical segment whose axis is in the *b* direction (b = x, y, or z) and whose cylindrical surface exists only in the *c* direction from a plane perpendicular to the *d* axis (c = + or -, d = x, y or z). The position of this plane (cut surface) can be specified in the optional region data. This plane cuts the cylinder parallel to the axis at some distance, ρ, from the axis. By default, the axis passes through the origin and ρ is zero. (Examples: **ZHEMICYL+X, YHEMICYL-Z, XHEMICYL+Y**)

The *MIXTURE ID mat* specifies the mixture that is to occupy the volume defined by the region. This is followed by one or more blanks.

The *BIAS ID imp* specifies the weights that are to be used in the volume defined by the geometry card. Default weights are used for every *imp* not specified in the *BIASING INFORMATION* data. For clarification of how to use *imp* to specify nondefault weights, see Sect. F11.4.7. The *BIAS ID* number is followed by one or more blanks.

APPROPRIATE REGION DIMENSIONS [*xx(1)* through *xx(6)*] are separated by one or more blanks and define the size of the region.

xx(1) Radius for a sphere, cylinder, hemisphere, or hemicylinder,
+X dimension for a cube or cuboid.

xx(2) -X dimension for cube or cuboid,
+Z for cylinder or Z cylinder,
+X for X cylinder,
+Y for Y cylinder,
+length for hemicylinder,
omit *xx(2)* for a sphere or hemisphere.

xx(3) +Y dimension for cuboid,
-Z for cylinder or Z cylinder,
-X for X cylinder,
-Y for Y cylinder,
-length for hemicylinder,
omit *xx(3)* for a sphere, hemisphere, or cube.

xx(4) -Y dimension for cuboid,
omit for all other geometry types.

xx(5) +Z dimension for cuboid,
omit for all other geometry types.

xx(6) -Z dimension for cuboid,
omit for all other geometry types.

Enter origin data in the form:

fgeom enter the word **ORIG** or **ORIGIN**

xx(1) the X coordinate of the origin of a sphere or hemisphere,
the X coordinate of the centerline of a Z or Y cylinder or hemicylinder,
the Y coordinate of the centerline of an X cylinder or hemicylinder.

xx(2) the Y coordinate of the origin of a sphere or hemisphere,
the Y coordinate of a Z cylinder or hemicylinder,
the Z coordinate of an X or Y cylinder or hemicylinder.

xx(3) the Z coordinate of the origin of a sphere or hemisphere; omit for all cylinders and hemicylinders.

Enter chord data in the form:

fgeom enter the word **CHORD**

xx(7) The distance, ρ, from the "cut surface" to the center of the spherical surface or axis of a hemicylinder. See (1) and (2) above and Figs. F11.4.6 and F11.4.7. For example if ρ is 5 cm for both pictures, the chord data for Fig. F11.4.6 would be **CHORD** -5.0 and the chord data for Fig. F11.4.7 would be **CHORD** 5.0. Entering a positive value with **CHORD** implies that more than half of the spherical or cylindrical body exists; entering a negative value with **CHORD** implies that less than half of the spherical or cylindrical body exists.

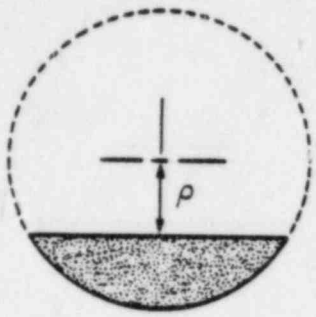


Fig. F11.4.6. Partially filled hemisphere or hemicylinder less than half full

ORNL-DWG 83-9726

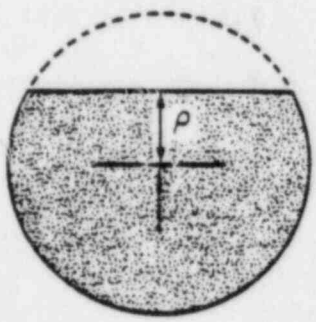


Fig. F11.4.7. Partially filled hemisphere or hemicylinder more than half full

(c) *EXTENDED GEOMETRY DESCRIPTION* . . . This type of geometry data references geometric description data from other geometry or array descriptions. Free form input is used to enter the data. Options R, *, $, and P from Table F11.4.2 can be used. Each EXTENDED GEOMETRY DESCRIPTION is entered in the form:

GEOMETRY WORD
fgeom

REFERENCE ID
mat

BIAS ID
imp

THICKNESS PER REGION
xx

ORIGIN COORDINATES
origin data

GENERATED REGIONS
nreg

The *GEOMETRY WORD fgeom* is followed by one or more blanks and must be one of the keywords below.

ARRAY, CORE, COREBDY, COREBNDS, COREBOUN, HOLE, REPLICATE, REFLECTOR

NOTE: *fgeom* may be no more than 12 characters long.

ARRAY, CORE, COREBDY, COREBNDS or **COREBOUN** are *ARRAY PLACEMENT* descriptions. They always start a new unit and generate a rectangular parallelepiped that fits the outer boundaries of the specified array. When an array placement description is the first geometry region in a unit, the specified array is positioned in the unit according to its origin coordinates. The origin coordinates specify the most negative point in the array with respect to the coordinate system of the surrounding unit.

HOLE is used to place a specified unit within the single geometry region that precedes it. The position of the **HOLE** within the region is determined by the origin coordinates.

REPLICATE is used to generate additional geometry regions having the shape of the previous region. The geometry word **REFLECTOR** is a synonym for **REPLICATE.** The desired weighting functions can be applied to those regions by specifying biasing data as described in Sect. F11.4.7. The **REPLICATE** specification includes (1) the *MIXTURE, ID, mat*, (2) the first *BIAS ID, imp*, for which these weighting data apply, (3) the thickness per region for each surface, *xx(1)...xx(6)*, as necessary to specify the desired thickness per region for each surface of the shape being generated, and (4) the number of regions to be generated, *nreg*. The total thickness generated for each surface is the thickness per region for that surface times *nreg*.

The replicate card is frequently used to generate weighting regions external to the *ARRAY PLACEMENT* description. Thus an *ARRAY PLACEMENT* description followed by a **REPLICATE** description would generate regions of a cuboidal shape. A cylindrical reflector could be generated by following the *ARRAY PLACEMENT* description with a **CYLINDER** and then a **REPLICATE**. A **HOLE** cannot immediately follow a **REPLICATE**.

Extra regions using default weights can be generated by specifying the first importance region, *imp*, to be one that was not defined in the *BIASING INFORMATION*. This capability can be used to generate extra regions for collecting information such as fluxes, leakage, etc.

Multiple replicate descriptions can be used in any problem. This can be utilized to handle different reflector materials of different thickness on different faces.

The number of appropriate region dimensions, *xx*, for specifying **REPLICATE** are determined by the preceding region. (For example, if the previous region was a sphere, one entry is required. If the previous region was a cylinder, the first entry is the thickness/region in the radial direction, the second entry is the thickness/region in the positive length direction, and the third entry is the thickness/region in the negative length direction, etc.) The replicate specification requirements for a cube are the same as for a cuboid.

REFERENCE ID mat specifies (1), (2) or (3) below

(1) *numara*, the array referenced by the *ARRAY PLACEMENT* description. (**ARRAY, CORE, COREBDY**, etc.)

(2) *lhole*, the unit that is to be placed within the preceding *SIMPLE GEOMETRY REGION* for a **HOLE** description.

(3) *mat*, the mixture that is to be used in the generated regions for a **REPLICATE** description.

BIAS ID imp

(1) omit if *fgeom* is **ARRAY**. Enter a positive number for other *ARRAY PLACEMENT* descriptions.

(2) omit if *fgeom* is **HOLE**

(3) For **REPLICATE**, *imp* is the *BIAS ID* number corresponding to the first generated region and is incremented by one for each generated region. If *imp* is negative, it is not incremented, and all the generated regions use the absolute value of *imp* as their *BIAS ID*.

APPROPRIATE REGION DIMENSIONS [*xx(1)* through *xx(6)* are separated by one or more blanks.

THICKNESS PER REGION xx

(1) omit for *ARRAY PLACEMENT* descriptions

(2) omit for **HOLE**

(3) for **REPLICATE**, [*xx(1)* through *xx(6)*] are separated by one or more blanks and define the size of the region. The thickness per region is always positive.

xx(1) First thickness per region for the generated geometry (i.e., in the radial direction for a spherical or cylindrical shape, or in the positive X direction for a cube, cuboid or core).

xx(2) Second thickness per region for the generated geometry (i.e., in the direction of positive length for a cylinder or hemicylinder or in the negative X direction for a cube, cuboid or core).
Omit for sphere or hemisphere.

xx(3) Third thickness per region for the generated geometry (i.e., in the direction of negative length for a cylinder or hemicylinder or in the positive Y direction for a cube, cuboid, or core).
Omit for sphere or hemisphere.

xx(4) Fourth thickness per region for the generated geometry (i.e., in the negative Y direction for a cube, cuboid, or core).
Omit for all other geometry types.

xx(5) Fifth thickness per region for the generated geometry (i.e., in the positive Z direction for a cube, cuboid, or core).
Omit for all other geometry types.

xx(6) Sixth thickness per region for the generated geometry (i.e., in the negative Z direction for a cube, cuboid, or core).
Omit for all other geometry types.

ORIGIN COORDINATES *xxorg* [*xxorg(1)* through *xxorg(3)* are separated by one or more blanks]

xxorg(1) (1) enter the X coordinate of the most negative point of the array with respect to the coordinate system of surrounding unit for *ARRAY PLACEMENT* descriptions

(2) enter *xhole*, the offset in the X direction of the origin of the **HOLE UNIT** with respect to the origin of the region in which it is placed

(3) omit for **REPLICATE**

xxorg(2) (1) enter the Y coordinate of the most negative point of the array with respect to the surrounding unit for *ARRAY PLACEMENT* descriptions

(2) enter *yhole*, the offset in the Y direction of the origin of the **HOLE** unit with respect to the origin of the region in which it is placed

(3) omit for **REPLICATE**

xxorg(3) (1) enter the Z coordinate of the most negative point of the array with respect to the surrounding unit for array placement descriptions

(2) enter *zhole*, the offset in the Z direction of the origin of the **HOLE** unit with respect to the origin of the region in which it is placed

(3) omit for **REPLICATE**

NUMBER OF GENERATED REGIONS nreg (1) omit for array placement descriptions

(2) omit for **HOLE**

(3) enter the number of regions to be generated for **REPLICATE**

OPTIONAL GEOMETRY COMMENTS . . . these data allow the user to enter a comment for any unit. Data are entered in the form:

COM = ***delim coment delim***

The keyword COM= signals that a comment is to be read. The first non-blank character following the keyword is the beginning delimiter, *delim*. The comment can be as long as 132 characters, including imbedded blanks. It must be terminated with the same delimiter that signaled the beginning of the comment. The optional geometry comment must follow the UNIT INITIALIZATION for a unit but can precede or follow any SIMPLE or EXTENDED GEOMETRY REGION DESCRIPTION within the unit. In the event that more than one comment is entered within a unit, the last one is used. Existing comments are printed at the beginning of each unit description in the computer printout.

F11.4.5 ARRAY DATA

ARRAY DEFINITION DATA . . . The array definition data block is used to define the size of an array and to position units (defined in the geometry data) in a three-dimensional lattice that represents the array being described. As many arrays as are necessary can be described in a problem, subject to computer storage limitations. A single unit problem is one for which an array definition data block is not entered.

An array definition data block consists of *ARRAY PARAMETERS* followed by a *UNIT ORIENTATION DESCRIPTION*. The sequence *ARRAY PARAMETERS, UNIT ORIENTATION DESCRIPTION* must be repeated for each array that is to be used in the problem. *ARRAY PARAMETERS* must be entered for all array problems and consist of parameter input defining (1) an array identification number for the lattice, (2) the number of units in each direction of the three-dimensional lattice, (3) the global array number, and (4) a comment to be printed at the beginning of the array in the printout. The global array is the array referenced by the global unit in a problem. If there is no global unit, the global array is the outermost array in the description of the problem. The array identification number, (1), and the number of units in each direction, (2), and the associated *UNIT ORIENTATION DESCRIPTION* must be entered for each array that is used in the problem. The global array number, (3), should be entered only if a global unit does not exist. If it is entered more than once, the last value is used. The optional array comment is available for the user's convenience, but is not necessary for the problem. The *UNIT ORIENTATION DESCRIPTION* consists of a *KEYWORD, UNIT ORIENTATION DATA* and a *DELIMITER* which must be entered in order. The *KEYWORD* is used to indicate the method of entering the unit orientation data. The *UNIT ORIENTATION DATA* sequence is used to position units in three-dimensional lattice and the *DELIMITER* is used to terminate the unit orientation data for the array. The *UNIT ORIENTATION DESCRIPTION* does not need to be entered for a problem in which only one box type or unit is described *unless that unit number is larger than 1. It must be entered for all problems utilizing more than one unit or box type. The adjacent faces of units in contact with each other within an array must be the same size and shape.* Multiple arrays are defined by entering the sequence *ARRAY PARAMETERS, UNIT ORIENTATION DESCRIPTION* for each array that is to be described in the problem.

Enter **ARRAY DEFINITION DATA** in the form:

READ ARRAY *ARRAY PARAMETERS UNIT ORIENTATION DESCRIPTION* **END ARRAY**

A 1x1x1 array of Unit 1 can be defined by entering only the **READ ARRAY END ARRAY.** *ARRAY PARAMETERS* define the array number and the array size. They utilize the following keywords. Enter only those whose value you wish to change.

ARRAY PARAMETERS

ARA = *numa*	array number for the array being input, default = 1
GBL = *iglobl*	array number for the global array (enter no more than once for a problem), default = largest value of *numa* for an unreflected system or the value of *numara* entered in the global unit.
NUX = *nbxmax*	number of units in the X direction of the array, default = 1
NUY = *nbymax*	number of units in the Y direction of the array, default = 1
NUZ = *nbzmax*	number of units in the Z direction of the array, default = 1
COM = *delim coment delim*	allows entering a comment that will be printed with the unit orientation data. Maximum comment length is 132 characters.

The *UNIT ORIENTATION DESCRIPTION* is composed of a *KEYWORD, ORIENTATION DATA* and a *DELIMITER* as described below:

KEYWORD

type enter the word **LOOP** or **FILL**, followed by one or more blanks.

LOOP enters unit orientation data in a manner resembling *FORTRAN DO* loops. The first field contains the unit number, followed by three fields that are treated like *FORTRAN DO* loops. The arrangement of boxes may be considered as consisting of a three-dimensional matrix of unit numbers, with the unit position increasing in the positive X, Y, and Z directions, respectively. Each set of mixed box orientation data for the **LOOP** option consists of the following parameters, separated by one or more blanks.

ORIENTATION DATA for **LOOP**

ltype The unit or box type, *ltype* must be greater than zero.

ix1 The starting position in the X direction, *ix1* must be at least 1 and less than or equal to *nbxmax* (parameter *NUX*).

ix2 The ending position in the X direction, *ix2* must be at least 1 and less than or equal to *nbxmax*.

incx The number of units by which increments are made in the positive X direction. *incx* must be greater than zero and less than or equal to *nbxmax*.

iy1 The starting position in the Y direction. *iy1* must be at least 1 and less than or equal to *nbymax* (parameter *NUY*).

iy2 The ending position in the Y direction, *iy2* must be at least 1 and less than or equal to *nbymax*.

incy The number of boxes by which increments are made in the positive Y direction, *incy* must be greater than zero and less than or equal to *nbymax*.

iz1 The starting position in the Z direction, *iz1* must be at least 1 and less than or equal to *nbzmax* (parameter *NUZ*).

iz2 The ending position in the Z direction, *iz2* must be at least 1 and less than or equal to *nbzmax*.

incz The number of boxes by which increments are made in the positive Z direction. *incz* must be greater than zero and less than or equal to *nbzmax*.

The sequence *ltype* through *incz* is repeated until the entire array is described. If any portion of an array is defined in a conflicting manner, the last entry to define that portion will determine the array's configuration. To utilize this feature, fill the entire array with the most prevalent unit type and superimpose the other unit types in their proper places. An example showing the use of the **LOOP** option is given below.

Given:

1 2 1 2 1	2 1 2 1 2	1 1 1 1 1
1 1 1 1 1	2 2 2 2 2	1 3 3 3 1
1 1 1 1 1	2 2 2 2 2	1 3 3 3 1
1 2 1 2 1	2 1 2 1 2	1 1 1 1 1
Z Layer 1	Z Layer 2	Z Layer 3

The data for this array could be entered using the following entries:

(1) 1 1 5 1 1 4 1 1 3 1 This fills the entire array with 1's.

(2) 2 2 5 2 1 4 3 1 1 1 This loads the four 2's in the first layer.

(3) 2 1 5 1 2 3 1 2 2 1 This loads the second and third rows of 2's in the second layer.

(4) 2 1 5 2 1 4 3 2 2 1 This loads the desired 2's in the first and fourth rows of the second layer.

(5) 3 2 4 1 2 3 1 3 3 1 This loads the 3's in the third layer and completes the data input for the array.

The second layer could have been defined by substituting the following data for entries (3) and (4).

(3) 2 1 5 1 1 4 1 2 2 1 This completely fills the second layer with 2's.

(4) 1 2 4 2 1 4 3 2 2 1 This loads the four 1's in the second layer.

When using the **LOOP** option, there is no single correct method of entering the data. If a unit is improperly positioned in the array or if some positions in the array are left undefined, it is often easier to add additional data to correctly define it than to try to correct the existing data.

FILL enters data by stringing in unit numbers starting at X=1, Y=1, Z=1, and varying X, then Y, and then Z to fill the array. *nbxmax* x *nbymax* x *nbzmax* entries are required. FIDO-like input options are also available for filling the array.

ORIENTATION DATA for **FILL**

The **FILL** option consists of entering a unit number for every position in the array by using the FIDO-like input options specified in TableF11.4.2. The orientation data for the **FILL** option may be terminated with a T.

DELIMITER

Enter the word **END** followed by a blank and the previously entered *KEYWORD* (i.e., enter **END FILL** or **END LOOP**). The delimiter need not be entered if only one set of *ARRAY DEFINITION DATA* is to be read.

Table F11.4.2. FIDO-like input for mixed box orientation fill option

Count Field	Option Field	Operand Field	Function
		j	stores *j* at the current position in the array
i	R	*j*	stores *j* in the next *i* positions in the array
i	*	*j*	stores *j* in the next *i* positions in the array
i	S	*j*	stores *j* in the next *i* positions in the array
i	P	*j*	alternately stores *j* and *-j* in the next *i* positions of the array
	F	*j*	fills the remainder of the array with unit number *j*, starting with the current position in the array
	A	*j*	sets the current position in the array to *j*
i	S		increments the current position in the array by *i* (This allows skipping *i* positions. *i* may be positive or negative.)
i	Q	*j*	repeats the previous *j* entries *i* times. The default value of *i* is 1.
i	N	*j*	repeats the previous *j* entries *i* times, inverting the sequence each time. The default value of *i* is 1.
i	B	*j*	back i entries. From that position, repeat the previous j entries in reverse order. See Example 4. The default value of i is 1.
i	I	*j k*	provides *i* entries linearly interpolated between *j* and *k*, and the end points (i.e., a total of *i+2* points) *j* and *k* must be separated by at least one blank.
i	L	*j k*	provides *i* entries logarithmically interpolated between *j* and *k*, and the end points (i.e., a total of *i+2* points) *j* and *k* must be separated by at least one blank.
	T		terminates the data reading for the array

When entering data utilizing the options in this table, the *count field* and *option field must be adjacent with no imbedded* blanks. The operand field may be separated from the option field by one or more blanks.

To illustrate the use of the options available in Table F11.4.2, consider the following examples. The positions in an array are numbered sequentially from left to right, bottom to top. A 3x3x1 array is numbered as shown below.

7 8 9

4 5 6

1 2 3

EXAMPLE 1. Consider a 3x3x1 array filled as shown below.

1 1 1

1 2 1

1 1 1

The input data to describe this array could be entered as follows:

(1) 1 1 1 1 2 1 1 1 1 T — This fills the array, one position at a time, starting at the lower left corner. The T terminates the data.

(2) F1 A5 2 T — The F1 fills the entire array with 1's, the A5 locates the fifth position in the array, and the 2 loads a 2 in that position. The T terminates the data.

(3) F1 A1 4S 2 T — The F1 fills the entire array with 1's, the A1 locates the first position in the array (lower left corner), the S4 skips over the next four positions in the array, and the 2 loads a 2 in the next position. The T terminates the data.

(4) 4R1 2 4R1 T — The first 4R1 loads 1's in the first four positions of the array, a 2 is loaded in the next (fifth) position of the array, and the last 4R1 loads 1's in the next four positions of the array. The T terminates the data.

(5) 4*1 2 4$1 T — The 4*1 loads 1's in the first four positions of the array. A 2 is loaded in the next position of the array, and the 4$1 loads 1's in the next four positions of the array. The T terminates the data.

EXAMPLE 2. Consider a 4x3x1 array filled as shown below.

1 2 2 1

1 2 2 1

1 2 2 1

The input data to describe this array could be entered as follows.

(1) 1 2 2 1 1 2 2 1 1 2 2 1 T

This fills the array, one position at a time, starting at the lower left corner. The T terminates the data.

(2) 1 2R2 2R1 Q4 2R2 1 T

A 1 is loaded in the first position of the array. The 2R2 loads 2's in the next two positions of the array (positions two and three). The 2R1 loads 1's in the next two positions of the array (positions four and five). The Q4 loads the 2 2 1 1 from positions two through five in positions six through nine. The last 2R2 loads 2's in positions ten and eleven of the array. The last 1 loads a 1 in the next position (position twelve). The T terminates the data.

(3) 1 2 5N2 T

The 1 is entered in the first position of the array and the 2 is entered in the second position. The 5N2 causes the previous two entries to be repeated five times, reversing their order each time. The T terminates the data.

(4) 1 2R2 1 2Q4 T

The first 1 is loaded in the first position of the array. The 2's are loaded in positions two and three of the array. Then a 1 is loaded in the fourth position of the array. This describes the entire bottom row of the array. The 2Q4 then repeats the previous four entries (those loaded in positions one through four of the array) two times. Thus the first row is repeated twice, filling the remainder of the array.

EXAMPLE 3. Consider a 6x3x1 array as shown below.

1 2 2 1 1 2

1 2 2 1 1 2

1 2 2 1 1 2

A simple input description for this array is:

(1) 1 2 2N2 2Q6 T

The first position of the array is filled with a 1. The second position of the array is filled with a 2. The 2N2 causes the previous two entries to be repeated two times, reversing their order each time. This completes loading the bottom row of the array. The 2Q6 repeats the previous six entries twice, completing the second and third rows of the array. The T terminates the data.

EXAMPLE 4. Consider a 7x3x1 array as shown below.

1 2 3 4 3 2 1

2 3 4 5 4 3 2

1 2 3 4 3 2 1

The input data to describe this array could be entered as follows:

(1) 1 2 3 4 1B3 2 3 4 5 1B3 1 2 3 4 1B3 T	The 1 2 3 and 4 are loaded in the first four positions of the array. The 1B3 steps back over the 4 and repeats the 1 2 3 sequence in reverse order (i.e., 3 2 1). This yields the 1 2 3 4 3 2 1 in the first row of the array. The same procedure applies to the next two rows.
(2) 1 2 3 4 1B3 2 3 4 5 1B10 T	The 1 2 3 4 1B3 yields the first row as explained above. The 2 3 4 5 enters the 2 3 4 5 at the left of the second row. The 1B10 enters the 4 3 2 at the right of the second row and the entire third row.

F11.4.6 ALBEDO DATA

ALBEDO DATA . . . Albedo boundary conditions are entered using a *FACE CODE* to define **where** albedo conditions are to be used, and an *ALBEDO NAME* to indicate **which** albedo condition is to be used on that face. The **default value for each face is vacuum.** The default values are overridden only on faces for which other albedo names are specified. Albedo boundary conditions are applied **only** to **the outermost region of a problem.** This geometry region must be a rectangular parallelepiped.

Enter **ALBEDO DATA** in the form:

READ BOUNDS

FACE CODE
face

ALBEDO NAME
aname

END BOUNDS

The sequence *FACE CODE, ALBEDO NAME* is entered as many times as necessary to define the appropriate albedo boundary conditions. If multiple entries are made for a face, the *ALBEDO NAME* associated with the last *FACE CODE* specifying that face is used.

The *FACE CODES* are described in Table F11.4.3 and the *ALBEDO NAMES* are given in Table F11.4.4.

Table F11.4.3. Face codes for entering boundary (albedo) conditions

Face Code	Faces that are defined by the Face Codes
+XB=	positive X face
&XB=	positive X face
-XB=	negative X face
+YB=	positive Y face
&YB=	positive Y face
-YB=	negative Y face
+ZB=	positive Z face
&ZB=	positive Z face
-ZB=	negative Z face
ALL=	all 6 faces
XFC=	both positive and negative X faces
YFC=	both positive and negative Y faces
ZFC=	both positive and negative Z faces
+FC=	positive X, Y, and Z faces
&FC=	positive X, Y, and Z faces
-FC=	negative X, Y, and Z faces
XYF=	positive and negative X and Y faces
XZF=	positive and negative X and Z faces
YZF=	positive and negative Y and Z faces
+XY=	positive X and Y faces
+YX=	positive X and Y faces
&XY=	positive X and Y faces
&YX=	positive X and Y faces
+XZ=	positive X and Z faces
+ZX=	positive X and Z faces
&XZ=	positive X and Z faces
&ZX=	positive X and Z faces
+YZ=	positive Y and Z faces
+ZY=	positive Y and Z faces
&YZ=	positive Y and Z faces
&ZY=	positive Y and Z faces
-XY=	negative X and Y faces
-XZ=	negative X and Z faces
-YZ=	negative Y and Z faces
YXF=	positive and negative X and Y faces
ZXF=	positive and negative X and Z faces
ZYF=	positive and negative Y and Z faces
-YX=	negative X and Y faces
-ZX=	negative X and Z faces
-ZY=	negative Y and Z faces

Table F11.4.4. Albedo names available on the KENO V albedo library for use with the face codes

Albedo Name	Albedo Description
DP0H2O DPOH2O DP0 DPO	12-in. (30.48-cm) double P_0 water differential albedo with 4 incident angles
H2O WATER	12-in. (30.48-cm) water differential albedo with 4 incident angles
PARAFFIN PARA WAX	12-in. (30.48-cm) paraffin differential albedo with 4 incident angles
CARBON GRAPHITE C	78.74-in. (200.00-cm) carbon differential albedo with 4 incident angles
ETHYLENE POLY CH2	12-in. (30.48-cm) polyethylene differential albedo with 4 incident angles
CONC-4 CON4 CONC4	4-in. (10.16-cm) concrete differential albedo with 4 incident angles
CONC-8 CON8 CONC8	8-in. (20.32-cm) concrete differential albedo with 4 incident angles
CONC-12 CON12 CONC12	12-in. (30.48-cm) concrete differential albedo with 4 incident angles
CONC-16 CON16 CONC16	16-in. (40.64-cm) concrete differential albedo with 4 incident angles
CONC-24 CON24 CONC24	24-in. (60.96-cm) concrete differential albedo with 4 incident angles
VACUUM VOID VACU VAC	vacuum condition
SPECULAR MIRROR MIRR SPEC SPE MIR REFL REFLECT	mirror image reflection
PERIODIC PERI PER	periodic boundary condition

F11.4.7 BIASING OR WEIGHTING DATA

BIASING INFORMATION . . . The average weight of a neutron that survives Russian roulette, *wtavg*, is defaulted to *dwtav* (**WTA=** in the parameter data) for all *BIAS ID's* and can be overridden by entering biasing information.

The biasing information is used to relate a *BIAS ID* to the desired energy-dependent values of *wtavg*. This concept is similar to the way the *MIXTURE ID, mat* is related to the macroscopic cross-section data.

The weighting functions used in KENO V are energy-dependent values of *wtavg* that are applicable over a given thickness interval of a material. For example, the weighting function for water[2] is composed of sets of energy-dependent values of *wtavg* for 11 intervals, each interval being 3-cm thick. The first set of *wtavg's* is for the 0 to 3-cm interval of water, the second set of *wtavg's* is for the 3 to 6-cm interval of water, etc. The eleventh set of *wtavg's* is for the 30 to 33-cm interval of water.

To input biasing information, a *BIAS ID* must be assigned to correspond to a set of *wtavg*. Biasing data can specify a *MATERIAL ID* from the existing KENO V weight library or from the *AUXILIARY DATA* input. The materials available from the KENO V weights library are listed in Table F11.4.5.

BIASING INFORMATION is entered in the following form:

READ BIAS KEYWORD CORRELATION DATA AUXILIARY DATA END DATA

KEYWORD
enter **ID=**, **WT=** or **WTS=**

ID= specifies that *CORRELATION DATA* will be entered next.

WT= or **WTS=** specifies that *AUXILIARY DATA* will be entered next.

CORRELATION DATA are used to correlate a set of *wtavg* to a BIAS ID, *imp*, as specified in the geometry data. This causes the specified *wtavg* to be used as the weighting function in the volume defined by that geometry region.
CORRELATION DATA must be entered in the order shown.

id enter the identification (MATERIAL ID) for the material whose weighting function is to be used. A material ID can be taken from the existing KENO V weights library (Table F11.4.5) or from the *AUXILIARY DATA* as described later. If a material ID appears in both the KENO V weights library and the *AUXILIARY DATA*, the *wtavg* from the auxiliary data will be used.

ibgn is the BIAS ID of the weighting function for the first interval of material *id*. The geometry card having *imp* = *ibgn* will use the group-dependent *wtavg's* from the first interval of material *id*.

iend is the BIAS ID of the group-dependent *wtavg's* from the (*iend* - *ibgn* +1)th interval of material *id*.

AUXILIARY DATA are used to enter biasing or weighting information from cards. It can be used to supply biasing information for materials not found in the KENO V weights library or to override the *wtavgs* from that library.
AUXILIARY DATA must be entered in the order shown.

wttitl enter an arbitrary 12 character title name such as CONCRETE, WATER, SPECIALH2O, etc., to identify the material for which you are entering data.

id enter an identification number (MATERIAL ID). The value is arbitrary. However, if the data are to be utilized in the problem, this ID must also be used at least once in the *CORRELATION DATA*.

nsets enter the number of sets of group structures for which *wtavg* will be read for this id.

The sequence *thkinc, numinc, ngpwt, wtavg*, described below, is repeated *nsets* times.

thkinc enter the thickness of each increment for which *wtavg* will be read for this id.

numinc enter the number of increments for which *wtavg* will be read for this id.

ngpwt enter the number of energy groups for this set of *wtavg*.

wtavg enter *numinc* x *ngpwt* values of *wtavg*. For each value of *numinc*, *ngpwt* values of *wtavg* must be supplied.

CAUTIONS:

1. Each set of auxiliary or correlation data must be completely described in conjunction with its keyword. Complete sets of these data can be interspersed in an arbitrary order but data within each set must be entered in the specified order.

2. Auxiliary data: If the same *id* is specified in more than one set of data, the last set having the group structure used in the problem is the set that will be utilized.

3. Correlation data: If biasing data define the same bias ID (*imp*, from the geometry data) more than once, the value that is entered last supersedes previous entries. *Be well aware that multiple definitions for the same bias ID can cause erroneous answers due to overbiasing.* Error messages K5-125 and K5-128 may be printed.

An example of multiple definitions for the same bias ID is given below:

READ BIAS ID=400 2 7 ID=500 5 7 END BIAS .

The data for paraffin (ID=400) will be used for bias ID's 2, 3, and 4, and the data for water (ID=500) will be used for bias ID's 5, 6, and 7.

Multiple definitions for the same bias ID are not necessarily incorrect. However, the user should be cautious about doing it and assure that the desired biasing or weighting functions are utilized in the desired geometry regions.

Table F11.4.5. ID's, group structure and incremental thicknesses for weighting data available on the weighting library

Material	Material ID	Group Structure For Which Weights Are Available	Increment* Thickness (cm)	Total Number of Increments Available
Concrete	301	16	5	20
		27	5	20
		123	5	20
Paraffin	400	16	3	10
		27	3	11
		123	3	10
Water	500	16	3	10
		27	3	11
		123	3	10
		218	3	11
Graphite	6100	16	20	6
		27	20	10
		123	20	6

*Group-dependent weight averages are supplied for each increment of the specified incremental thickness, i.e., for any given material, the first *ngp* (number of energy groups) weights apply to the first increment of the thickness specified in Table F11.4.4, the next *ngp* weights apply to the next increment of that thickness, etc. CAUTION: If bias id's defined in the weighting information data are used in the geometry, the region thickness should be consistent with the incremental thickness of the weighting data in order to avoid over biasing or under biasing.

An example of how the *BIAS ID* relates to the energy-dependent values of *wtavg* is given below.

Assume that a paraffin reflector is to be used and it is desirable to use the weighting function from the KENO V weighting library to minimize the running time for the problem. Also assume that these weighting functions are to be used in the volumes defined in the geometry cards having *imp*= 6, 7, 8, and 9. *CORRELATION DATA* are then entered.

KEYWORD is ID=
id is 400, the ID for paraffin
ibgn is 6, the first *imp* that uses the weighting function
iend is 9, the last *imp* that uses the weighting function

and *AUXILIARY DATA* will not be entered. The biasing data would be: READ BIAS ID=400 6 9 END BIAS

The results of these data are:

(1) the group-dependent *wtavg* for the 0 to 3-cm interval of paraffin will be used in the volume defined by the geometry card having *imp*= 6.

(2) the group-dependent *wtavg* for the 3 to 6-cm interval of paraffin will be used in the volume defined by the geometry card having *imp*= 7.

(3) *wtavg* for the 6 to 9-cm interval of paraffin will be used where *imp*= 8.

(4) *wtavg* for the 9 to 12-cm interval of paraffin will be used where *imp*= 9.

F11.4.8 START DATA

START DATA . . . Special start options are available for controlling the initial neutron distribution. The default starting distribution for an array is flat over the overall array dimensions, in fissile material only. The default starting distribution for a single unit is flat over the system, in fissile material only. See Table F11.4.6 for the starting distributions available in KENO V.

READ START

The starting information that can be entered is given below. Enter only the data necessary to describe the desired starting distribution.

NST= *ntypst* start type, default = 0
See Table F11.4.6 for available options.

TFX= *tfx* the x coordinate of the point at which neutrons are to be started, default= 0.0
Use for start types 3, 4, and 6.

TFY= *tfy* the y coordinate of the point at which neutrons are to be started, default= 0.0
Use for start types 3, 4, and 6.

TFZ= *tfz* the z coordinate of the point at which neutrons are to be started, default= 0.0
Use for start types 3, 4, and 6.

Table F11.4.6. Starting distributions available in KENO V

Start Type	Required Data	Optional Data	Starting Distribution
0	None	NST XSM XSP YSM YSP ZSM ZSP RFL PSP	Uniform throughout fissile material within the volume defined by (1) the outer region of a single unit, (2) the outer region of a reflected array having the reflector key set true, (3) the core boundary of the global array, or (4) a cuboid specified by XSM, XSP, YSM, YSP, ZSM, and ZSP.
1	NST	XSM XSP YSM YSP ZSM ZSP RFL PSP	The starting points are chosen according to a cosine distribution throughout the volume of a cuboid defined XSM, XSP, YSM, YSP, ZSM, and ZSP. Points that are not in fissile material are discarded.
2	NST NXS NYS NZS FCT	XSM XSP YSM YSP ZSM ZSP RFL PSP	An arbitrary fraction (FCT) of neutrons are started uniformly in the unit located at position NXS, NYS, NZS in the global array. The remainder of the neutrons are started in fissile material, from points chosen from a cosine distribution throughout the volume of a cuboid defined by XSM, XSP, YSM, YSP, ZSM, ZSP.
3	NST TFX TFY TFZ NXS NYS NZS	KFS PSP	All neutrons are started at position TFX, TFY, TFZ within the unit located at position NXS, NYS, NZS in the global array.
4	NST TFX TFY TFZ NBX	KFS PSP	All neutrons are started at position TFX, TFY, TFZ within units NBX in the global array.
5	NST NBX	PSP	Neutrons are started uniformly in fissile material in units NBX in the global array.
6	NST TFX TFY TFZ LNU*	NXS NYS NZS KFS PS6 PSP	The starting distribution is arbitrarily input. LNU is the final neutron to be started at a point TFX, TFY, TFZ relative to the global coordinate system or at a point TFX, TFY, TFZ, relative to the unit located at the global array position NXS, NYS, NZS.

*When entering data for start 6, *LNU must be the last entry for each set of data and the LNU in each successive set of data must be larger than the last. A set of data consists of required and optional data. The last LNU entered should be equal to the number per generation (parameter NPG in the parameter input).*

NXS= *nbxs* the x index of the unit's position in the global array, default = 0
Use for start types 2, 3, and 6.

NYS= *nbys* the y index of the unit's position in the global array, default = 0
Use for start types 2, 3, and 6.

NZS= *nbzs* the z index of the unit's position in the global array, default = 0
Use for start types 2, 3, and 6.

KFS= *kfis* the mixture whose fission spectrum is to be used for starting neutrons that are not in a fissionable medium. Defaulted to the first fissionable mixture. Available for start types 3, 4, and 6.

LNU= *lfin* the final neutron to be started at a point. Default = 0. Each *lfin* should be greater than zero and less than or equal to NPG.
Use for start type 6.

NBX= *nboxst* the unit or box type in which neutrons will be started. Default = 0
Use for start types 4 and 5.

FCT= *fract* the fraction of neutrons that will be started as a spike. Default = 0
Use for start type 2.

XSM= *xsm* the -X dimension of the cuboid in which the neutron will be started. For an array problem, XSM is defaulted to the minimum X coordinate of the global array. If the reflector key RFL is YES, and the outer reflector region is a cube or cuboid, XSM is defaulted to the minimum X coordinate of the outer reflector region. If RFL is YES and the outer region of the reflector is not a cube or cuboid, XSM must be entered in the data and must fit inside the outer reflector region. Available for start types 0, 1, and 2.

XSP= *xsp* the +X dimension of the cuboid in which the neutrons will be started. For an array problem XSP is defaulted to the maximum X coordinate of the global array. If the reflector key RFL is YES, and the outer reflector region is a cube or cuboid, XSP is defaulted to the maximum X coordinate of the outer reflector region. If RFL is YES and the outer region of the reflector is not a cube or cuboid, XSP must be entered in the data and must fit inside the outer reflector region. Available for start types 0, 1, and 2.

YSM= *ysm* the -Y dimension of the cuboid in which the neutron will be started. For an array problem YSM is defaulted to the minimum Y coordinate of the global array. If the reflector key RFL is YES, YSM is defaulted to the minimum Y coordinate of the outer reflector region, provided that region is a cube or cuboid. If RFL is YES and the outer region of the reflector is not a cube or cuboid, YSM must be entered in the data and must fit inside the outer reflector region. Available for start types 0, 1, and 2.

YSP= *ysp* the +Y dimension of the cuboid in which the neutrons will be started. For an array problem YSP is defaulted to the maximum Y coordinate of the global array. If the reflector key RFL is YES, YSP is defaulted to the maximum Y coordinate of the outer reflector region, provided that region is a cube or cuboid. If RFL is YES and the outer region of the reflector is not a cube or cuboid, YSP must be entered in the data and must fit inside the outer reflector region.
Available for start types 0, 1, and 2.

ZSM= *zsm* the -Z dimension of the cuboid in which the neutrons will be started. For an array problem ZSM is defaulted to the minimum Z coordinate of the global array. If the reflector key RFL is YES, ZSM is defaulted to the minimum Z coordinate of the outer reflector region, provided that region is a cube or cuboid. If RFL is YES and the outer region of the reflector is not a cube or cuboid, ZSM must be entered in the data and must fit inside the outer reflector region.
Available for start types 0, 1, and 2.

ZSP= *zsp* the +Z dimension of the cuboid in which the neutrons will be started. For an array problem ZSP is defaulted to the maximum Z coordinate of the global array. If the reflector key RFL is YES, ZSP is defaulted to the maximum Z coordinate of the outer reflector region, provided that region is a cube or cuboid. If RFL is YES and the outer region of the reflector is not a cube or cuboid, ZSP must be entered in the data and must fit inside the outer reflector region.
Available for start types 0, 1, and 2.

RFL= *rflkey* the reflector key. If the reflector key is YES, neutrons can be started in the reflector. If it is NO, all the neutrons will be started in the array. Enter YES or NO. Default = NO.
Available for start types 0, 1, and 2.

PS6= *lprt6* the key for printing start type 6 input data. If the key is YES, start type 6 data is printed. If it is NO, start type 6 data is not printed. Enter YES or NO. Default = NO.
Available for start type 6.

PSP= *lpstp* the key for printing the neutron starting points using the tracking format. If the key is YES, print the neutron starting points. If it is NO, do not print the starting points. Enter YES or NO. Default = NO.
Available for all start types.

END START

F11.4.9 EXTRA 1-D XSECS ID'S DATA

EXTRA 1-D CROSS SECTION ID'S . . . Extra 1-D cross-section ID's need not be entered. They are allowed as input in order to simplify future modifications to calculate reaction rates, etc.

READ X1DS *EXTRA 1-D CROSS SECTION ID'S* **END X1DS**

EXTRA 1-D CROSS SECTION ID'S

Enter a 1-D identification number for each extra 1-D cross section to be used. These cross sections must be available on the mixture cross-section library. X1D entries are expected to be read (see integer PARAMETER data).

F11.4.10 MIXING TABLE DATA

CROSS SECTION MIXING TABLE . . . A cross-section mixing table must be entered if KENO V is being run "stand alone" and a premixed ICE tape is not being used. A cross-section mixing table is entered in the form:

READ MIXT *XSEC PARAMETERS MIXING TABLE* **END MIXT**

The *XSEC PARAMETERS* include the number of scattering angles and the cross-section message cutoff value.

The number of scattering angles specifies the number of discrete scattering angles to be used for the cross sections. It needs to be entered only once for a problem. If more than one value is entered, the last one is used for the problem. For assistance in determining the number of discrete scattering angles for the cross sections, see Sect. F11.5.4.3.

SCT= *nsct* where *nsct* is the number of discrete scattering angles, default = 1

The cross-section message cutoff value is the value of the P_0 cross section for each energy transfer above which cross-section processing warning messages will be printed. The primary purpose of entering this cutoff value is to suppress printing these messages when they are generated during cross-section processing. For assistance in determining a value for EPS, see Sect. F11.5.4.7.

EPS= *pbxs* where *pbxs* is the value of the P_0 cross-section for each transfer, above which generated warning messages will be printed, default = 3×10^{-5}

The *MIXING TABLE* is used to specify each mixture and the nuclide ID's and number densities used in the mixtures. It consists of (a) a *MIXTURE ID* and a set of (b) *NUCLIDE ID's* and (c) *NUMBER DENSITIES* (atoms/barn-cm).

(a) *MIXTURE ID* **MIX** = *mix* where mix defines the mixture being described

(b) *NUCLIDE ID* *nucl* enter the nuclide ID number from the AMPX working library

(c) *NUMBER DENSITY* *dens* enter the number density associated with nuclide ID number *nucl*

REPEAT the sequence (b) (c) until the mixture has been completely described.

REPEAT the sequence (a) (b) (c) until all the mixtures have been described.

NOTE: If a given nuclide ID is entered more than once in the same mixture, the number densities for that nuclide are effectively summed.

If a mixture number is used as a nuclide ID, it is treated as a nuclide and the number density associated with it is really a density modifier. (If the density is entered as 1, the mixture is mixed in at full density. If it is entered as 0.5, the mixture is mixed in at 1/2 its full density.) A Monte Carlo formatted cross-section library is generated on the unit defined by the parameter **XSC=**. If this data set is saved, subsequent cases can utilize these mixtures without remixing.

F11.4.11 PLOT DATA

PLOT or **PICTURE DATA** . . . Printer plots or pictures of slices through the geometry can be printed showing mixtures number, unit numbers, or bias ID numbers. The *PLOT DATA* can include the data for any or all types of pictures. A picture by mixture number is the default kind of picture. The kind of picture is defined by the parameter **PIC=**. Printer plots are printed after the volumes are printed and before the final preparations for tracking are completed. *PLOT DATA* is not required for a problem but can be used to verify the problem description. The actual printing of the plot or picture can be suppressed by entering **PLT= NO** in the parameter data or picture data. This allows picture data to be kept in the problem data for reference purposes without actually printing the picture(s). Entering a value for **PLT** in the picture data will override any value entered in the parameter data. However, if a problem is restarted, the value of **PLT** from the parameter data is used.

Enter **PLOT DATA** in the form:

READ PLOT *PICTURE PARAMETERS* **END PLOT**

PICTURE PARAMETERS are entered using keywords followed by the appropriate data. The plot title and the plot character string must be contained within delimiters. As many picture parameters as are necessary to describe the plot should be entered. The parameter input for each plot is terminated by an **END**. Data for multiple plots are separated by the terminator **END**.

TTL= *delim ptitl delim* enter a one-character delimiter *delim* to signal the beginning of the title (132 characters maximum). The title is terminated when *delim* is encountered the second time.
default = title of the KENO case

PIC= *wrd* The picture type, *wrd*, is followed by one or more blanks and must be one of the keywords listed below. Initialized to MAT; default = value from previous plot.

MAT
MIX
MIXT
MIXTURE
MEDI
MEDIA

These keywords will cause the printer plot to represent the mixtures of mixture numbers in the specified geometry slice.

BOX
BOXT
BOXTYPE
UNT
UNIT
UNITTYPE

These keywords will cause the printer plot to represent the units or box types in the specified geometry slice.

IMP
BIAS
BIASID
WTS
WEIG
WEIGHTS
WGT
WGTS

These keywords will cause the printer plot to represent the Bias ID numbers used in the specified geometry slice.

Upper left coordinates — enter the X, Y and Z coordinates of the upper left-hand corner of the plot. **Data must be entered for all three coordinates unless all three values from the previous plot are to be used.**

XUL= *fnam(1)* — enter the X coordinate of the upper left-hand corner of the plot.
default = value from previous plot; initialized to zero if any other coordinates are entered

YUL= *fnam(2)* — enter the Y coordinate of the upper left-hand corner of the plot.
default = value from previous plot; initialized to zero if any other coordinates are entered

ZUL= *fnam(3)* — enter the Z coordinate of the upper left-hand corner of the plot.
default = value from previous plot; initialized to zero if any other coordinates are entered

Lower right coordinates — enter the X, Y, and Z coordinates of the lower right-hand corner of the plot. **Data must be entered for all three coordinates unless all three values from the previous plot are to be used.**

XLR= *fnam(4)* — enter the X coordinate of the lower right-hand corner of the plot.
default = value from previous plot; initialized to zero if any other coordinates are entered

YLR= *fnam(5)* enter the Y coordinate of the lower right-hand corner of the plot.
default = value from previous plot; initialized to zero if any other coordinates are entered

ZLR= *fnam(6)* enter the Z coordinate of the lower right-hand corner of the plot.
default = value from previous plot; initialized to zero if any other coordinates are entered

Direction cosines across the page — enter direction numbers proportional to the direction cosines for the AX axis of the plot. The AX axis is from left to right across the page. If any one of the AX direction cosines are entered, the other two are set to zero. The direction cosines are normalized by the code.

UAX= *fnam(7)* enter the X component of the direction cosines for the AX axis of the plot.
default = value from previous plot; initialized to zero if any other direction cosines are entered

VAX= *fnam(8)* enter the Y component of the direction cosines for the AX axis of the plot.
default = value from previous plot; initialized to zero if any other direction cosines are entered

WAX= *fnam(9)* enter the Z component of the direction cosines for the AX axis of the plot.
default = value from previous plot; initialized to zero if any other direction cosines are entered

Direction cosines down the page — enter direction numbers proportional to the direction cosines for the DN axis of the plot. The DN axis is from top to bottom down the page. If any one of the DN direction cosines are entered, the other two are set to zero. The direction cosines are normalized by the code.

UDN= *fnam(10)* enter the X component of the direction cosines for the DN axis of the plot.
default = value from previous plot; initialized to zero if any other direction cosines are entered

VDN= *fnam(11)* enter the Y component of the direction cosines for the DN axis of the plot.
default = value from previous plot; initialized to zero if any other direction cosines are entered

WDN=*fnam(12)* enter the Z component of the direction cosines for the DN axis of the plot.
default = value from previous plot; initialized to zero if any other direction cosines are entered

DLX= *fnam(13)* horizontal spacing between points on the plot.
default = value from previous plot; initialized to zero if NAX or NDN is entered

DLD= *fnam(14)* vertical spacing between points on the plot.
default = value from previous plot; initialized to zero if NAX or NDN is entered

NOTE: If either DLX or DLD is entered, the code will calculate the value of the other. If both are entered, the plot may be distorted.

NAX= *inam(15)* number of intervals to be printed across the page.
default = value from previous plot; initialized to zero if DLX or DLD is entered

NDN= *inam(16)* number of intervals to be printed down the page.
default = value from previous plot; initialized to zero if DLX or DLD is entered

NOTE: If either **NAX** or **NDN** is entered, the code will calculate the value of the other. If both are entered, the plot may be distorted. If the coordinates of the lower right-hand corner have been entered, only one item of (**DLX, DLD, NAX, NDN**) need be entered. If the coordinates have not been entered, it is necessary to enter both **NAX** and **NDN** in addition to either **DLX** or **DLD**. If the coordinates and **NAX** or **NDN** are entered, then **DLX** and/or **DLD** will be recomputed.

LPI= *inam(17)* number of lines per inch that will be printed down a page
default = 8

NCH= *delim CHAR delim* enter a one-character delimiter to signal the beginning of a character string, *CHAR*. The character string is terminated when *delim* is encountered the second time. *CHAR* is a character string with each entry representing a media (mixture) number or unit number. These are the characters that will be used in the plot. The first entry represents media, unit or bias ID zero; the second entry represents the smallest media, unit or bias ID used in the problem; the third, the next larger media, unit or bias ID used in the problem; etc.

The default values of *CHAR* are:

MEDIA	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SYMBOL		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MEDIA	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
SYMBOL	V	W	X	Y	Z	#	.	$	—	-	)	!	&	>	:	,
MEDIA	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
SYMBOL	G	H	I	J	K	L	M	N	C	P	Q	R	S	T	U	
MEDIA	47	48	49	50	51	52	53	54	55	56	57	58				
SYMBOL		.	%	*	'	=	!	(	@	<	/	0				

RUN= *fnam(19)* enter YES or NO. A value of YES means the problem will be executed if all the data were acceptable. A value of NO specifies the problem will be terminated after data checking is completed. The default value of **RUN** is YES. The value entered in the plot data will override the value entered in the parameter data, Sect. F11.4.3, unless the problem is being restarted.

PLT= *fnam(20)* enter YES or NO. A value of YES specifies that a plot is to be made. If plot data are entered, **PLT** is defaulted to YES. If it is desirable to retain the plot data in the problem description without generating a plot, **PLT=NO** should be entered. **PLT=** can also be entered in the parameter data, Sect. F11.4.3, but will be overridden by the value entered here unless the problem is being restarted.

F11.4.12 REFERENCE

1. N. M. Greene et al., *AMPX: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Libraries from ENDF/B*, ORNL/TM-3706 (March 1976). Also see Sect. F2.3.8 of the SCALE manual.
2. J. R. Knight and L. M. Petrie, *16 and 123 Group Weighting Functions for KENO*, ORNL/TM-4660 (1975).

Tables summarizing KENO-V input data follow.

Table F11.4.7. Summary of parameter data

TITLE — The title must be entered first (80 columns) See Section F11.4.3

PARAMETERS — Format: READ PARAM enter parameter data here END PARAM
If parameters are entered, they must follow the title. See Sections F11.4.3, F11.5.2 and F11.5.3.

KEY	STD.	DEFINITION
RND-	given	random number
TME-	30.min	execution time(min)
TBA-	0.5min	batch time (min)
WTA-	0.	average weight
WTH-	3.0	wt. for splitting
RTL-	1/WTH	Russian Roulette wt.
GEN-	103	No. of generations
NPG-	300	No. per generation
NSK-	3	generations skipped
RES-	0	gens. between restart
NBK-	GEN+25	neutron bank positions
XNB-	0	extra bank entries
NFB-	NPG	fission bank positions
XFB-	0	extra bank entries
X1D-	0	No. of extra 1-D's
LNG-	4000000	words of storage
BEG-	1	restart at this gen.
NBB-	200	blocks for d.a. unit
NLB-	789	length of d.a. block

KEY	STD.	DEFINITION
RUN-	YES	execute problem
FLX-	NO	fluxes
FDN-	NO	fission densities
ADJ-	NO	adjoint calculation
AMX-	NO	all mixture xsecs
XAP-	NO	xsec angles & probs.
XS1-	NO	1-D xsecs
XS2-	NO	2-D xsecs
FKI-	NO	fission spectrum
P1D-	NO	extra 1-D xsecs
FAB-	NO	fiss & abs.
MKP-	NO	matrix by location
CKP-	NO	cofactor k by loc.
FMP-	NO	fiss. prod. by loc.
MKU-	NO	matrix by unit
CKU-	NO	cofactor k by unit
FMU-	NO	fiss. prod. by unit

KEY	STD.	DEFINITION
MKH-	NO	matrix by hole
CKH-	NO	cofactor k by hole
FMH-	NO	fiss. prod. by hole
MHL-	NO	MKH at highest level
MKA-	NO	matrix by array
CKA-	NO	cofactor k by array
FMA-	NO	fiss. prob. by array
MAL-	NO	MKA at higest level
PLT-	YES	printer plots tracks
BUG-	NO	debug print
TRK-	NO	print neutron tracks
PWT-	NO	print avg. weight
PGM-	NO	unprocessed geometry
SMU-	NO	self-multiplication
NUB-	NO	neutrons per fission
PAX-	NO	albedo-xsec array

KEY	STD.	DEFINITION
XSC-	14	mixed xsecs
ALB-	79	albedo
WTS-	80	weights
LIB-	0	working xsecs
SKT-	16	scratch
RST-	0	read restart
WRS-	0	write restart

Table F11.4.8. Summary of array data

ARRAY Format: READ ARRAY array parameters data type orientation data END ARRAY
See Sections F11.5.5, F11.5.6 and F11.5.7

Repeat the sequence ARRAY PARAMETERS DATA TYPE ORIENTATION DATA for each array used in the problem.

ARRAY PARAMETERS

KEYWORD	DEFAULT	DEFINITION
ARA=	1	No. defining the array
NUX=	1	No. of units in X direction
NUY=	1	No. of units in Y direction
NUZ=	1	No. of units in Z direction
GBL=	mxzsrs	The global, universe or overall array number**
COM=	none	delim comment delim optional comment is a maximum of 132 characters.

** Can be defaulted by the code. If specified, it need be entered only once per problem.

DATA TYPE
FILL
LOOP

ORIENTATION DATA FOR FILL

Enter unit numbers to define every position in the array. When entering data utilizing the options in this table, the count field and option field must be adjacent with no imbedded blanks. The operand field may be separated from the option field by one or more blanks. Fill data for an array ends with END FILL.

COUNT FIELD	OPTION FIELD	OPERAND FIELD	COMMENTS
		j	stores j at the current position in the array
i	S	j	stores j in the next i positions in the array
i	*	j	stores j in the next i positions in the array
i	$	j	stores j in the next i positions in the array
	F	j	fills remainder of the array with unit No. j starting with the current array position
	A	j	sets the current position in the array to j
i	S		increments current position in the array by i allows skipping i positions. The value of i may be positive or negative.
i	Q	j	repeats the previous j entries i times. The default value of i is 1.
i	M	j	repeats the previous j entries i times, inverting the sequence each time. The default value of i is 1.
i	R	j	Starting with the entry at -i from the current position, store entries in inverse order until position -(i+j) is reached. Default value of i-1.
i	P	j	alternately stores j and -j in the next i positions of the array.
i	I	j k	provides i entries linearly interpolated between j and k, and the end points (i.e. a total of i+2 points.) j and k must be separated by at least one blank.
	T		terminates the data reading for the array.

ORIENTATION DATA FOR LOOP

Enter the unit number and nine numbers that define the position(s) of that unit. Data for each of these ten entries are repeated until every position in the array has been defined. Data for an array is terminated by entering END LOOP.

ENTER DATA IN THE FORM:

DATA ENTRY	COMMENTS
LTYPE	The unit or box type. LTYPE must be greater than 1.
IX1	Starting position in the X direction. IX1 must be at least 1 and no larger than the value entered for NUX.
IX2	Ending position in the X direction. IX2 must be at least 1 and no larger than the value of NUX.
INCX	The number of units by which increments are made in the X direction.
IY1	The starting position in the Y direction. IY1 must be at least 1 and less than the value entered for NUY.
IY2	Ending position in the Y direction. IY2 must be at least 1 and no larger than the value of NUY.
INCY	The number of units by which increments are made in the positive Y direction.
IZ1	Starting position in the Z direction. IZ1 must be at least 1 and no larger than NUZ. in the positive Y direction.
IZ2	Ending position in the Z direction. IZ2 must be at least 1 and no larger than NUZ.
INCZ	The number of units by which increments are made in the positive Z direction.

Table F11.4.9. Summary of biasing data

BIAS (weighting) Format: READ BIAS keyword correlation data auxiliary END BIAS
See Sections F11.4.7 and F11.5.8

KEYWORD	DESCRIPTION
ID=	CORRELATION DATA will be read next.
	id material ID. Enter from table at right.
	to use weighting data from the library.
	ibgn beginning bias ID
	iend ending bias ID
WT=	AUXILIARY DATA will be read next.
WTS=	AUXILIARY DATA will be read next.
	wttitl material title
	id material ID.
	nsets number of sets of group structures
	REPEAT THKINC, NUMINC, NGPWT, WTAVG NSETS TIMES
	thkinc thickness per increment
	numinc number of increments
	ngpwt number of energy groups for this set of wts
	wtavg enter numinc* ngpwt values of wtavg.

MATERIAL	ID	ENERGY GROUPS	THICKNESS/INCREMENT
concrete	301	16, 27, 123	5 cm
paraffin	400	16, 27, 123	3 cm
water	500	16, 27, 123, 218	3 cm
graphite	6100	16, 27, 123	20 cm

For CORRELATION DATA, the Material ID is chosen from material ID column above (the keyword is ID=).
For AUXILIARY DATA, the material ID is chosen by the user and the keyword is WT= or WTS=.

Beginning and ending bias ID's are defined by the user. The geometry specification that has the bias ID equal to the beginning bias ID utilizes the wt avg's from the first interval of material ID.

Table F11.4.10. Summary of boundary condition data

BNDS (albedo or boundary conditions)

Format: READ BNDS face code albedo name END BNDS
See Section F11.4.7

The sequence FACE CODE ALBEDO NAME is entered as many times as necessary to define the appropriate albedo boundary conditions. The default for all faces is vacuum.

FACE CODES FOR ENTERING BOUNDARY (ALBEDO) CONDITIONS

FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION
+XB=	positive X face	XFC=	both X faces	+YX=	positive X and Y faces	&XY=	positive Y and X faces
&XB=	positive X face	YFC=	both Y faces	&XY=	positive X and Y faces	-XY=	negative X and Y faces
-XB=	negative X face	ZFC=	both Z faces	&YX=	positive X and Y faces	-XZ=	negative X and Z faces
+YB=	positive Y face	+FC=	all positive faces	+XZ=	positive X and Z faces	-YZ=	negative Y and Z faces
&YB=	positive Y face	&FC=	all positive faces	+ZX=	positive X and Z faces	YXF=	all X and Y faces
-YB=	negative Y face	-FC=	all negative faces	&XZ=	positive X and Z faces	ZXF=	all X and Z faces
+ZB=	positive Z face	XYF=	all X and Y faces	&ZX=	positive X and Z faces	ZYF=	all Y and Z faces
&ZB=	positive Z face	XZF=	all X and Z faces	+YZ=	positive Y and Z faces	-YX=	negative X and Y faces
-ZB=	negative Z face	YZF=	all Y and Z faces	+ZY=	positive Y and Z faces	-ZX=	negative X and Z faces
ALL=	all 6 faces	+XY=	positive X and Y faces	&YZ=	positive Y and Z faces	-ZY=	negative Y and Z faces

ALBEDO NAMES AVAILABLE ON THE KENO V ALBEDO LIBRARY, FOR USE WITH THE FACE CODES

ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION
DPGH2O	12 inch double PG water	CONC-4	4 inch concrete differential	VACUUM	vacuum condition
DPOH2O	differential albedo with	CON4	albedo with 4 incident	VOID	
DPG	4 incident angles	CONC4	angles	VACU	
DPO				VAC	
		CONC-8	8 inch concrete differential		
H2O	12 inch water differential	CON8	albedo with 4 incident	SPECULAR	mirror image reflection
WATER	albedo with 4 incident	CONC8	angles	MIRROR	
	angles			MIRR	
				SPEC	
PARAFFIN	12 inch paraffin differen-	CONC-12	12 inch concrete differential	SPE	
PARA	tial albedo with 4 incident	CON12	albedo with 4 incident	MIR	
WAX	angles	CONC12	angles		
CARBON	200 cm. carbon differential	CONC-16	16 inch concrete differential	PERIODIC	periodic boundary condition
GRAPHITE	albedo with 4 incident angles	CON16	albedo with 4 incident	PERI	
C		CONC16	angles	PER	
ETHYLENE	12 inch polyethylene	CONC-24	24 inch concrete differential		
POLY	differential albedo with	CON24	albedo with 4 incident		
CH2	4 incident angles	CONC24	angles		

Table F11.4.11. Summary of geometry data

GEOMETRY (region)

Format: READ GEOM enter geometry region data here END GEOM
See Sections F11.4.4, F11.5.1.2, F11.5.6 and F11.5.7

GEOMETRY REGION DATA consists of SIMPLE GEOMETRY REGION DATA and EXTENDED GEOMETRY REGION DATA.

ENTER GEOMETRY REGION DATA IN THE FOLLOWING FORM:

OPTIONAL GLOBAL SPECIFICATION
UNIT n
OPTIONAL GEOMETRY COMMENT
SIMPLE GEOMETRY REGION DATA
and/or
EXTENDED GEOMETRY REGION DATA

* *

ENTER SIMPLE GEOMETRY REGION DATA IN THE FOLLOWING FORM:

GLOBAL Enter only to specify this unit as the global unit.
UNIT n
COM-delim comment delim This optional comment can be up to 132 characters.
It must begin and end with a delimiter.
fgeom mix no. bias ID dimensions optional origin data (ORIGN coordinates) optional chord data (CHORD)

Enter as many geometry description specifications as necessary to describe the unit and as many units as necessary to describe the system.

SIMPLE GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA	TYPE 4 DATA	TYPE 5 DATA	TYPE 6 DATA
fgeom	SPHERE HEMISPHERE HEMISPHERE+X HEMISPHERE-X HEMISPHERE+Y HEMISPHERE-Y HEMISPHERE+Z HEMISPHERE-Z	XCYLINDER XHEMICYL+Y XHEMICYL-Y XHEMICYL+Z XHEMICYL-Z	YCYLINDER YHEMICYL+X YHEMICYL-X YHEMICYL+Z YHEMICYL-Z	CYLINDER ZCYLINDER ZHEMICYL+X ZHEMICYL-X ZHEMICYL+Y ZHEMICYL-Y	CUBE	CUBOID
dimensions	R (radius)	R +N -N	R +N -N	R +N -N	+X -X	+X -X +Y -Y +Z -Z
optional origin coord.	Enter the X Y Z coord. of origin	Enter the Y Z coord. of center-line	Enter the X Z coord. of center-line	Enter the X Y coord. of center-line	omit	omit
optional chord data**	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	omit	omit

**Chord data is not applicable for SPHERE XCYLINDER YCYLINDER CYLINDER or ZCYLINDER

Table F11.4.11 (continued)

GEOMETRY (region) (cont.)

ENTER EXTENDED GEOMETRY DATA IN THE FOLLOWING FORM:

fgeom ref. ID bias ID thickness per region origin coordinates nreg

EXTENDED GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA
fgeom	ARRAY CORE COREBDY COREBNDS COREBOUN	HOLE	REPLICATE REFLECTOR
ref. ID	array no.	emplaced unit number	mixture no. in generated regions
bias ID	omit for ARRAY	omit	first bias ID
thick./reg.	omit	omit	variable**
origin coord.	Enter the X Y Z coord. of most neg. pt. of array	Enter the X Y Z coord. of origin	omit
nreg	omit	omit	No. of regions to be generated

**The number of dimensions to be entered is the same as the region preceding the replicate or reflector specification because the generated regions have that shape. The value of the dimensions is the thickness of each generated region of material on that surface.

Table F11.4.12. Summary of mixing table data

MIXTURES Format: READ MIXT xsec parameters mixing table END MIXT
This data is entered only if an AMPX working format library is being used. (LIB-) in the parameter data. Do not enter if an ICE mixed library is used, (XSC-) in the parameter data.
See Sections F11.4.10 and F11.5.5

XSEC PARAMETERS consists of keywords and associated values. These parameters, if entered, need be entered only once.

KEYWORD	DEFAULT	DEFINITION
SCT-	1	No. of discrete scattering angles. 0 is isotropic 1 is P1 2 is P3 3 is P5
EPS-	.00003	Cross section message cutoff value. Use to suppress message No. E5-60

MIXING TABLE DATA consists of (1) a keyword and mixture ID for the mixture
The keyword is MIX-
The desired mixture number follows the keyword.
(2) nuclide ID**
(3) number density**
** the sequence (2) (3) is repeated for each nuclide in the mixture.

REPEAT the sequence (1) (2)'s (3)'s until all the mixtures have been described.

Table F11.4.13. Summary of plot data

PLOT Format: READ PLOT plot parameters END PLOT
plot parameters must be entered for each plot that is to be made.
See Sections F11.4.11 and F11.5.9

KEYWORD	DEFAULT	DEFINITION
TTL-	prob. title	delim ptitl delim A one character delimiter signals the beginning and end of the title. ptitl is the plot title.
PIC-	MAT	Type of picture: MIXTURE, UNIT NO. or BIAS ID NO.
		MIXTURE -------- MAT
		MIX
		MIXT
		MIXTURE
		MED1
		MEDIA
		UNIT NO. ------- BOX
		BOXT
		BOXTYPE
		UNT
		UNIT
		UNITTYPE
		BIAS ID NO. ---- IMP
		BIAS
		BIASID
		WTS
		WEIG
		WEIGHTS
		WGT
		WGTS
XUL-	prev. plot	X coord. of upper left corner of plot
YUL-	prev. plot	Y coord. of upper left corner of plot
ZUL-	prev. plot	Z coord. of upper left corner of plot
XLR-	prev. plot	X coord. of lower right corner of plot
YLR-	prev. plot	Y coord. of lower right corner of plot
ZLR-	prev. plot	Z coord. of lower right corner of plot

KEYWORD	DEFAULT	DEFINITION
UAX-	prev. plot 0 IF VAX OR WAX is read	X component of direction cosine for the AX axis of the plot (across)
VAX-	prev. plot 0 IF UAX OR WAX is read	Y component of direction cosine for the AX axis of the plot (across)
WAX-	prev. plot 0 IF UAX OR VAX is read	Z component of direction cosine for the AX axis of the plot (across)
UDN-	prev. plot 0 IF VDN OR WDN is read	X component of direction cosine for the DN axis of the plot (down)
VDN-	prev. plot 0 IF UDN OR WDN is read	Y component of direction cosine for the DN axis of the plot (down)
WDN-	prev. plot 0 IF UDN OR VDN is read	Z component of direction cosine for the DN axis of the plot (down)
DLX-		horizontal spacing between points on plot
DLD-		vertical spacing between points on plot
NAX-		no. of intervals to be printed across page
NDN-		no. of intervals to be printed down page
LPI-	6	lines per inch printed down the page
NCH-	CHRS**	delim CHRS delim a one character delimiter signals the beginning and end of the character string.
RUN-	YES	YES allows the problem to execute NO terminates problem after data checking
PLT-	YES	YES allows the plot(s) to be made NO allows reading the plot data without making a plot.

PLOT ORIGIN:

(1) SINGLE UNIT-coincides with origin of geometry description.

(2) BARE ARRAY-at the most negative point of the array, (lower left-hand back corner of the global array).

(3) REFLECTED ARRAY- coincides with the origin of the CORE or ARRAY description of the global array.

**default values of CHRS are given below.

MEDIA	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CHRS		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M
MEDIA	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42			
CHRS	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	$	,	$	-	+	)	(			
MEDIA	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58							
CHRS	&	>	:	;	.	-	%	*	"	=	!	(	@	<	/	0							

Table F11.4.14. Summary of starting data

START

Format: READ START enter start data here END START
The default value of start type is zero.
See Section F11.4.8

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
0	none	MST	uniform
		XSM	
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
1	MST	XSM	cosine
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
2	MST	XSM	cosine with
	MXS	XSP	fraction in
	MYS	YSM	specified
	MZS	YSP	unit
	FCT	ZSM	
		ZSP	
		RFL	
		PSP	

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
3	MST	XFS	spike
	TFX	PSP	
	TFY		
	TFZ		
	MXS		
	MYS		
	MZS		
4	MST	XFS	multiple
	TFX	PSP	spikes
	TFY		
	TFZ		
	MBX		
5	MST	PSP	in specified
	MBX		units
6	MST	MXS	arbitrary
	TFX	MYS	points
	TFY	MZS	
	TFZ	XFS	
	LNU*	PS6	
		PSP	

*LNU must be the last entry for each set of start 6 data. The LNU of each successive set of data must be larger than the last.

KEYWORD	DEFAULT	DEFINITION
MST=	0	start type
TFX=	0.0	X coordinate
TFY=	0.0	Y coordinate
TFZ=	0.0	Z coordinate
MXS=	0	X index of unit pos.
MYS=	0	Y index of unit pos.
MZS=	0	Z index of unit pos.
XFS=		fission spectra
LNU=	0	number of last neutron
MBX=	0	source unit number
FCT=	0	fraction
XSM=	-X	-X of source cuboid
XSP=	+X	+X of source cuboid
YSM=	-Y	-Y of source cuboid
YSP=	+Y	+Y of source cuboid
ZSM=	-Z	-Z of source cuboid
ZSP=	+Z	+Z of source cuboid
RFL=	NO	start in reflector
PS6=	NO	print start 6 input
PSP=	NO	print starting points

F11.5 NOTES FOR KENO V USERS

This section provides assorted tips primarily designed to assist the KENO V user with problem mock-ups. Some information concerning methods utilized by KENO V is also included.

F11.5.1 DATA ENTRY

The KENO V data input is entered in blocks that begin and end with keywords as described in Sect. F11.4.1. Only one set of parameter data can be entered for a problem. However, for other data blocks, it is possible to enter more than one block of the same kind of data. When this is done, only the last block of that kind of data is retained for use by the problem.

Within data blocks, a number, x, can be repeated n times by specifying nRx, n*x, or n$x.

F11.5.1.1 *Multiple and Scattered Entries in the Mixing Table*

In the following examples, assume 1001 is the nuclide ID for hydrogen, 8016 is the nuclide ID for oxygen, 92235 is the nuclide ID for ^{235}U, and 92238 is the nuclide ID for ^{238}U. If a given nuclide ID is used more than once in the same mixture, the result is the summing of all the number densities associated with that nuclide. For example:

MIX=1 92235 4.3-2 92238 2.6-3 1001 3.7-2 92235 1.1-3 8016 1.8-2

would be the same as entering:

MIX=1 92235 4.41-2 92238 2.6-3 1001 3.7-2 8016 1.8-2

A belated entry for a mixture can be made as follows:

MIX=1 1001 6.6-2 MIX=2 92235 4.3-2 92238 2.6-3 MIX=1 8016 3.3-2

This is the same as entering:

MIX=1 1001 6.6-2 8016 3.3-2 MIX=2 92235 4.3-2 92238 2.6-3

F11.5.1.2 *Multiple Entries in Geometry Data*

Individual geometry regions cannot be replaced by adding an additional description. However, entire unit or box type descriptions can be replaced by adding a new description having the same unit number. The last description entered for a unit is used in the calculation. For example:

```
READ GEOM  UNIT 1 SPHERE 1 1 5.0 CUBE 0 1 10.0 -10.0
UNIT 2 CYLINDER 1 1 2.0 5.0 -5.0  CUBE  0 1 10.0 -10.0
UNIT 1  CUBOID  1 1 1.0 -1.5 2.5 -2.0 5.0 -6.0  CUBE  0 1 10.0 -10.0
END GEOM
```

is the same as entering:

READ GEOM UNIT 1 CUBOID 1 1 1.0 -1.5 2.5 -2.0 5.0 -6.0
CUBE 0 1 10.0 -10.0
UNIT 2 CYLINDER 1 1 2.0 5.0 -5.0 CUBE 0 1 10.0 -10.0 END GEOM

or:

READ GEOM UNIT 2 CYLINDER 1 1 2.0 5.0 -5.0 CUBE 0 1 10.0 -10.0
UNIT 1 CUBOID 1 1 1.0 -1.5 2.5 -2.0 5.0 -6.0 CUBE 0 1 10.0 -10.0
END GEOM

The order of entry for unit or box type descriptions is not important because the unit number is assigned as the value following the word UNIT. They need not be entered sequentially nor be numbered sequentially. It is perfectly acceptable to input Units 2, 3, and 5, omitting Units 1 and 4 as long as Units 1 and 4 are not referenced in the problem. It is also acceptable to scramble the order of entry as in entering Units 3, 2, and 5.

F11.5.2 DEFAULT LOGICAL UNIT NUMBERS FOR KENO V

The logical unit numbers for data utilized by KENO V are listed in Table F11.5.1.

Table F11.5.1. KENO V logical unit numbers

Function	Parameter Name	Unit No.	Variable Name
Card Input Data		5	INPT
Program Output		6	OUTPT
Albedo Data	ALB =	79	ALBDO
Scratch Unit	SKT =	16	SKRT
Read Restart Data	RST =	0[a]	RSTRT
		34[b]	RSTRT
Write Restart Data	WRS =	0[a]	WSTRT
		35[c]	WSTRT
Direct Access Storage for Input Data		8	DIRECT(1)
Direct Access Storage for Supergrouped Data		9	DIRECT(2)
Direct Access Storage for Cross Section Mixing		10	DIRECT(3)
Mixed Cross Section Data Set	XSC =	14[d]	ICEXS
Group-Dependent Weights	WTS =	80	WTS
AMPX Working Format Cross Sections	LIB =	0[a]	AMPXS

[a]defaulted to zero
[b]defaulted to 34 if BEG= a number greater than 1 and RSTRT=0
[c]defaulted to 35 if RES= a number greater than zero and WSTRT=0
[d]defaulted to 0; if LIB= a number greater than zero, ICEXS is defaulted to 14

F11.5.3 PARAMETER INPUT

When the parameter data block is input for a problem, the same keyword may be entered several times. The last value that is entered is used in the problem. Data may be entered as follows:

READ PARAM FLX=YES NGP=1000 TME=0.5 TME=1.0
NPG=50 TME=10.0 FLX=NO
NPG=500
END PARA

This will result in the problem having FLX=NO, TME=10.0, and NPG=500. It may be more convenient for the user to insert a new value than to change the existing data.

Certain parameter default values should not be overridden unless the user has a very good reason to do so. These parameters are (1) X1D= which defines the number of extra 1-D cross sections. The use of extra 1-D cross sections -- other than the use of the fission cross section for calculating the average number of neutrons per fission — requires programming changes to the code; (2) NFB= which defines the number of neutrons that can be entered in the fission bank; (3) XFB= which defines the number of extra positions in the fission bank (the fission bank is where the information related to a fission is stored); (4) NBK= which defines the number of neutrons that can be entered in the neutron bank; (5) XNB= which defines the number of extra positions in the neutron bank (the neutron bank contains information about each history); (6) WTH= which defines the factor that determines when Russian roulette is played; (7) WTA= which defines the default average weight given to a neutron that survives Russian roulette; (8) WTL= which defines the weight at which Russian roulette is played; and (9) LNG= which sets the maximum words of storage available to the program. It is recommended that BUG=, the flag for printing debug information, never be set to YES. The user would have to look at the FORTRAN coding to determine what information is printed. BUG=YES prints massive amounts of sparsely labeled information. The user should only rarely consider using TRK=YES. This generates thousands of lines of well-labeled print that provides information about each history at key locations during the tracking procedure. All other parameters can be changed at will to provide features the user wishes to activate.

F11.5.4 CROSS SECTIONS

KENO V always uses cross sections from a mixed cross-section data file. The format of this file is the Monte Carlo processed cross-section file from ICE-II or ICE-S (Sect. F8). A mixed cross-section file can be created by (1) executing ICE, or (2) by using an AMPX working format library and entering mixing table data in KENO V.

F11.5.4.1 *Use a Premixed Cross-Section Data File*

A premixed cross-section data file from ICE or a previous KENO V case may be used. This file should be specified in the job control language on the unit number associated with the parameter XSC=. When a mixing table data block is entered, the premixed cross-section data file will be rewritten. Therefore, a mixing table should not be entered if a premixed cross-section data file is used. The user should verify that the mixtures created by ICE or the previous KENO V case are consistent with those used in the geometry data of the problem.

F11.5.4.2 *Use an AMPX Working Library*

When an AMPX working library is used, a file definition card must be used in the job control language to specify the AMPX working library on the unit associated with the parameter LIB=. If the mixed cross-section data file is to be saved, a file definition card must be used in the job control language to specify the mixed cross-section file on the unit associated with the parameter XSC=.

Mixing table data must always be entered when an AMPX working library is used. ID's used in the mixing table must match the ID's on the AMPX working library.

F11.5.4.3 *Number of Scattering Angles*

The number of scattering angles is defaulted to one. This default is not adequate for many applications. The user should specify the scattering angle to be consistent with the cross sections being used. The number of scattering angles is entered in the cross-section mixing table by using the keyword SCT=. See Sect. F11.4.10.

The order of the last Legendre coefficient to be preserved in the scattering distribution is equal to (2xSCT-1). SCT=1 could be used with a P_1 cross-section set such as the 16-group Hansen Roach cross-section library, and SCT=2, for a P_3 cross-section set such as the SCALE 27-group cross-section library. Isotropic scattering is achieved by entering SCT=0.

F11.5.4.4 *Cross-Section Message Cutoff*

The cross-section message cutoff value, *pbxs*, is defaulted to $3\text{x}10^{-5}$. Warning messages that are generated when errors are encountered in the P_L expansion of the group-to-group transfers will be suppressed if the P_0 cross section for that particular energy transfer is less than *pbxs*. The value of *pbxs* is specified in the cross-section mixing table by using the keyword EPS=. See Sect. F11.4.10.

The default value of *pbxs* is sufficient to assure that warning messages will not be printed for most of the SCALE P_1 and P_3 cross section libraries. However, the 123 GROUPGMTH library requires a value of *pbxs* as large as $1\text{x}10^{-1}$ if P_3 cross sections are specified.

If the default value of *pbxs* allows too many warning messages to be printed, a value can be determined from the printed messages by choosing a number larger than the P_0 component on the first line as shown below.

THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
(P_0) (P_1) (P_2) ... (P_n)
THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
(M_1) (M_2) ... (M_n)
THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
(M_1) (M_2) ... (M_n)
THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
(P_0) (P_1) (P_2) ... (P_n)

_____ MOMENTS WERE ACCEPTED

For the following messages, EPS=7.5-4 would cause all three messages to be suppressed. A value less than 7.295643-4 and greater than 5.662553-4 would suppress the last two messages, and a value less than 5.662553-4 and greater than 4.398203-4 would suppress only the last message.

```
K5-60  THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE  2 HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP  4 TO GROUP 53
         2 MOMENTS WERE ACCEPTED
       THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
     7.295643E-04  4.786644E-05 -3.543105E-04 -2.530790E-04
       THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
     6.560963D-02  9.568884D-03 -9.939045D-02
       THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
     6.560963D-02  9.568884D-03 -9.713710D-04
       THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
     7.295643E-04  4.786644D-05 -3.543105D-04 -7.002796D-05

K5-60  THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE  2 HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP  4 TO GROUP 54
         2 MOMENTS WERE ACCEPTED
       THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
     5.662553E-04  3.289824E-05 -2.767628E-04 -1.976878E-04
       THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
     5.809790D-02  7.493547D-03 -1.047869D-01
       THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
     5.809790D-02  7.493547D-03  6.736284D-04
       THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
     5.662553E-04  3.289824D-05 -2.767628D-04 -4.839374D-05

K5-60  THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE  2 HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP  4 TO GROUP 55
         2 MOMENTS WERE ACCEPTED
       THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
     4.398203E-04  2.261055E-05 -2.160441E-04 -1.543172E-04
       THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
     5.140860D-02  5.860046D-03 -1.095005D-01
       THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
     5.140860D-02  5.860046D-03  4.661151D-04
       THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
     4.398203E-04  2.261055D-05 -2.160441D-04 -3.340330D-05
```

The user need not attempt to suppress all the K5-60 messages. They are printed to inform the user of the fact that the moments of the angular distribution are not moments of a valid probability distribution. The fourth and fifth lines of the message list the original P_n components and their moments. The sixth and seventh lines list the new corrected moments and their corresponding P_n components.

For most criticality problems, the first moment contributions are much more significant than the contributions of the higher order moments. Thus, the higher order moments may not affect the results significantly. The user can look at the original moments and corrected moments and make a judgment as to the significance of the change in the moments.

F11.5.5 MIXING TABLE

Mixtures can be utilized in defining other mixtures. When defining mixture numbers, care should be taken to avoid using a mixture number that is identical to a nuclide ID number if the mixture is to be used in defining another mixture. If a mixture number is defined more than once, it results in a summing effect.

The nuclide mixing loop is done before the mixture mixing loop which mixes in the order of data entry. Thus, the order of mixing mixtures into other mixtures is important because a mixture must be defined before it is used in another mixture. Some examples of correct and incorrect mixing are shown below, using 1001 as the nuclide ID for hydrogen, 8016 as the nuclide ID for oxygen, 92235 as the nuclide ID for ^{235}U, and 92238 as the nuclide ID for ^{238}U.

EXAMPLES OF CORRECT USAGE

(1) READ MIXT MIX=1 1001 6.6-2 8016 3.3-2 MIX=2 1 0.5 END MIXT

This results in mixture 1 being full density water and mixture 2 being half density water.

(2) READ MIXT MIX=1 2 0.5 MIX=3 1 0.5 MIX=2 1001 6.6-2 8016 3.3-2 END MIXT

This results in mixture 1 being half density water, mixture 2 being full density water, and mixture 3 being quarter density water. Because the nuclide mixing loop is done first, mixture 2 is created first and is available to create mixture 1, which is then available to create mixture 3.

(3) READ MIXT MIX=1 1001 6.6-2 8016 3.3-2 MIX=2 92235 7.5-4 92238 2.3-2 8016 4.6-2 1 .01 END MIXT

This results in mixture 1 being full density water and mixture 2 being uranium oxide containing 0.01 density water.

(4) READ MIXT MIX=1 1001 6.6-2 8016 3.3-2 MIX=2 92235 4.4-2 92238 2.6-3 MIX=1 1 0.5 END MIXT

This results in mixture 1 being water at 1.5 density (1001 9.9-2 and 8016 4.95-2) and mixture 2 is highly enriched uranium metal.

EXAMPLES OF INCORRECT USAGE

(1) READ MIXT MIX=3 1 0.75 MIX=1 2 0.5 MIX=2 1001 6.6-2 8016 3.3-2 END MIXT

Here the intent is for mixture 2 to be full density water, mixture 1 to be half density water, and mixture 3 to be 3/8 (0.75 x 0.5) density water. Instead, the result for mixture 3 is a void, mixture 1 is half density water, and mixture 2 is full density water. This is because the nuclide mixing loop is done first, thus defining mixture 2. The mixture mixing loop is done next. Mixture 3 is defined to be mixture 1 multiplied by 0.75, but since mixture 1 has not been defined, 0.75 of zero is zero. Mixture 1 is then defined to be mixture 2 multiplied by 0.5. If the definition of mixture 1 preceded the definition of mixture 3, as in (2) under examples of correct usage, it would work correctly.

(2) READ MIXT MIX=1 1001 6.6-2 8016 3.3-2 MIX=1001 92235 4.4-2 92238 2.6-3 MIX=2 1001 0.5 END MIXT

This results in mixture 1 being full density water, mixture 1001 being uranium metal and mixture 2 being hydrogen with a number density of 0.5 because 1001 is the nuclide ID number for hydrogen. When a mixture number is identical to a nuclide ID and is used in mixing, that number is assumed to be a nuclide ID rather than a mixture number. The intent was for mixture 1 to be full density water, mixture 1001 to be uranium metal, and mixture 2 to be half density uranium metal.

F11.5.6 GEOMETRY

In general, KENO V geometry descriptions consist of (1) geometry data (Sect. F11.4.4) defining the geometrical shapes present in the problem, and (2) array data (Sect. F11.4.5) defining the placement of the units that were defined in the geometry data. The geometry data block is prefaced by READ GEOM and the array data block is prefaced by READ ARRAY

When a three-dimensional geometrical configuration is described as KENO V geometry data, it may be necessary to describe portions of the configuration individually. These individual partial descriptions of the configuration are called units. KENO V geometry modeling is subject to the following restrictions:

(1) Each geometry region in a unit must completely enclose all geometry regions which precede it. Boundaries of the surfaces of the regions may be shared or tangent, but they must not intersect.

(2) All geometrical surfaces must be describable as spheres, hemispheres, cylinders, hemicylinders, cubes, or cuboids.

(3) When one or more units are utilized to describe an array, each unit used in the array must have a cube or cuboid as its outer region.

(4) When several units are utilized to describe an array, the adjacent faces of units in contact with each other must be the same size and shape.

(5) As many holes as will snugly fit without intersecting can be placed in a region. Holes cannot intersect each other or any of the regions within the unit they are placed in. Holes are described in more detail in Sect. F11.5.6.1.

(6) Complicated systems may require multiple arrays to describe the system. Arrays may be placed in units. These units may be used to create other arrays or may be placed in other units by using holes. Multiple arrays are described in more detail in Sect. F11.5.6.2.

The geometry package allows any applicable shape to be enclosed by any other applicable shape, subject only to the complete enclosure restriction. The implication of this type of description is that the entire volume between two sequential geometrical surfaces contains only one mixture unless holes are present. Therefore, the entire volume within the surface, defined by the first geometry card in a unit, contains the mixture that is specified on that card. The volume between the surfaces, defined by two consecutive geometry cards, contains the mixture that is specified on the second card. A void is specified by a mixture ID of zero. If holes are present in the volume between two surfaces, the volume of that region is reduced by the hole volume(s).

If the problem requires several units to describe its geometrical characteristics, each unit that is used in an array must have as its outer surface a rectangular parallelepiped. Thus, it may be necessary to define a void region that is used to achieve the desired spacing. In order to describe the composite overall geometrical characteristics of the problem, these units may be arranged in a rectangular array by specifying the number of units in the x, y, and z directions. If more than one unit is involved, data must be entered to define the number assigned to the array and the placement of the individual units in the array. The array number, the number of units in the x, y, and z directions, and the placement data are called array data (Sect. F11.4.5).

Surrounding regions of any shape may be placed around an array and may consist of any number of regions in any order, subject to the complete enclosure restriction. All of the surrounding regions must be described about a point of origin; consequently, the location of this origin is defined by the ARRAY PLACEMENT description. The surrounding regions are then described relative to this origin.

In addition to array problems, the geometry package allows single unit problems (i.e., problems that do not contain array data). The last geometrical region in a single unit problem need not be a cube or cuboid. Boundary conditions can be applied to a single unit problem only if the outer region is a rectangular parallelepiped. Matrix information can be calculated only if the corresponding geometry

description is utilized in the problem. For example, matrix by hole number requires the utilization of holes, matrix by array position requires an array, etc.

To create a geometry mock-up from a physical configuration, the user should exercise a degree of ingenuity and keep in mind the restrictions mentioned earlier. It is important to realize there may be several ways of correctly describing the same physical configuration. Careful analysis of the system can pay off in terms of a simpler mock-up and shorter computer running time. A mock-up with fewer geometry regions may run faster than the same mock-up with extraneous regions. The number of units or box types used can affect the running time, because a transformation of coordinates must be made every time a history moves from one unit into another. Thus, if the size of a unit is small, relative to the mean free path, a larger percentage of time is spent processing the transformation of coordinates.

CORE or ARRAY: The geometry word CORE was used in KENO IV and earlier versions of KENO V to create a rectangular parallelepiped that snugly fit the exterior of the array. The purpose of this region was to define the location of the array relative to the geometry regions external to the CORE. KENO IV and early versions of KENO V allowed only one array in a problem.

KENO V.a allows multiple arrays within a problem, so the geometry word CORE is used to specify the array number of the array, in addition to encasing it in a snug fitting rectangular parallelepiped. The data that must be entered are the geometry word (CORE), the array number, the bias ID, and the x, y, and z coordinates of the most negative point in the array. The geometry word ARRAY was added to allow specifing the array number and encasing the specified array in a snug fitting rectangular parallelepiped without having to specify the bias ID. The data that must be entered are the geometry word (ARRAY), the array number, and the x, y, and z coordinates of the most negative point in the array. The authors feel that the geometry word ARRAY is more descriptive of the function performed, but the geometry word CORE has been retained to allow executing problems that were run on earlier versions of KENO V.

REFLECTOR or REPLICATE: The geometry word REFLECTOR was used in KENO IV to generate cuboidal regions external to the array. If a snug fitting rectangular parallelepiped (CORE) was not entered in the problem, it was generated by the code. KENO V retained the geometry word REFLECTOR but with a different function. It does not generate a snug fitting rectangular parallelepiped (CORE) for an array, and it is not limited to generating cuboidal regions. Instead, it generates regions having the shape of the geometry region preceding the REFLECTOR description. The geometry word REPLICATE is used in KENO V.a as a synonym for REFLECTOR. The data is identical for both geometry words and their function is identical. Because these geometry words produce regions having the shape of the previous regions, REPLICATE may be considered to be more representative of the actual situation.

Geometry dimensions: The geometry dimensions utilized in KENO have traditionally required an entry for each required dimension. For example, a 20x20x2.5-cm rectangular parallelepiped would have been described as: CUBOID 1 1 10.0 -10.0 10.0 -10.0 1.25 -1.25. By using the P option (see Table F11.4.2), the same rectangular parallelepiped could be described as: CUBOID 1 1 4P10.0 2P1.25. The P option simply adds the dimension following the P, the number of times stated before the P and reverses the sign every other time. 6P8.0 is equivalent to 8.0 -8.0 8.0 -8.0 8.0 -8.0.

Geometry Comments: A comment can be entered for each unit in the geometry region data. Similarly, a comment can be entered for each array in the array definition data. A comment can be entered using the keyword COM=. This is followed by a comment whose maxmum length is 132 characters. The comment must be preceded by and be terminated by a delimiter character, which is the first non-blank character encountered after the COM=. One comment is allowed for each unit in the geometry region data. If multiple comments are entered for a unit, the last one is used. The comment can be entered anywhere after the UNIT NUMBER DESCRIPTION where a keyword is expected (Sect. F11.4.4). See the following example.

```
READ GEOM
UNIT 1
COM=*SPHERICAL METAL UNIT*
SPHERE 1 1 5.0
CUBE 0 1 2P5.0
UNIT 2
CYLINDER 1 1 5.0 2P5.0
CUBE 0 1 2P5.0
COM=/CYLINDRICAL METAL UNIT/
UNIT 3
HEMISPHE+X 1 1 5.0
COM='HEMISPHERICAL METAL UNIT'
CUBE 0 1 2P5.0
UNIT 4
COM='ARRAY OF SPHERICAL UNITS'
ARRAY 1 3*0.0
UNIT 5
COM='ARRAY OF CYLINDRICAL UNITS'
ARRAY 2 3*0.0
UNIT 6
COM='ARRAY OF HEMISPHERICAL UNITS'
ARRAY 3 3*0.0
END GEOM
```

One comment is allowed for each array in the ARRAY DEFINITION DATA. The rules governing these comments are the same as those listed above. However, the comment for an array must precede the UNIT ORIENTATION DESCRIPTION. It can precede the array number (Sect. F11.4.5). Examples of correct array comments are given below.

```
READ ARRAY
COM='ARRA OF SPHERICAL METAL UNITS'
ARA=1 NUX=2 NUY=2 NUZ=2 FILL F1 END FILL
ARA=2 COM='ARRAY OF CYLINDRICAL METAL UNITS'
NUX=2 NUY=2 NUZ=2 FILL F2 END FILL
ARA=3 NUX=2 NUY=2 NUZ=2
COM='ARRAY OF HEMISPHERICAL METAL UNITS'
FILL F3 END FILL
ARA=4 COM='COMPOSITE ARRA OF ARRAYS. Z=1 IS SPHERES, Z=2 IS CYLINDERS,
  Z=3 IS HEMISPHERES'
NUX=1 NUY=1 NUZ=3 FILL 4 5 6 END FILL
```

Some of the basics of KENO V geometry are illustrated in the following examples:

EXAMPLE 1. Assume a stack of six cylindrical disks, each 5 cm in radius and 2 cm thick. The bottom disk is composed of material 1, and the next disk is composed of material 2, etc., alternately throughout the stack. A square plate of material 3, 20 cm on a side and 2.5 cm thick, is centered on top of the stack. This configuration is shown in Fig. F11.5.1.

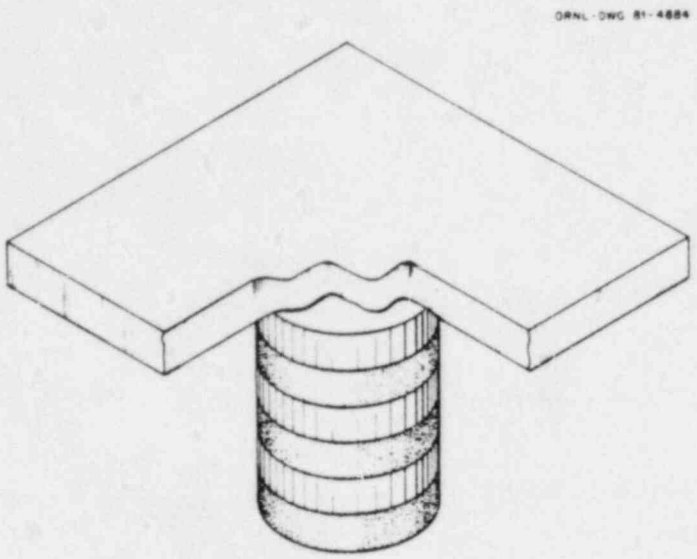


Fig. F11.5.1. Stack of disks with a square cap

This problem can be described as a single unit problem, by describing the cylindrical portion first. In this instance, the origin has been chosen at the center bottom of the bottom disk. The bottom disk is defined by the first cylinder description; the next disk is defined by the difference between the first and second cylinder descriptions. That is, since they both have a radius of 5.0 and a -z length of 0.0, the first cylinder containing material 1 exists from z=0.0 to z=2.0 and the second cylinder, containing material 2, exists from z=2.0 to z=4.0. When all the disks have been described, a void cuboid having the same x and y dimensions as the square plate and the same z dimensions as the stack of disks is defined. The square plate of material 3 is then defined on top of the stack. Omission of the first cuboid description would result in the stack of disks being encased in a solid cuboid of material 3, instead of having a flat plate on top of the stack. The geometry input is shown below.

Data description 1, Example 1.

```
READ GEOM
CYLINDER 1 1   5.0 2.0   0.0
CYLINDER 2 1   5.0 4.0   0.0
CYLINDER 1 1   5.0 6.0   0.0
CYLINDER 2 1   5.0 8.0   0.0
CYLINDER 1 1   5.0 10.0  0.0
CYLINDER 2 1   5.0 12.0  0.0
CUBOID 0 1 10.0 -10.0 10.0 -10.0 12.0 0.0
CUBOID 3 1 10.0 -10.0 10.0 -10.0 14.5 0.0
```

An alternate description of the same example is given below. The origin has been chosen at the center of the disk of material 1, nearest the center of the stack. This disk of material 1 is defined by the first cylinder description, and the disks of material 2 on either side of it are defined by the second cylinder description. The top and bottom disks of material 1 are defined by the third cylinder, and the top disk of material 2 is defined by the last cylinder. The square plate is defined by the two cuboids.

Data description 2, Example 1.

```
READ GEOM
CYLINDER 1 1  5.0   1.0 -1.0
CYLINDER 2 1  5.0   3.0 -3.0
CYLINDER 1 1  5.0   5.0 -5.0
CYLINDER 2 1  5.0   7.0 -5.0
CUBOID   0 1 10.0 -10.0 10.0 -10.0 7.0 -5.0
CUBOID   3 1 10.0 -10.0 10.0 -10.0 9.5 -5.0
```

Example 1 can also be described as a bare array. Define three different unit types. Unit 1 will define a disk of material 1, Unit 2 will define a disk of material 2, and Unit 3 will define the square plate of material 3. The origin of each unit is defined at the center bottom of the disk or plate being described. The geometry input for this arrangement is shown below.

Data description 3, Example 1.

```
READ GEOM
UNIT 1
CYLINDER  1 1  5.0   2.0  0.0
CUBOID    0 1 10.0 -10.0 10.0 -10.0 2.0 0.0
UNIT 2
CYLINDER  2 1  5.0   2.0  0.0
CUBOID    0 1 10.0 -10.0 10.0 -10.0 2.0 0.0
UNIT 3
CUBOID    3 1 10.0 -10.0 10.0 -10.0 2.5 0.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=7 FILL 1 2 1 2 1 2 3 END ARRAY
```

If the user wishes the origin of each unit to be at its center, the geometry region data can be input as shown below. The array data would be identical to that of data description 3, Example 1.

Data description 4, Example 1.

```
READ GEOM
UNIT 1
CYLINDER  1 1  5.0   1.0 -1.0
CUBOID    0 1 10.0 -10.0 10.0 -10.0 1.0 -1.0
UNIT 2
CYLINDER  2 1  5.0   1.0 -1.0
CUBOID    0 1 10.0 -10.0 10.0 -10.0 1.0 -1.0
UNIT 3
CUBOID      3 1 10.0 -10.0 10.0 -10.0 1.25 -1.25
END GEOM
```

Be aware that each unit in a geometry description can have its origin defined independent of the other units. It would be correct to use Units 1 and 3 from data descriptions 3, and Unit 2 from data description 4. The array data would remain the same as data description 3, Example 1. The user should define the origin of each unit to be as convenient as possible for the chosen description.

Another method of describing Example 1 as a bare array is to define Unit 1 to be a disk of material 1, topped by a disk of material 2. The origin has been chosen at the center bottom of the disk of material 1. Unit 2 is the square plate of material 3 with the origin at the center of the unit. The array consists of three Unit 1's, topped by a Unit 2 as shown below.

Data description 5, Example 1.

```
READ GEOM
UNIT 1
CYLINDER  1 1  5.0   2.0  0.0
CYLINDER  2 1  5.0   4.0  0.0
CUBOID    0 1 10.0 -10.0 10.0 -10.0 4.0 0.0
UNIT 2
CUBOID    3 1 10.0 -10.0 10.0 -10.0 1.25 -1.25
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=4 FILL 3R1 2 END ARRAY
```

Example 1 can be described as a reflected array by treating the square plate as a reflector in the positive z direction. One means of describing this situation is to define Units 1 and 2 as in data description 3, Example 1. The origin of the core boundary is defined to be at the center of the array. The corresponding input geometry is shown below.

Data description 6, Example 1.

```
READ GEOM
UNIT 1
CYLINDER  1 1   5.0   2.0  0.0
CUBOID    0 1   10.0 -10.0 10.0 -10.0 2.0  0.0
UNIT 2
CYLINDER  2 1   5.0   2.0  0.0
CUBOID    0 1   10.0 -10.0 10.0 -10.0 2.0  0.0
CORE      0 1 -10.0 -10.0 -6.0
CUBOID    3 1   10.0 -10.0 10.0 -10.0 8.5 -6.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=6 FILL 1 2 1 2 1 2 END ARRAY
```

The user could have chosen the origin of the core boundary to be at the center bottom of the array. The last two cards would then be:

```
CORE     0 1 -10.0 -10.0 0.0
CUBOID   3 1 10.0 -10.0 10.0 -10.0 14.5 0.0
```

A simpler method of describing Example 1 as a reflected array is to define only one unit as in data description 5, Example 1. The square plate is treated as a reflector as in data description 6, Example 1. The input for this arrangement is given below.

Data description 7, Example 1.

```
READ GEOM
CYLINDER 1 1  5.0  2.0  0.0
CYLINDER 2 1  5.0  4.0  0.0
CUBOID   0 1 10.0 -10.0 10.0 -10.0 4.0 0.0
CORE     0 1 -10.0 -10.0  0.0
CUBOID   3 1 10.0 -10.0 10.0 -10.0 14.5 0.0
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=3 END ARRAY
```

Note that the unit orientation data need not be entered in the array data if only one unit is defined in the geometry region data. Similarly, a unit or box-type definition need not be entered when only one unit is defined.

EXAMPLE 2. Assume the stack of six disks in Example 1 is placed at the center bottom of a cylindrical container composed of material 6 whose inside diameter is 16.0 cm. The bottom and sides of the container are 0.25 cm thick, the top is open and the total height of the container is 18.25 cm. Assume the square plate of Example 1 is centered on top of the container.

The geometry input can be described utilizing most of the data description methods associated with Example 1. One method of describing Example 2 as a single unit is given below.

Data description 1, Example 2.

```
READ GEOM
CYLINDER 1 1 5.0   1.0  -1.0
CYLINDER 2 1 5.0   3.0  -3.0
CYLINDER 1 1 5.0   5.0  -5.0
CYLINDER 2 1 5.0   7.0  -5.0
CYLINDER 0 1 8.0  13.0  -5.0
CYLINDER 6 1 8.25 13.0  -5.25
CUBOID   0 1 10.0 -10.0 10.0 -10.0 13.0 -5.25
CUBOID   3 1 10.0 -10.0 10.0 -10.0 15.5 -5.25
END GEOM
```

In the above description, the origin is defined to be at the center of the disk of material 1 nearest the center of the stack. This disk is defined by the first cylinder description. The disks of material 2 above and below it are defined by the second cylinder description. The disks of material 1 above and below them are defined by the third cylinder description. The top disk of material 2 is defined by the fourth cylinder description. The void interior of the container is defined by the fifth cylinder description. The container is defined by the last cylinder description. The first cuboid card is used to define a void whose x and y dimensions are the same as the square plate, and whose z dimensions are the same as the container. The last cuboid card defines the square plate. Omission of the first cuboid card would result in the container being encased in a solid cuboid of material 3. Thus, both cuboids are necessary to properly define the square plate.

Example 2 can be described as a reflected array. One of the descriptions uses only one unit and is similar to data description 7, example 1. This is shown below. Note that the use of only one unit type allows omission of the unit type definition and the unit orientation data.

Data description 2, Example 2.

```
READ GEOM
CYLINDER 1 1 5.0   2.0  0.0
CYLINDER 2 1 5.0   4.0  0.0
CUBOID   0 1 5.0  -5.0  5.0  -5.0 4.0 0.0
CORE     0 1 -5.0 -5.0  0.0
CYLINDER 0 1 8.0  18.0  0.0
CYLINDER 6 1 8.25 18.0 -0.25
CUBOID   0 1 10.0 -10.0 10.0 -10.0 18.0 -0.25
CUBOID   3 1 10.0 -10.0 10.0 -10.0 20.5 -0.25
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=3 END ARRAY
```

In this data description, the first two cylinder descriptions define a disk of material 1 with a disk of material 2 directly on top of it. A tight fitting void cuboid is placed around them so they can be stacked three high to achieve the stack of disks shown in Example 1, Fig. F11.5.1. This array comprises the core or array portion of the geometry region description. The origin of the core boundary, a tight fitting cube or cuboid that encompasses the core, is defined by the CORE description. Everything after the CORE description is considered part of the reflector. The first cylinder after the CORE defines the void interior of the cylindrical container. The next cylinder defines the walls of the container. The next to last cuboid defines a void volume outside the container from its bottom to its top and having the same x and y dimensions as the square plate. The last cuboid defines the square plate of material 3 that is sitting on top of the container.

Example 2 cannot be described as a reflected array if the inner radius of the container is smaller than 7.071 cm ($\sqrt{2x5.0^2}$). This is the radius at which the inner boundary of the container is tangent to the cuboid around the disks. If the inner radius is smaller than 7.071 cm, the data must be described as a single unit; or alternatively, the container must be included within the units. Assume the container of Example 2 has an inner radius of 6.0 cm with all other data remaining the same. An example of the geometry data describing this configuration is given below in data description 3, Example 2.

Data description 3, Example 2.

```
READ GEOM
UNIT 1
CYLINDER 6 1 6.25  0.25  0.0
CUBOID   0 1 10.0 -10.0  10.0 -10.0 0.25  0.0
UNIT 2
CYLINDER 1 1 5.0   2.0   0.0
CYLINDER 2 1 5.0   4.0   0.0
CYLINDER 0 1 6.0   4.0   0.0
CYLINDER 6 1 6.25  4.0   0.0
CUBOID   0 1 10.0 -10.0  10.0 -10.0 4.0   0.0
UNIT 3
CYLINDER 0 1 6.0   3.0  -3.0
CYLINDER 6 1 6.25  3.0  -3.0
CUBOID   0 1 10.0 -10.0  10.0 -10.0 3.0  -3.0
CUBOID   3 1 10.0 -10.0  10.0 -10.0 5.5  -3.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=5 FILL 1 3R2 3 END ARRAY
```

In the above description, Unit 1 is the bottom of the cylindrical container. The void cuboid is only as tall as the bottom of the container, and its x and y dimensions are the same as the square plate on top of the container. If all the units in the array utilize these same dimensions in the x and y directions, the restriction that adjacent faces of units in contact with each other be the same size and shape is satisfied. This array is stacked in the z direction, so all units must have the same overall dimensions in the x direction and in the y direction. Unit 2 will be used in the array three times to create the stack of disks. It contains a disk of material 1, topped by a disk of material 2. The portion of the container that contains the disks and the cuboid that defines the outer boundaries of the unit are included in Unit 2. Unit 3 describes the empty top portion of the container and the square plate on top of it. The z dimensions of Unit 3 were determined by subtracting three times the total z dimension of Unit 2 from the inside height of the container [18.0 - (3x4.0) = 6.0]. This can also be determined from the overall height of the container by subtracting off the bottom thickness of the container and three times the height of Unit 2 [18.25 - 0.25 - (3x4.0) = 6.0]. The origin of Unit 3 is located at the center of this distance. If the origin were chosen at the bottom of that height, it would be described as:

```
UNIT 3
CYLINDER  0 1  6.0   6.0  0.0
CYLINDER  6 1  6.25  6.0  0.0
CUBOID    0 1 10.0  -10.0 10.0 -10.0 6.0  0.0
CUBOID    3 1 10.0  -10.0 10.0 -10.0 8.25 0.0
```

EXAMPLE 3. Refer to Example 1, Fig. F11.5.1, and imagine a hole 1.5 cm in diameter is drilled along the centerline of the stack through the disks and the square plate. This eliminates the possibility of describing the system as a single unit. This is because the hole in the center of the alternating materials of the stack cannot be described in a manner that allows each successive geometry region to encompass the regions interior to it. Therefore, it must be described as an array. The square plate on the top of the disks is defined as a unit in the array. In the geometry description given below, the square plate is defined in Unit 3.

Data description 1, Example 3.

```
READ GEOM
UNIT 1
CYLINDER  0 1  0.75  2.0  0.0
CYLINDER  1 1  5.0   2.0  0.0
CUBOID    0 1 10.0  -10.0 10.0 -10.0 2.0 0.0
UNIT 2
CYLINDER  0 1  0.75  2.0  0.0
CYLINDER  2 1  5.0   2.0  0.0
CUBOID    0 1 10.0  -10.0 10.0 -10.0 2.0 0.0
UNIT 3
CYLINDER  0 1  0.75  2.5  0.0
CUBOID    3 1 10.0  -10.0 10.0 -10.0 2.5 0.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=7 FILL 1 2 2Q2 3 END FILL END ARRAY
```

In data description 1, Example 3 above, Unit 1 describes a disk of material 1 with a hole through its centerline. The first cylinder defines the hole, the second defines the rest of the disk, and the cuboid defines the size of the unit to be consistent with the square plate so they can be stacked together in the array. Unit 2 describes a disk of material 2 in similar fashion. Unit 3 describes the square plate of material 3 with a hole through its center. The cylinder defines the hole and the cuboid defines the

square plate. These three units are stacked in the z direction to achieve the composite system. This is represented by FILL 1 2 2Q2 3. The 2Q2 repeats the two entries preceding the 2Q2, two times. Alternatively, this can be achieved by entering FILL 1 2 1 2 1 2 3 END FILL. The same array can also be achieved using the LOOP option. An example of the data for this option is: LOOP 1 6R1 1 5 2 2 6R1 2 6 2 3 6R1 7 7 1 END LOOP. Unit 1 is placed at the x=1, y=1 and z=1,3,5 positions of the array by entering 1 6R1 1 5 2. Unit 2 is positioned at the x=1, y=1 and z=2,4,6 positions in the array by entering 2 6R1 2 6 2. Unit 3 is placed at the x=1, y=1, z=7 position of the array by entering 3 6R1 7 7 1. See Sect. F11.4.5 for additional information regarding array specifications. Table F11.4.2 lists other available input options.

EXAMPLE 4. Assume two large cylinders, 2.5 cm in radius and 5 cm long, are connected by a smaller cylinder, 0.5 cm in radius and 10 cm long, as shown in Fig. F11.5.2. All of the cylinders are composed of material 1. By starting the geometry description in the small cylinder, this system can be described as a single unit.

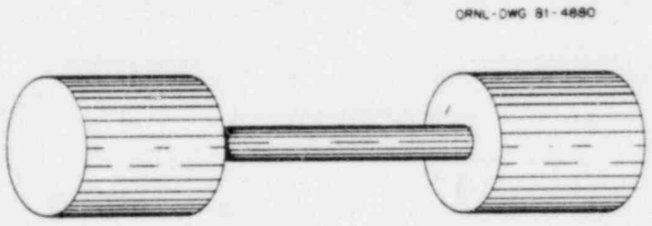


Fig. F11.5.2. Two large cylinders joined axially by a small cylinder

Data description i, Example 4.

```
READ GEOM
CYLINDER  1 1 0.5  5.0  -5.0
CYLINDER  0 1 2.5  5.0  -5.0
CYLINDER  1 1 2.5 10.0 -10.0
END GEOM
```

The origin is at the center of the small cylinder which is described by the first cylinder card. The second cylinder card defines a void cylinder surrounding the small cylinder. Its radius is the same as the large cylinders, and its height coincides with that of the small cylinder. The last cylinder card defines the large cylinders on either end of the small cylinder.

EXAMPLE 5. Assume two large cylinders with a center-to-center spacing of 15 cm, each having a radius of 2.5 cm and length of 5 cm, are connected radially by a small cylinder having a radius of 1.5 cm, as shown in Fig. F11.5.3.

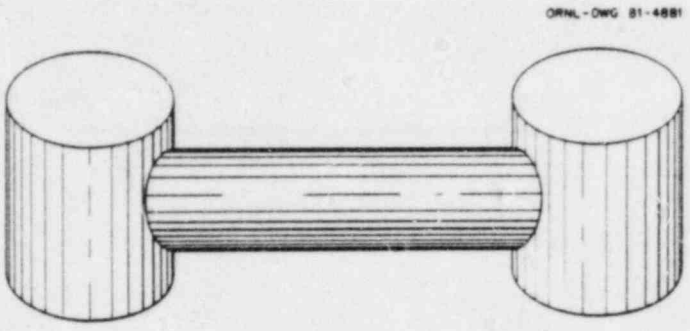


Fig. F11.5.3. Two large cylinders radially connected by a small cylinder

This system cannot be rigorously described in KENO V geometry because the intersection of the cylinders cannot be described. However, it can be approximated two ways, as shown in Fig. F11.5.4. The top approximation is described in data description 1, example 5. The bottom approximation is described in data description 2, example 5, and data description 3, example 5. These may be poor approximations for criticality safety calculations.

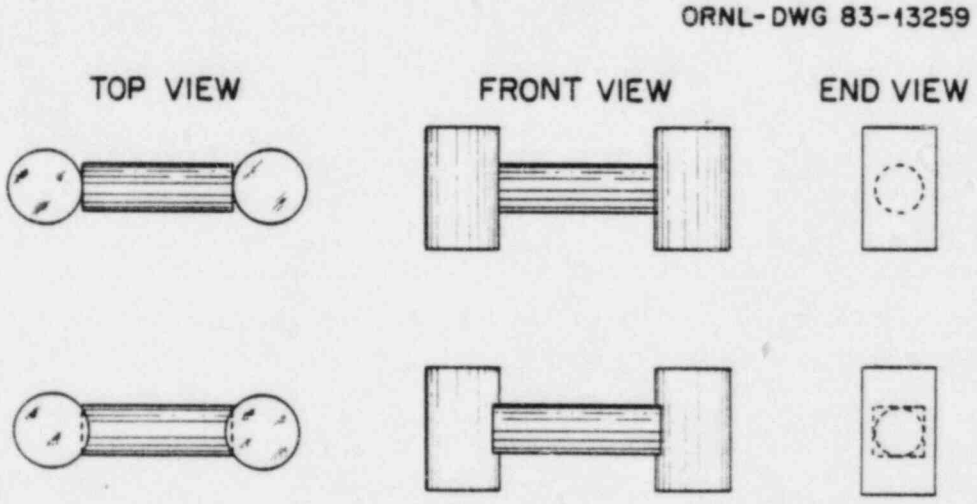


Fig. F11.5.4. KENO V approximations of cylindrical intersections

Data description 1, Example 5.

```
READ GEOM
UNIT 1
CYLINDER  1 1 2.5 2.5 -2.5
CUBE      0 1 2.5 -2.5
UNIT 2
XCYLINDER 1 1 1.5 5.0 -5.0
CUBOID    0 1 5.0 -5.0 2.5 -2.5 2.5 -2.5
END GEOM
READ ARRAY NUX=3 NUY=1 NUZ=1 FILL 1 2 1 END ARRAY
```

Unit 1 defines a large cylinder, and Unit 2 describes the small cylinder. In both units the origin is at the center of the cylinder.

Data description 2, Example 5.

```
READ GEOMETRY
UNIT 1
CYLINDER 1 1 2.5 1.0 0.0
CUBOID 0 1 4P2.5 1.0 0.0
UNIT 2
ZHEMICYCL-X 1 1 2.5 2P1.5 CHORD 2.0
CUBOID 0 1 20 3P-2.5 2P1.5
UNIT 3
ZHEMICYL+X 1 1 2.5 2P1.5 CHORD 2.0
CUBOID 0 1 2.5 -2.0 2P2.5 2P1.5
UNIT 4
XCYLINDER 1 1 1.5 2P5.5
CUBOID 0 1 02P5.5 2P2.5 2P1.5
UNIT 5
CUBOID 0 1 2P5.0 2P2.5 1.0 0.0
UNIT 6
ARRAY 1 3*0.0
UNIT 7
ARRAY 2 3*0.0
END GEOMETRY
READ ARRAY ARA=1 NUX=3 NUY=1 NUZ=1 FILL 1 5 1 END FILL
ARA=2 NUX=3 NUY=1 NUZ=1 FILL 2 4 3 END FILL
ARA=3 NUX=1 NUY=1 NUZ=3 FILL 6 7 6 END FILL
END ARRAY
```

This geometry description uses arrays of arrays (see Section F11.5.6.3) to describe the bottom approximation of Fig. F11.5.4. Unit 1 defines a large cylinder 2.5 cm in radius and 1.0 cm tall inside a close fitting cuboid. This is used in both large cylinders as the portion of the large cylinder that exists both above and below the region where the small cylinder joins it. Unit 5 is the spacing between the tops of the two large cylinders and the spacing between the bottoms of the two large cylinders. Array 1 thus defines the bottom of the system: two short cylinders (Unit 1's) separated by 10 cm (Unit 5 is the separation). Unit 6 contains array 1.

Unit 2 is the left "hemicylinder" that adjoins the horizontal cylinder, and Unit 3 is the right "hemicylinder" that adjoins the horizontal cylinder. Unit 4 defines the horizontal cylinder. Array 2 contains Units 2, 4, and 3, left to right. This defines the central portion of the system where the horizontal cylinder adjoins the two "hemicylinders." These "hemicylinders" are larger than half cylinders. Unit 7 contains array 2.

The entire system is achieved by stacking a Unit 6 above and below the Unit 7 as defined in array 3, the global array.

Data description 3, Example 5.

```
READ GEOMETRY
UNIT 1
CYLINDER 1 1 2.5 1.0 0.0
UNIT 2
CYLINDER 1 1 2.5 1.0 0.0
CUBOID 0 1 17.5 -2.5 2P2.5 1.0 0.0
HOLE 1 15.0 0.0 0.0
UNIT 3
ZHEMICYL-X 1 1 2.5 2P1.5 CHORD 2.0
UNIT 4
ZHEMICYL+X 1 1 2.5 2P1.5 CHORD 2.0
UNIT 5
XCYLINDER 1 1 1.5 2P5.5
CUBOID 0 1  2P10.0 2P2.5 2P1.5
HOLE 3 -7.5 2*0.0
HOLE 4 7.5 2*0.0
END GEOMETRY
READ ARRAY
ARA=1 NUX=1 NUY=1 NUZ=3 FILL 2 5 2 END FILL
END ARRAY
```

This geometry description uses holes (see Sect. F11.5.6.1) to describe the bottom approximation of Fig. F11.4. Unit 1 defines a large cylinder 2.5 cm in radius and 1.0 cm tall. Unit 2 defines the same cylinder within a cuboid that extends from x=-2.5 to x=17.5, from y=-2.5 to y=2.5, and z=0.0 to z=1.0. The origin of the cylinder is at (0.0,0.0,0.0). Thus Unit 2 describes the top and bottom of the cylinder on the left. Unit 1 is positioned within this cuboid as a hole with its origin at (15.0,0.0,0.0) to describe the top and bottom of the cylinder on the right. Unit 3 is the left "hemicylinder" that adjoins the horizontal cylinder, and Unit 4 is the right "hemicylinder" that adjoins the horizontal cylinder. Unit 5 defines the horizontal cylinder with its origin at the center within a cuboid that extends from x=-10.0 to x=+10.0, y=-2.5 to y=2.5, and z=-1.5 to z=1.5. Unit 3 is positioned to the left of the horizontal cylinder, and Unit 4 is positioned to the right of the horizontal cylinder by using holes.

The entire system is achieved by stacking a Unit 2 above and below Unit 5 as shown in the array data.

This same geometry description can be used with Unit 2 redefined to have its origin defined so the unit extends from x=-10 to x=10, y=-2.5 to y=2.5, and z=0.0 to z=1. In this instance, the geometry data would be identical except for Unit 2. This alternate description of Unit 2 is:

```
UNIT 2
CYLINDER 1 1 2.5 1.0 0.0 ORIGIN -7.5 0.0
CUBOID 0 1 2P10.0 2P2.5 1.0 0.0
HOLE 1 7.5 0.0 0.0
```

EXAMPLE 6. Assume 2 small cylinders 1.0 cm in radius and 10 cm long are connected by a large cylinder 2.5 cm in radius and 5 cm long as shown in Fig. F11.5.5.

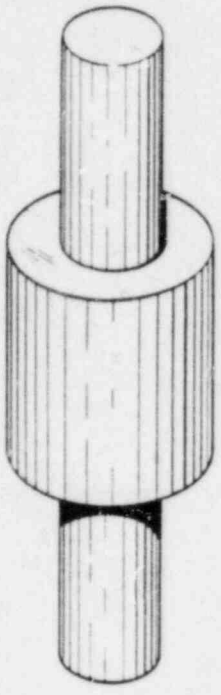


Fig. F11.5.5. Two small cylinders joined axially by a large cylinder

This problem is very similar to example 4, but it cannot be described as a single unit. It must be described as an array. Unit 1 defines the large cylinder, and Unit 2 defines the small cylinder. The origin of each unit is at its center. The composite system consists of two Unit 2's and one Unit 1 as shown below.

Data description 1, Example 6.

```
READ GEOM
UNIT 1
CYLINDER  1 1 2.5 2.5 -2.5
CUBE      0 1 2.5 -2.5
UNIT 2
CYLINDER  1 1 1.0 5.0 -5.0
CUBOID    0 1 2.5 -2.5 2.5 -2.5 5.0 -5.0
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=3 FILL 2 1 2 END ARRAY
```

EXAMPLE 7. Assume an 11x5x3 array of spheres of material 1, radius 3.75 cm, with a center-to-center spacing of 10 cm in the x, y, and z directions. The data for this system are given below.

Data description 1, Example 7.

```
READ GEOM
SPHERE 1 1 3.75
CUBE   0 1 5.0 -5.0
END GEOM
READ ARRAY NUX=11  NUY=5 NUZ=3 END ARRAY
```

EXAMPLE 8. Assume an 11x5x3 array of spheres of material 1 whose radius is 3.75 cm, and whose center-to-center spacing is 10 cm in the x direction, 15 cm in the y direction, and 20 cm in the z direction. The input for this geometry is given below.

Data description 1, Example 8.

```
READ GEOM
SPHERE  1 1 3.75
CUBOID  0 1 5.0  -5.0 7.5 -7.5 10.0 -10.0
END GEOM
READ ARRAY  NUX=11 NUY=5 NUZ=3 END ARRAY
```

EXAMPLE 9. Assume an 11x5x3 array of spheres of material 1 whose radius is 3.75 cm, and whose center-to-center spacing is 10 cm in the x, y, and z directions. This array is reflected by 30 cm of material 2 (water) on all faces. The array spacing defines the perpendicular distance from the outer layer of spheres to the reflector to be 5 cm in the x, y, and z directions. The geometry input for this system is given below.

Data description 1, Example 9.

```
READ GEOM
SPHERE    1 1   3.75
CUBE      0 1   5.0   -5.0
CORE      0 1 -55.0  -25.0  -15.0
REFLECTOR   2 2 6*3.0  10
END ARRAY
READ ARRAY  NUX=11 NUY=5 NUZ=3 END ARRAY
READ BIAS ID=500 2 11 END BIAS
```

The core boundary defines the origin of the reflector to be at the center of the array. The 6*3.0 on the reflector card repeats the 3.0 six times. The reflector card is used to generate ten reflector regions, each 3.0 cm thick, on all six faces of the array. The first bias ID is 2, so the last bias ID will be 11 if 10 regions are created. The biasing data block is necessary to apply the desired weighting or biasing function to the reflector. The biasing material ID is obtained from Table F11.4.5. If the biasing data block is omitted from the problem description, the 10 reflector regions will not have a biasing function applied to them, and the default value of the average weight will be used. This may cause the problem to execute more slowly, and therefore require the use of more computer time.

EXAMPLE 10. Assume the reflector in Example 9 is present only on both x faces, both y faces, and the negative z face. The reflector is only 15.24 cm thick on these faces. The top of the array (positive z face) is unreflected.

Data description 1, Example 10.

```
READ GEOM
SPHERE    1 1   3.75
CUBE      0 1   5.0   -5.0
CORE      0 1 -55.0  -25.0 -15.0
REFLECTOR 2 2 4*3.0    0.0   3.0  5
REFLECTOR 2 7 4*0.24   0.0   0.24 1
READ ARRAY NUX=11 NUY=5 NUZ=3 END ARRAY
READ BIAS ID=500 2 7 END BIAS
```

The first reflector card generates five regions around the array, each region being 3.0 cm thick in the +x, -x, +y, -y, and -z directions, and of zero thickness in the +z direction. This defines a total thickness of 15 cm of reflector material on the appropriate faces. The second reflector card generates the last 0.24 cm of material 2 on those faces. Thus, the total reflector thickness is 15.24 cm on each face of the array, except the top which has no reflector. Five reflector regions were generated by the first reflector card, and one was generated by the second reflector card; so, six biasing regions must be defined in the biasing data. Thus, the beginning bias ID is 2, and the ending bias ID is 7. The biasing material ID and thickness per region are obtained from Table F11.4.5. The thickness per region should be very nearly the thickness per region from the table to avoid overbiasing in the reflector. Partial increments at the outer region of a reflector are exempt from this recommendation. If a biasing function is not to be applied to a region generated by the reflector card, the thickness per region can be any desired thickness. In this instance, a biasing data block is not entered.

EXAMPLE 11. Assume the array of example 7 has the central unit of the array replaced by a cylinder of material 4, 5 cm in radius and 10 cm tall. Assume a 20-cm-thick spherical reflector of material 3 (concrete) is positioned so its inner radius is 65 cm from the center of the array. The minimum inner radius of a spherical reflector for this array is 62.25 cm ($\sqrt{55^2+25^2+15^2}$). If the inner radius is smaller than this, the problem cannot be described using KENO V geometry.

Data description 1, Example 11.

```
READ GEOM
UNIT 1
SPHERE 1 1 3.75
CUBE 0 1 5.0 -5.0
UNIT 2
CYLINDER 4 1 5.0 5.0 -5.0
CUBE 0 1 5.0 -5.0
CORE 0 1 -55.0 -25.0 -15.0
SPHERE 0 1 65.0
REPLICATE 3 2 5.0 4
END GEOM
READ ARRAY NUX=11 NUY=5 NUZ=3 LOOP 1 1 11 1 1 5 1 1 3 1
2 6 6 1 3 3 1 2 2 1 END ARRAY
READ BIAS ID=301 2 5 END BIAS
```

Unit 1 describes the sphere and spacing utilized in the array. Unit 2 defines the cylinder that is located at the center of the array. The CORE defines the origin of the reflector to be at the center of the array. The sphere following the core card defines the inner radius of the reflector. The replicate card will generate four spherical regions of material 3, each 5.0 cm thick. The first 10 entries following the word LOOP fills the 11x5x3 array with Unit 1. The next 10 entries position Unit 2 at the center of the array (x=6, y=3, and z=2), replacing the Unit 1 that had been placed there by the first 10 entries. The biasing data block is used to apply the biasing function for concrete to the generated reflector regions.

EXAMPLE 12. Assume a data profile such as fission densities is desired in a cylinder at 1-cm intervals in the radial direction and 1.5-cm intervals axially. The cylinder, composed of material 1, has a radius of 15 cm and a height of 45 cm. The REPLICATE description can be used to generate these regions as shown below. A biasing data block is not entered because default biasing is desired throughout the cylinder.

Data description 1, Example 12.

```
READ GEOM
CYLINDER  1 1 1.0 1.5  -1.5
REPLICATE 1 2 1.0 2*1.5 14
END GEOM
```

F11.5.6.1 *Use of Holes in the Geometry*

Section F11.5.6 tells how each KENO geometry region in a unit must completely enclose all previously described regions in that unit. Holes can be used to circumvent this restriction to some degree. A HOLE is a means of placing an entire unit within a geometry region. A separate HOLE description is required for every location in a geometry region where a unit is to be placed. The information contained in a hole description is: (1) the geometry word, HOLE, (2) the unit number of the unit to be placed, and (3) the x, y, and z coordinates specifying where the origin of the placed unit is to be located. A hole is placed inside the geometry region that precedes it (excluding holes ... i.e., if a CUBE geometry region is followed by four HOLE descriptions, all four of the HOLES are located within the CUBE.) Holes are subject to the restriction that they cannot intersect any other geometry region. Holes can be nested to any depth (see Sect. F11.5.6.2).

Tracking in regions that contain holes is less efficient than tracking in regions that do not contain holes. Therefore holes should be used only when the system cannot be easily described by conventional methods. One example of the use of holes is shown in Fig. F11.5.6.

The large rods are 1.4 cm in radius and composed of mixture 3. The small rods are 0.6 cm in radius and composed of mixture 1. The inside radius of the annulus is 3.6 cm, and the outside radius is 3.8 cm. The annulus is made of mixture 2. The rods and annulus are both 30 cm long. The annulus is centered in a cuboid having an 8-cm-square cross section and a length of 32 cm. All nonshaded areas are void. In this mock-up, a small rod is defined by Unit 1 and a large rod is defined by Unit 2. Unit 3 defines the central small rod, the annulus and the cuboid. Units 1 and 2 are placed within the annulus using holes.

ORNL-DWG 83-10284

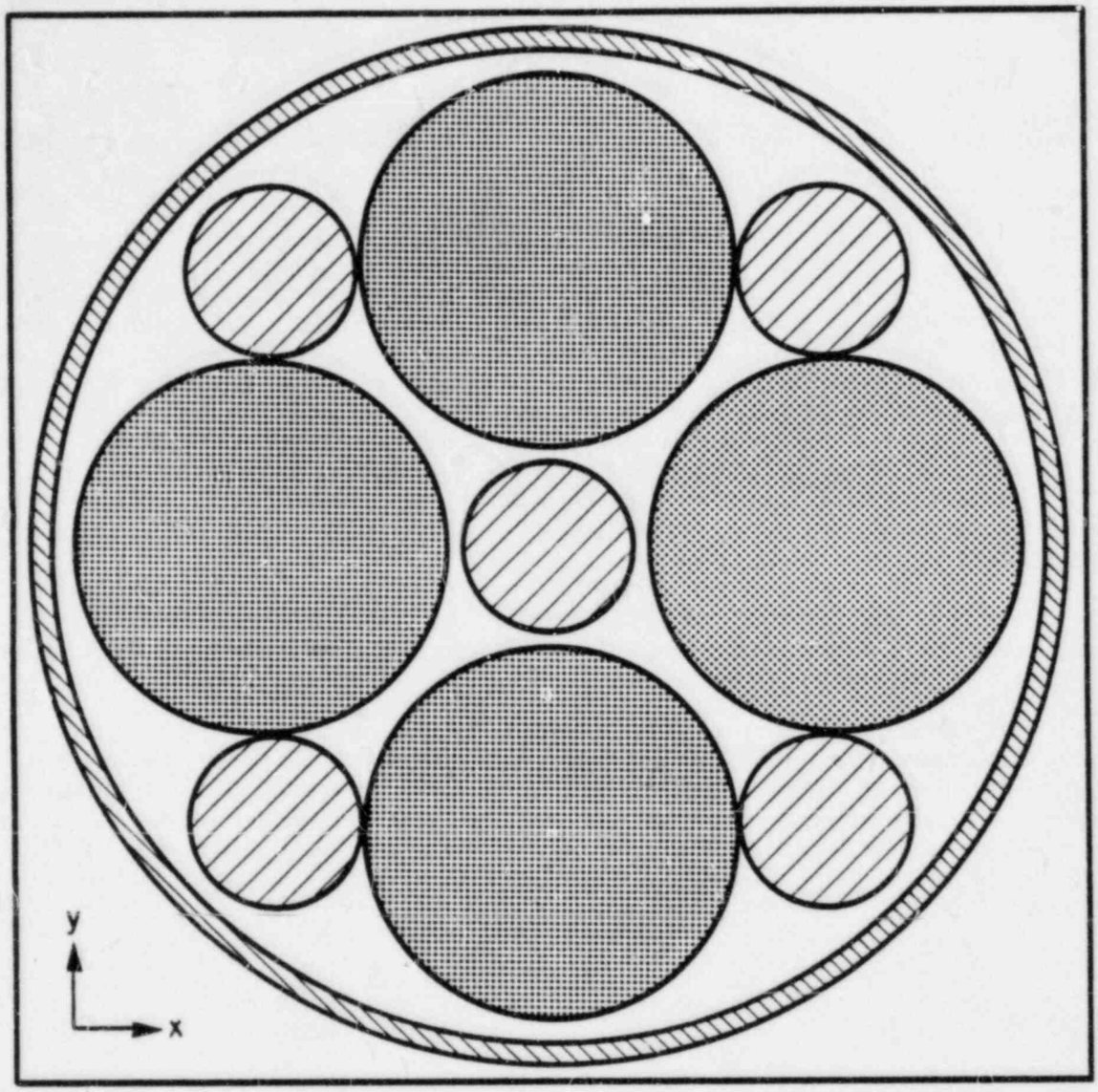


Fig. F11.5.6. Close-packed rods in an annulus

The geometry mock-up can be given by:

```
READ GEOM
UNIT 1
CYLINDER 1 1 0.6 2P15.0
UNIT 2
CYLINDER 3 1 1.4 2P15.0
UNIT 3
CYLINDER 1 1 0.6 2P15.0
CYLINDER 0 1 3.6 2P15.0
HOLE 2 0.0 -2.0 0.0
HOLE 1 2.0 -2.0 0.0
HOLE 2 2.0 0.0 0.0
HOLE 1 2.0 2.0 0.0
HOLE 2 0.0 2.0 0.0
HOLE 1 -2.0 2.0 0.0
HOLE 2 -2.0 0.0 0.0
HOLE 1 -2.0 -2.0 0.0
CYLINDER 2 1 3.8 2P15.0
CUBOID 0 1 4P4.0 2P16.0
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=1 FILL F3 END ARRAY
```

The first HOLE description is for the bottom large rod. It says to take Unit 2 and place its origin at (0.0,-2.0,0.0) in Unit 3. The second HOLE description is for the small rod to the right of the large rod just discussed. It places the origin of Unit 1 at (2.0,-2.0,0.0) in Unit 3. The third HOLE description is for the large rod to the right. It places the origin of Unit 2 at (2.0,0.0,0.0) in Unit 3. This procedure is repeated in a counterclockwise direction until all eight rods have been placed within the region that defines the inner surface of the annulus. The CYLINDER that defines the outer surface of the annulus is described after all the holes for the previous region have been placed. Then the outer cuboid is described.

This example illustrates that a unit that is to be placed using a HOLE description need not have a cube or cuboid as its last region.

An alternate mock-up for this problem would be to omit the first region in Unit 3 and put it in as a hole. The hole description for the central rod would be: HOLE 1 3*0.0.

The order of the HOLE cards in any given region is not important (they can be interchanged with each other randomly). However, they must always appear immediately after the region in which they are placed.

An array of the arrangement shown in Fig. F11.5.6 can be easily described by altering the array description data. For example, a 5x3x2 array of these shapes with a center-to-center spacing of 8 cm in x and y and 32 cm in z can be achieved by utilizing the following array data:

```
READ ARRAY NUX=5 NUY=3 NUZ=2 FILL F3 END FILL END ARRAY
```
or
```
READ ARRAY NUX=5 NUY=3 NUZ=2 FILL 30*3 END FILL END ARRAY
```
or
```
READ ARRAY NUX=5 NUY=3 NUZ=2 LOOP 3 1 5 1 1 3 1 1 2 1 END ARRAY
```

Another example of the use of holes is shown in Fig. F11.5.7.

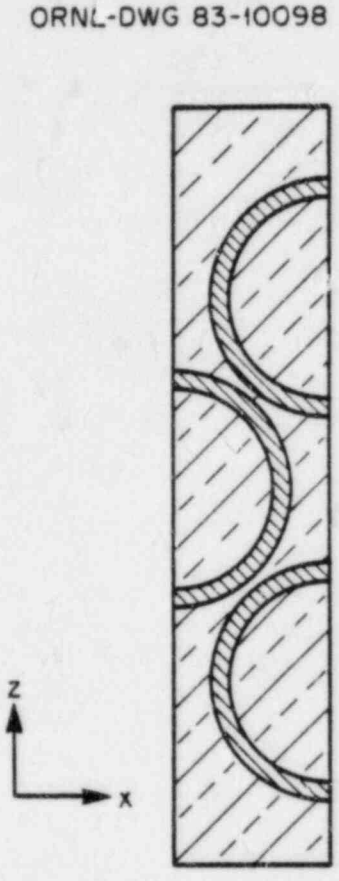


Fig. F11.5.7. Annular rods in triangular pitch lattice

Assume that an infinite linear array of annular rods are stacked three high in a triangular pitch. The array is infinite in the x direction, and the annular rods are stacked three high in the z direction. The rods are finite in the y direction and stacked only one deep in y. The pitch is $\sqrt{13}$. This array can be created by describing the geometrical arrangement shown in Fig. F11.5.7 and applying specular or mirror image reflection on both x faces.

The annular rods are all composed of mixture number one, have an inside radius of 1.3 cm, an outside radius of 1.6 cm, and are 24 cm long. The pitch is $\sqrt{13}$, approximately 3.606 cm. This system can be described in many ways.

This geometry mock-up assumes the origin of the basic unit to be on the left face (-x face), halfway up the cuboid. This is defined as Unit 1. Unit 2 is a hemicylindrical annulus which will be placed along the right face (+x face) using holes.

```
READ GEOM
UNIT 1
YHEMICYL+X  0 1 1.3 2P12.0
YHEMICYL+X  1 1 1.6 2P12.0
CUBOID 0 1 2.0 0.0 2P12.0 2P5.0
HOLE 2 2.0 -3.0 0.0
HOLE 2 2.0 3.0 0.0
UNIT 2
YHEMICYL-X  0 1 1.3 2P12.0
YHEMICYL-X  1 1 1.6 2P12.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=1  FILL F1  END ARRAY
READ BOUNDS  XFC= MIRROR  END BOUNDS
```

In the above geometry description, the hemicylindrical annulus on the left (-x face) is described in Unit 1. A hemicylindrical annulus that is oriented in the opposite direction is described in Unit 2. The first HOLE description locates the origin of Unit 2 at (2.0,-3.0,0.0) in Unit 1. This positions the lower hemicylindrical annulus as shown in Fig. F11.5.7. The second HOLE description locates the origin of Unit 2 at (2.0,3.0,0.0) in Unit 1. This positions the top hemicylindrical annulus as shown in the figure. The two holes can be interchanged without altering the results. The order in which holes are described is not important.

If Units 1 and 2 are interchanged, the geometry mock-up would be as follows:

```
READ GEOM
UNIT 1
YHEMICYL-X  0 1  1.3  2P12.0
YHEMICYL-X  1 0  1.6  2P12.0
UNIT 2
YHEMICYL+X  0 1  1.3  2P12.0
YHEMICYL+X  1 1  1.6  2P12.0
CUBOID     0 1  2.0  0.0  2P12.0  2P5.0
HOLE       1    2.0   -3.0   0.0
HOLE       1    2.0    3.0   0.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=1  FILL F2 END ARRAY
READ BOUNDS  XFC=MIRROR  END BOUNDS
```

In the above mock-up, Unit 2 describes the hemicylindrical annulus on the left and the cuboid exactly as Unit 1 did in the previous geometry description. Unit 1 now describes a hemicylindrical annulus that is oriented in the opposite direction (exactly the same as Unit 2 in the previous description). The first HOLE description places the origin of Unit 1 at (2.0,-3.0,0.0) in Unit 2. The second HOLE description places the origin of Unit 1 at (2.0,3.0,0.0) in Unit 2. This generates the same configuration as shown in Fig. F11.5.7.

Another way of describing this problem uses more unit descriptions. Define the origin of the basic unit to be on the right face at the origin of the lower annulus and call it Unit 3. Define Unit 1 to be a hemicylindrical annulus that has its origin on the left face. Define Unit 2 to be a hemicylindrical annulus that has its origin on the right face. The geometry mock-up for this situation is:

```
READ GEOM
UNIT 1
YHEMICYL+X  0 1 1.3 2P12.0
YHEMICYL+X  1 1 1.6 2P12.0
UNIT 2
YHEMICYL-X  0 1 1.3 2P12.0
YHEMICYL-X  1 1 1.6 2P12.0
UNIT 3
YHEMICYL-X  0 1 1.3 2P12.0
YHEMICYL-X  1 1 1.6 2P12.0
CUBOID  0 1 0.0 -2.0 2P12.0 8.0 -2.0
HOLE  1 -2.0 0.0 3.0
HOLE  2 0.0 0.0 6.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=1  FILL F3 END ARRAY
READ BOUNDS  XFC=MIRROR  END BOUNDS
```

Assume that the origin of the basic unit is at the center of the cuboid and include the lower right hemicylindrical annulus in the basic unit. Define the basic unit to be Unit 3. Define Unit 1 to be the hemicylindrical annulus having its origin on the left face. Define Unit 2 to be the upper hemicylindrical annulus. This mock-up will be the same as the previous one except for changes required by the shift in the location of the origin.

```
READ GEOM
UNIT 1
YHEMICYL+X  0 1 1.3 2P12.0
YHEMICYL+X  1 1 1.6 2P12.0
UNIT 2
YHEMICYL-X  0 1 1.3 2P12.0
YHEMICYL-X  1 1 1.6 2P12.0
UNIT 3
YHEMICYL-X  0 1 1.3 2P12.0 ORIGIN 1.0 -3.0
YHEMICYL-X  1 1 1.6 2P12.0 ORIGIN 1.0 -3.0
CUBOID  0 1 2P1.0 2P12.0 2P5.0
HOLE  1 -1.0 0.0 0.0
HOLE  2 1.0 0.0 3.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=1  FILL F3 END ARRAY
READ BOUNDS  XFC=MIRROR  END BOUNDS
```

Define Unit 1 to be a hemicylindrical annulus oriented like the top and bottom ones in Fig. F11.5.7. Define Unit 2 to be a hemicylindrical annulus oriented like the middle one in Fig. F11.5.7. Define Unit 3 to be the cuboid with the origin at its center. Use holes to place all three hemicylindrical annuli in the cuboid. The geometry mock-up for this situation follows:

```
READ GEOM
UNIT 1
YHEMICYL-X  0 1 1.3 2P12.0
YHEMICYL-X  1 1 1.6 2P12.0
UNIT 2
YHEMICYL+X  0 1 1.3 2P12.0
YHEMICYL+X  1 1 1.6 2P12.0
UNIT 3
CUBOID  0 1 2P1.0 2P12.0 2P5.0
HOLE  1 1.0 0.0 -3.0
HOLE  1 1.0 0.0 3.0
HOLE  2 -1.0 0.0 0.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=1  FILL F3 END ARRAY
READ BOUNDS  XFC=MIRROR  END BOUNDS
```

In the above description, the first HOLE description places the lower annulus at its proper location. The second HOLE description places the upper annulus at its proper position and the third HOLE description places the middle annulus at its correct position.

F11.5.6.2 *Nesting Holes*

This section illustrates how holes are nested. Holes can be nested to any level. Consider the configuration that was illustrated in Fig. F11.5.6 and replace the large rods with a complicated geometric arrangement. The resultant figure is shown in Fig. F11.5.8.

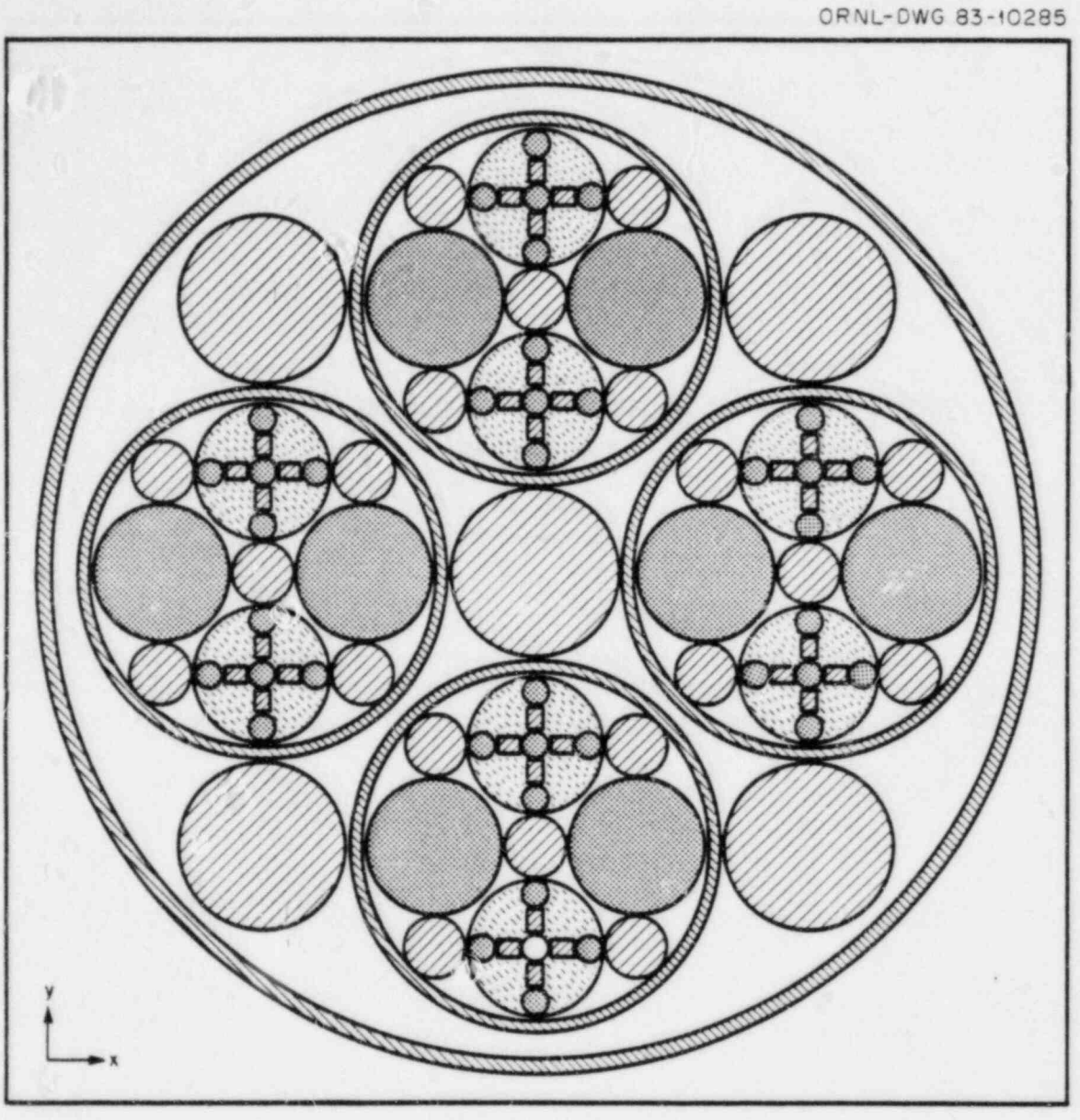


Fig. F11.5.8. Configuration using nested holes

Figure F11.5.9 shows the complicated geometric arrangement that replaced the large rods of Fig. F11.5.6.

ORNL-DWG 83-10283

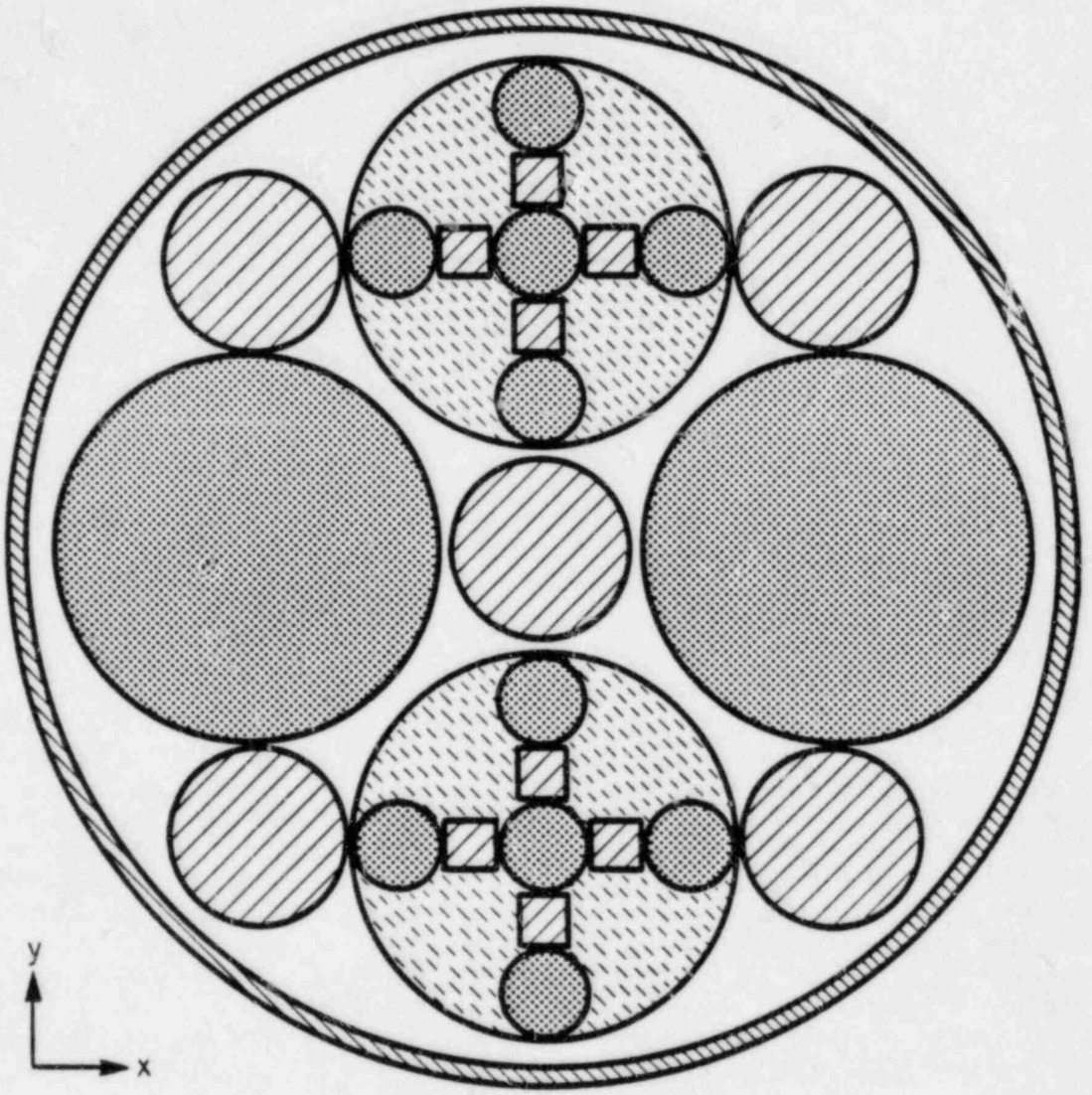


Fig. F11.5.9. Complicated geometric arrangement represented by Unit 7

Figure F11.5.10 shows a component of the arrangement shown in Fig. F11.5.9.

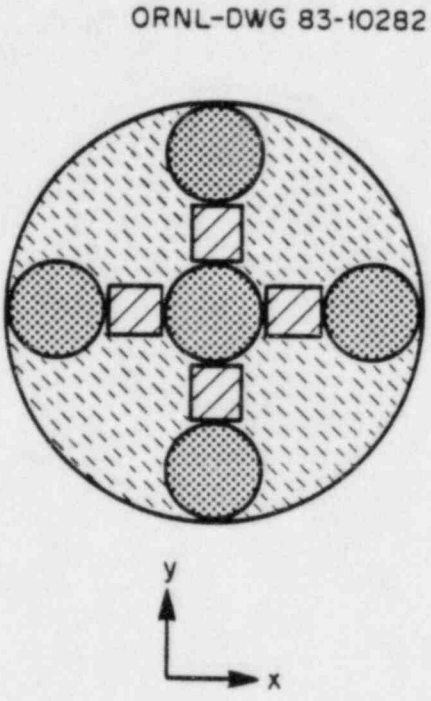


Fig. F11.5.10. Geometric component represented by Unit 4

There is no predetermined "best way" to create a geometry mock-up for a given physical system. The user should decide the order that is most convenient. In order to describe the configuration using nested holes, Fig. F11.5.8, it may be most convenient to start the geometry mock-up at the deepest nesting level as shown in Fig. F11.5.10. The small cylinders are composed of mixture 1, they are each 0.1 cm in radius and 30 cm long. There are five small cylinders used in Fig. F11.5.10. Their centers are located at (0,0,0) for the central one, at (0,-0.4,0) for the bottom one, at (.4,0,0) for the right one, at (0,.4,0) for the top one, and at (-.4,0,0) for the left one. The rectangular parallelepipeds (cuboids) are composed of mixture 2. Each one is 30 cm long and 0.1 cm by 0.2 cm in cross section. The large cylinder containing the configuration is composed of mixture 3, is 30 cm long and has a radius of 0.5 cm. The geometry mock-up for this system is described as follows:

(1) define a small cylinder to be Unit 1,
(2) define a small cuboid with its length in the x direction to be Unit 2,
(3) define a small cuboid with its length in the y direction to be Unit 3,
(4) define Unit 4 to be the large cylinder and place the cylinders and cuboids in it using holes.

```
UNIT 1
CYLINDER 1 1 0.1 2P15.0
UNIT 2
CUBOID 2 1 2P0.1 2P0.05 2P15.0
UNIT 3
CUBOID 2 1 2P0.05 2P0.1 2P15.0
UNIT 4
CYLINDER  1 1 0.1 2P15.0
CYLINDER  3 1 0.5 2P15.0
HOLE 1 0.0 -0.4 0.0
HOLE 1 0.4 0.0 0.0
HOLE 1 0.0 0.4 0.0
HOLE 1 -0.4 0.0 0.0
HOLE 2 -0.2 0.0 0.0
HOLE 2 0.2 0.0 0.0
HOLE 3 0.0 -0.2 0.0
HOLE 3 0.0 0.2 0.0
```

The first cylinder places the central rod,
the **first** HOLE places the bottom cylinder,
the **second** HOLE places the cylinder at the right,
the **third** HOLE places the top cylinder,
the **fourth** HOLE places the cylinder at the left,
the **fifth** HOLE places the left cuboid whose length is in x,
the **sixth** HOLE places the right cuboid whose length is in x,
the **seventh** HOLE places the bottom cuboid whose length is in y, and
the **eighth** HOLE places the top cuboid whose length is in y.

Now that Fig. F11.5.10 has been described, consider Fig. F11.5.9. The large plain cylinders are composed of mixture 1 and are 0.5 cm in radius and 30 cm long. The cylindrical component of Unit 4 is the same size, an outer radius of 0.5 cm and a length of 30 cm. The small cylinders that are located in the interstices between the large cylinders are composed of mixture 2, are 0.2 cm in radius, and are 30 cm long. Define Unit 5 to be the large plain cylinder and Unit 6 to be the small cylinder; Unit 7 is the annulus that contains the cylinders. Its origin is at its center. The annulus is composed of mixture 4, has a 1.3-cm inside radius and a 1.4-cm outer radius. The volume between the inner cylinders is void. The large cylinders each have a radius of 0.5 cm and are tangent. Therefore, their origins are offset from the origin of the unit by 0.7071068. This is from $x^2+y^2=(0.5^2+0.5^2)$ where x and y are equal. The geometry mock-up for this portion of the problem follows:

```
UNIT 5
CYLINDER  1 1 0.5 2P15.0
UNIT 6
CYLINDER  2 1 0.2 2P15.0
UNIT 7
CYLINDER  2 1 0.2 2P15.0
CYLINDER  0 1 1.3 2P15.0
HOLE     5 0.7071068 0.0 0.0
HOLE     6 0.7071068 0.7071068 0.0
HOLE     4 0.0 0.7071068 0.0
HOLE     6 -0.7071068 0.7071068 0.0
HOLE     5 -0.7071068 0.0 0.0
HOLE     6 -0.7071068 -0.7071068 0.0
HOLE     4 0.0 -0.7071068 0.0
HOLE     6 0.7071068 -0.7071068 0.0
CYLINDER  4 1 1.4 2P15.0
```

In Unit 7, the first cylinder description defines the central rod and the second cylinder defines the void volume in which the smaller cylinders are to be placed.

The **first** HOLE places the larger cylinder of mixture 1 at the right with its origin at (0.7071068,0.0,0.0),
the **second** HOLE places the small cylinder in the upper right quadrant,
the **third** HOLE places the top cylinder that contains the geometric component defined in Unit 4,
the **fourth** HOLE places the small cylinder in the upper left quadrant,
the **fifth** HOLE places the larger cylinder of mixture 1 at the left,
the **sixth** HOLE places the small cylinder in the lower left quadrant,
the **seventh** HOLE places the bottom cylinder that contains the geometric component defined in Unit 4,
and the **eighth** HOLE places the small cylinder in the lower right quadrant.
The last cylinder defines the outer surface of the annulus.

To complete the geometry mock-up, consider Fig. F11.5.8.

Define Unit 8 to be the cylinder of mixture 2 having a radius of 0.6 cm and a length of 30 cm.

Define Unit 9 to be the central rod and the large annulus centered in a cuboid having an 8-cm-square cross section and being 32 cm long.

```
UNIT 8
CYLINDER  2 1 0.6 2P15.0
UNIT 9
CYLINDER  2 1 0.6 2P15.0
CYLINDER  0 1 3.6 2P15
HOLE   7 2.0 0.0 0.0
HOLE   8 2*2.0 0.0
HOLE   7 0.0 2.0 0.0
HOLE   8 -2.0 2.0 0.0
HOLE   7 -2.0 2*0.0
HOLE   8 2*-2.0 0.0
HOLE   7 0.0 -2.0 0.0
HOLE   8 2P2.0 0.0
CYLINDER  4 1 3.8 2P15.0
CUBOID    0 1 4P4.0 2P16.0
```

In Unit 9, the first cylinder defines the rod of mixture 2, centered in the annulus.
The second cylinder defines the void volume between the central rod and the annulus.
The **first** HOLE places the composite annulus of Unit 7 to the right of the central rod,
the **second** HOLE places a rod defined by Unit 8 in the upper right quadrant of the annulus,
the **third** HOLE places the composite annulus of Unit 7 above the central rod,
the **fourth** HOLE places a rod defined by Unit 8 in the upper left quadrant of the annulus,
the **fifth** HOLE places the composite annulus of Unit 7 to the left of the central rod,
the **sixth** HOLE places a rod defined by Unit 8 in the lower left quadrant,
the **seventh** HOLE places the composite annulus of Unit 7 below the central rod,
and the **eighth** HOLE places a rod defined by Unit 8 in the lower right quadrant.
The last cylinder defines the outer surface of the annulus. The outer cuboid is the last region.

This problem illustrates three levels of hole nesting. The total input data for the problem is given below. The nuclide ID's are for the 16-group Hansen Roach working format library. The mixtures used in this problem are not realistic or meaningful. However, the geometry description accurately recreates the geometry arrangement of Fig. F11.5.8. This problem includes the data for a printer plot to be used to verify the validity of the geometry description. The plot data specify a picture that is 260 characters wide, so the picture is generated in two pieces. The left half of the printer plot is shown in Fig. F11.5.11 and the right half is given in Fig. F11.5.12. The user can tape the two halves together. If the plot were specified to be 130 characters wide, it would all print in one piece. However, some of the detail might have been lost.

```
NESTED HOLES SAMPLE
READ PARAM  RUN=NO LIB=41 TME=0.5 END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2 MIX=2 200 1.0 MIX=3 502 0.1
MIX=4 200 1.0
END MIXT
READ GEOM
UNIT 1
CYLINDER 1 1 0.1 2P15.0
UNIT 2
CUBOID 2 1 2P0.1 2P0.05 2P15.0
UNIT 3
CUBOID 2 1 2P0.05 2P0.1 2P15.0
UNIT 4
CYLINDER 1 1 0.1 2P15.0
CYLINDER 3 1 0.5 2P15.0
HOLE 1  0.0 -0.4 0.0
HOLE 1  0.4  0.0 0.0
HOLE 1  0.0  0.4 0.0
HOLE 1 -0.4  0.0 0.0
HOLE 2 -0.2  0.0 0.0
HOLE 2  0.2  0.0 0.0
HOLE 3  0.0 -0.2 0.0
HOLE 3  0.0  0.2 0.0
UNIT 5
CYLINDER 1 1 0.5 2P15.0
UNIT 6
CYLINDER 2 1 0.2 2P15.0
UNIT 7
CYLINDER 2 1 0.2 2P15.0
CYLINDER 0 1 1.3 2P15.0
HOLE 5 0.7071068 2*0.0
HOLE 6 0.7071068 0.7071068 0.0
HOLE 4 0.0 0.7071068 0.0
HOLE 6 -0.7071068 0.7071068 0.0
HOLE 5 -0.7071068 0.0 0.0
HOLE 6 -0.7071068 -0.7071068 0.0
HOLE 4 0.0 -0.7071068 0.0
HOLE 6 0.7071068 -0.7071068 0.0
CYLINDER 4 1 1.4 2P15.0
UNIT 8
CYLINDER 2 1 0.6 2P15.0
UNIT 9
CYLINDER 2 1 0.6 2P15.0
CYLINDER 0 1 3.6 2P15.0
HOLE 7 2.0 0.0 0.0
HOLE 8 2*2.0 0.0
HOLE 7 0.0 2.0 0.0
HOLE 8 -2.0 2.0 0.0
HOLE 7 -2.0 2*0.0
HOLE 8 2*-2.0 0.0
HOLE 7 0.0 -2.0 0.0
HOLE 8 2P2.0 0.0
CYLINDER 4 1 3.8 2P15.0
CUBOID 0 1 4P4.0 2P16.0
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=1  FILL 9 END ARRAY
READ PLOT  TTL='X-Y SLICE AT Z MIDPOINT.  NESTED HOLES'
XUL=-0.1  YUL=8.1 ZUL=16.0 XLR=8.1 YLR=-0.1 ZLR=16
UAX=1.0 VDN=-1.0  NAX=260 NCH=' *-.X'  END PLOT
END DATA
END
```

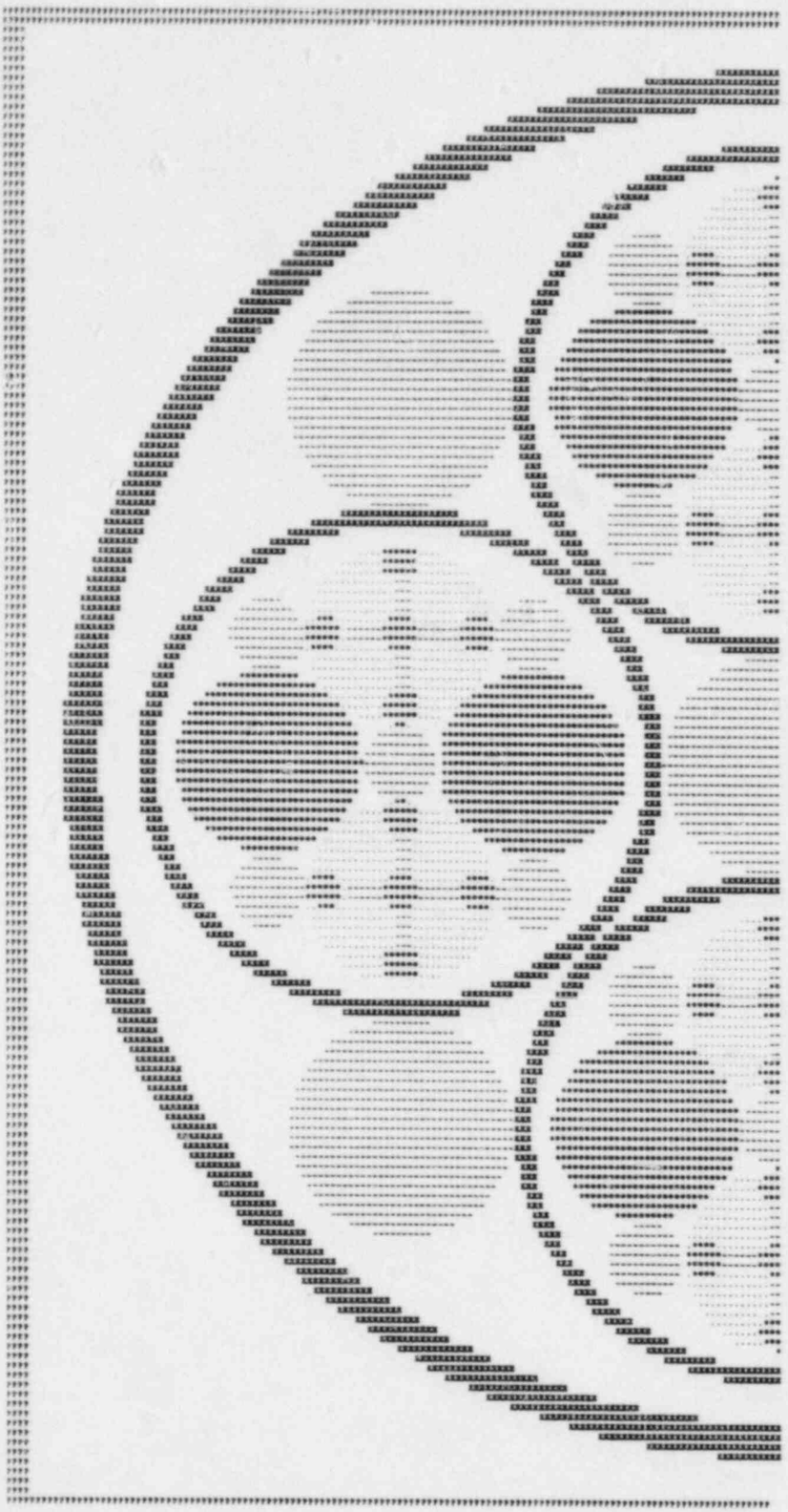


Fig. F11.5.11a. Left half of printer plot

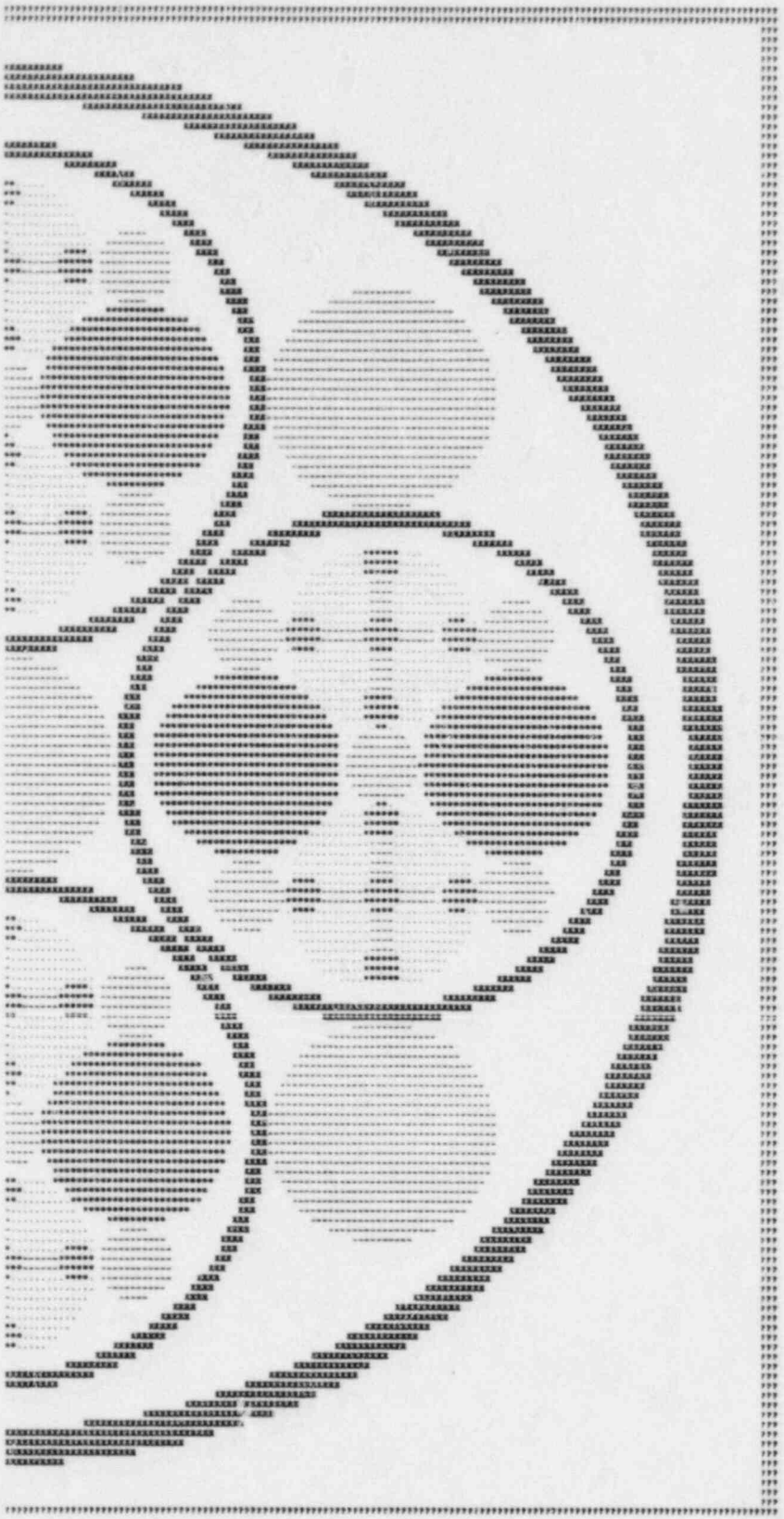


Fig. F11.5.11b. Right half of printer plot

F11.5.6.3 *Multiple Arrays*

Section F11.5.6 demonstrates how units are composed of geometry regions and how these units can be stacked in an array. This same procedure can be extended to create multiple arrays. Furthermore, arrays can be used as building blocks within other arrays.

Consider Sample Problem 12 from Sect. F11.D. The description of this sample problem is restated below as Sample Problem 19.

This problem is a critical experiment consisting of a composite array[1,3] of four highly enriched uranium metal cylinders and four cylindrical Plexiglas containers filled with uranyl nitrate solution. The metal units in this experiment are designated in Table II of ref. 1 as cylinder index 11 and reflector index 1. A photograph of the experiment is given in Fig. F11.D.3. The coordinate system is defined to be z up the page, y across the page, and x out of the page.

The Plexiglas containers have an inside radius of 9.525 cm and an outside radius of 10.16 cm. The inside height is 17.78 cm and the outside height is 19.05 cm. Four of these containers are stacked with a center-to-center spacing of 21.75 cm in the "y" direction and 20.48 cm in the "z" direction (vertical). This arrangement of four Plexiglas containers can be described as follows: mixture 2 is the uranyl nitrate and mixture 3 is Plexiglas, so the Plexiglas container with its appropriate spacing cuboid can be described as Unit 1. This considers the array to be bare and suspended with no supports.

```
UNIT 1
CYLINDER  2 1 9.525  2P8.89
CYLINDER  3 1 10.16  2P9.525
CUBOID    0 1 4P10.875  2P10.24
```

The array of four Plexiglas containers can be described as array 1 in the array data as follows:

```
ARA=1  NUX=1  NUY=2  NUZ=2  FILL  F1  END FILL
```

The four metal cylinders each have a radius of 5.748 cm and are 10.765 cm tall. They have a center-to-center spacing of 13.18 cm in the "y" direction and 12.45 cm in the "z" direction (vertical). Thus, one of the metal cylinders with its appropriate spacing cuboid can be described as Unit 2. This array is also considered to be bare and unsupported.

```
UNIT 2
CYLINDER 1 1  5.748 2P5.3825
CUBOID   0 1  4P6.59 2P6.225
```

The array of four metal cylinders can be described as array 2 in the array data.

```
ARA=2  NUX=1  NUY=2  NUZ=2  FILL  F2  END FILL
```

Now two arrays have been described. The overall dimensions of the array of Plexiglas containers is 21.75 cm in x, 43.5 cm in y, and 40.96 cm in z. The overall dimensions of the array of metal cylinders is 13.18 cm in x, 26.36 cm in y, and 24.9 cm in z.

In order to describe the composite array, these two arrays must be stacked together into an array. In order for them to be stacked into an array, the adjacent faces must match. This is accomplished by defining a Unit 3 which contains array 1, the array of Plexiglas solution containers. The overall dimensions of this unit are 43.5 cm in x and y and 40.96 cm in z. These dimensions are calculated by the code and need not be specified. Unit 3 is defined as follows:

```
UNIT 3
ARRAY 1  3*0.0
```

The array of metal cylinders will be defined to be Unit 4. However, this array is smaller in the y and z dimensions than the array of Plexiglas units. Therefore, a void region must be placed around the array in those directions so Unit 4 and Unit 3 will be the same size in y and z. This is accomplished by using a reflector card as follows:

```
UNIT 4
ARRAY 2 3*0.0
REPLICATE 0 1 2*0.0 2*8.57 2*8.03 1
```

Now that Unit 3 and Unit 4 have been defined, they must be placed in the global or universe array. This is done in the array data and the global array number is set to 3 as follows:

```
GBL=3 ARA=3 NUX=2 NUY=1 NUZ=1 FILL 4 3 END FILL
```

This completes the geometry description for the problem. The complete input description for the problem is given below. The nuclide ID's are for the 16-group Hansen Roach working format library.

```
=KENO5
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ PARAM  LIB=41  RUN=NO END PARAM
READ MIXT SCT=1  MIX=1  92860 3.2275-3  92501 4.4802-2  MIX=2  1102 5.81-2
7100 1.9753-3  8100 3.6927-2  92501 9.8471-4  92860 7.7697-5
MIX=3  6100 3.5552-2  1102 5.6884-2  8100 1.4221-2  END MIXT
READ GEOM
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
UNIT 2
CYLINDER  1 1 5.748 2P5.3825
CUBOID  0 1 4P6.59 2P5.3825
UNIT 3
ARRAY 1 3*0.0
UNIT 4
ARRAY 2 3*0.0
REPLICATE 0 1 2*0.0 2*8.57 2*8.03 1
END GEOM
READ ARRAY  ARA=1 NUX=1 NUY=2 NUZ=2  FILL F1 END FILL
ARA=2 NUX=1 NUY=2 NUZ=2 FILL F2 END FILL GBL=3 ARA=3 NUX=2 NUY=1 NUZ=1
FILL 4 3 END FILL
END ARRAY
READ PLOT TTL='X-Y SLICE AT Z=10.24'
XUL=-1.0 YUL=44.5 ZUL=10.24 XLR=35.93 YLR=-1.0 ZLR=10.24
UAX=1.0 VDN =-1.0 NAX =130 NCH=' *.-' PIC=MIX END
TTL='X-Z SLICE AT Y=10.875'
XUL=-1.0 YUL=10.875 ZUL=41.96 XLR=35.93 YLR=10.875 ZLR=-1.0
UAX=1.0 WDN=-1.0 PIC=MIX END  END PLOT
END DATA
END
```

A printer plot of an x-y slice taken through the bottom layer of the array is shown in Fig. F11.5.12. A printer plot of an x-z slice taken through the left half of the array is shown in Fig. F11.5.13. These printer plots were used to verify the geometry mock-up.

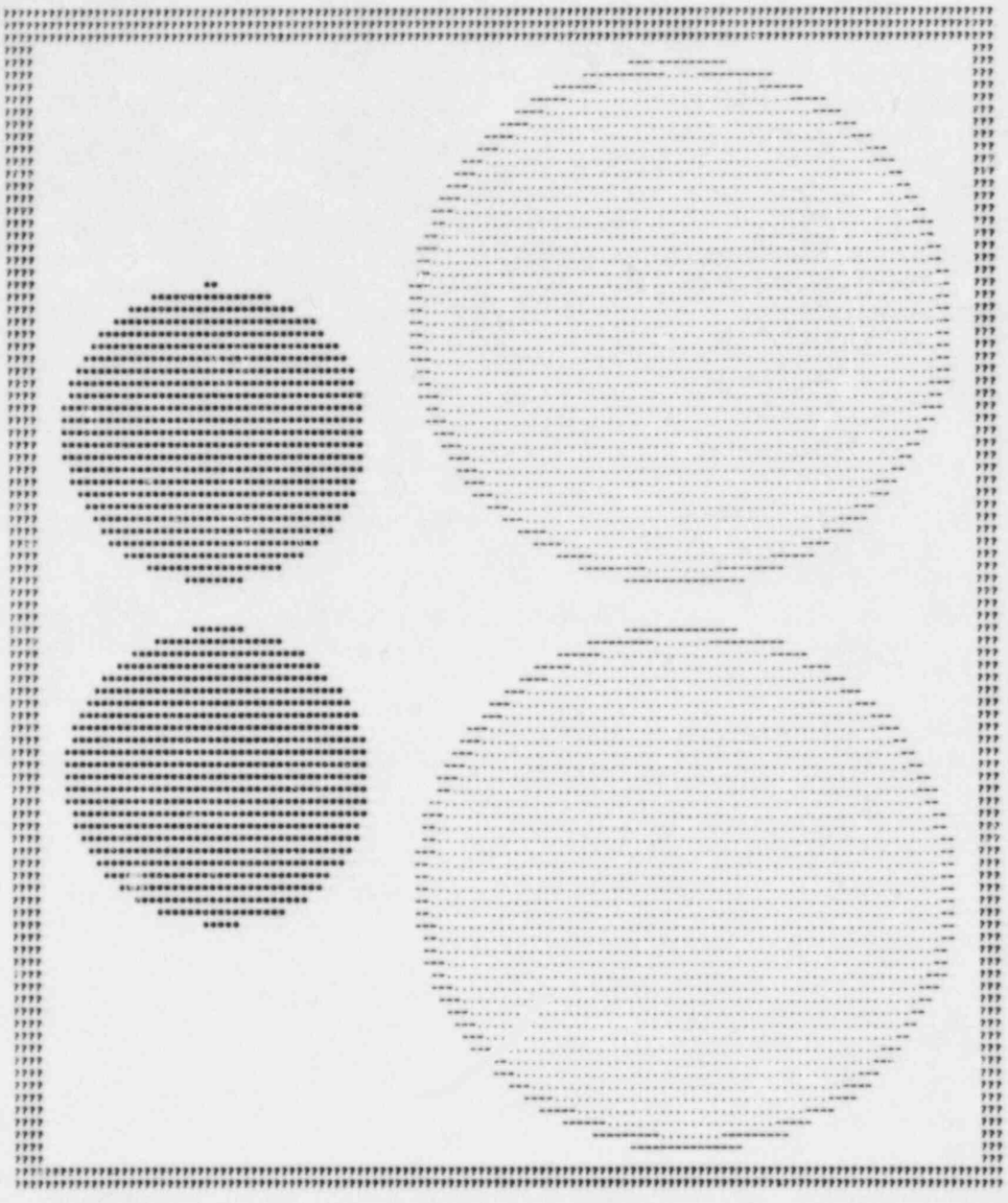


Fig. F11.5.12. x-y plot of mixed array

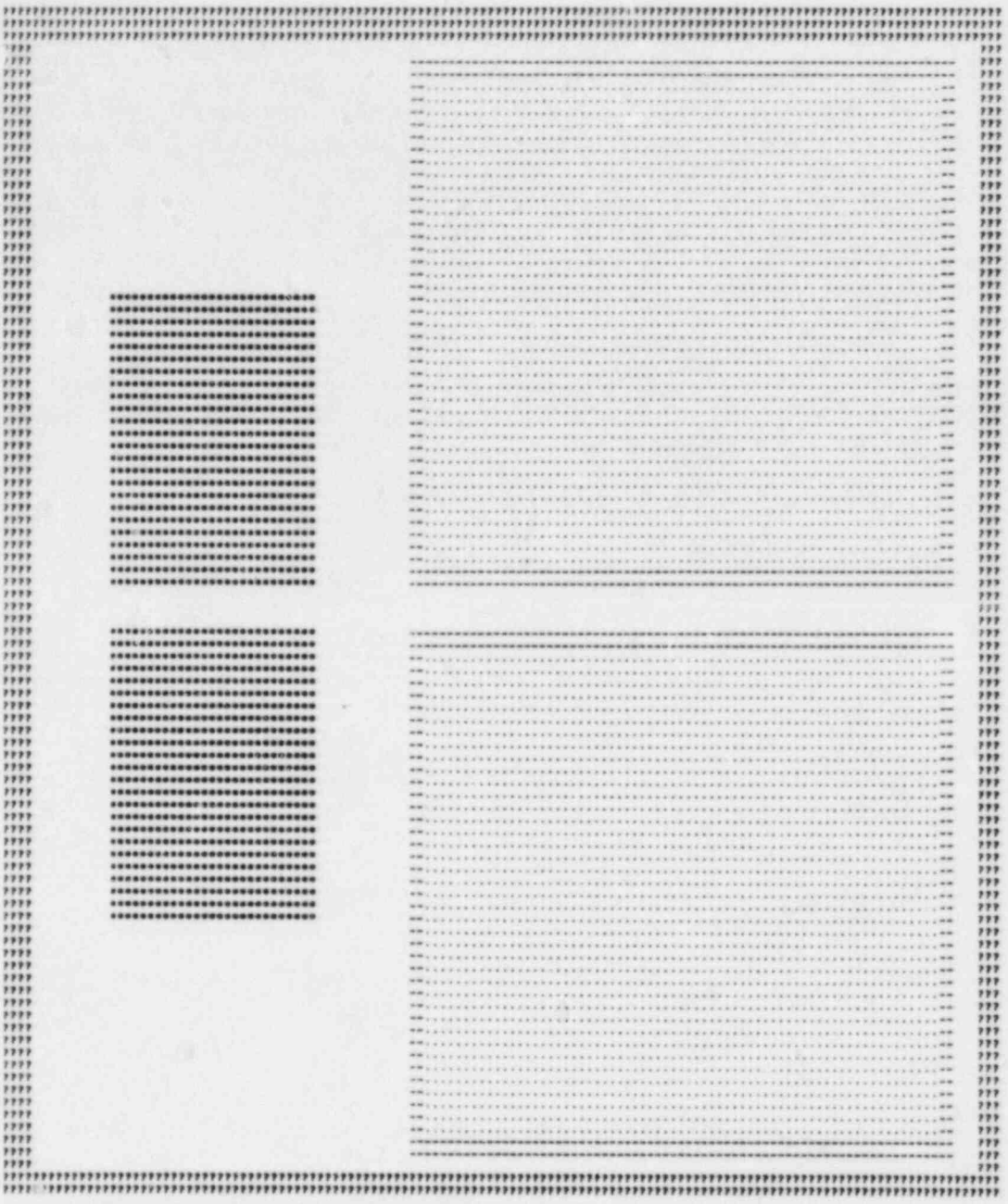


Fig. F11.5.13. x-z plot of mixed array

STORAGE ARRAY

Consider a storage array of highly enriched uranium buttons, each 1 in. tall and 4 in. in diameter. These buttons are stored on stainless steel shelves with a center-to-center spacing of 2 ft between them. The shelves are 1/4 in. thick, 18 in. wide, 20 ft long, and are 18 in. from the top of a shelf to the bottom of the shelf above it. Each rack of storage shelves is four shelves high, with the first shelf being 6 in. above the floor. The storage room is 19.5 ft in the x direction by 43 ft in the y direction with 12-ft ceilings in the z direction. The walls, ceiling, and floor are composed of concrete, 1 ft thick. All the aisles between the storage racks are 3 ft wide. The racks are arranged with their length in the y direction and an aisle between them. The array of racks are arranged with two in the y direction and five in the x direction. Mixture 1 is the uranium metal, mixture 2 is the stainless steel, and mixture 3 is the concrete.

First, describe the metal button and its center-to-center spacing. The void vertical spacing has arbitrarily been chosen to extend from the bottom of the button to the next shelf above the button. The shelf of stainless steel is described under the button.

```
UNIT 1
CYLINDER  1 1  5.08  2.54  0.0
CUBOID    0 1  2P22.86  2P30.48  45.72  0.0
CUBOID    2 1  2P22.86  2P30.48  45.72  -.635
```

Array 1 creates an array of these buttons that fills one shelf. Unit 2 then contains one of the shelves shown in Fig. F11.5.14.

```
ARA=1  NUX=1  NUY=10  NUZ=1  FILL  F1  END FILL

UNIT 2
ARRAY 1  3*0.0
```

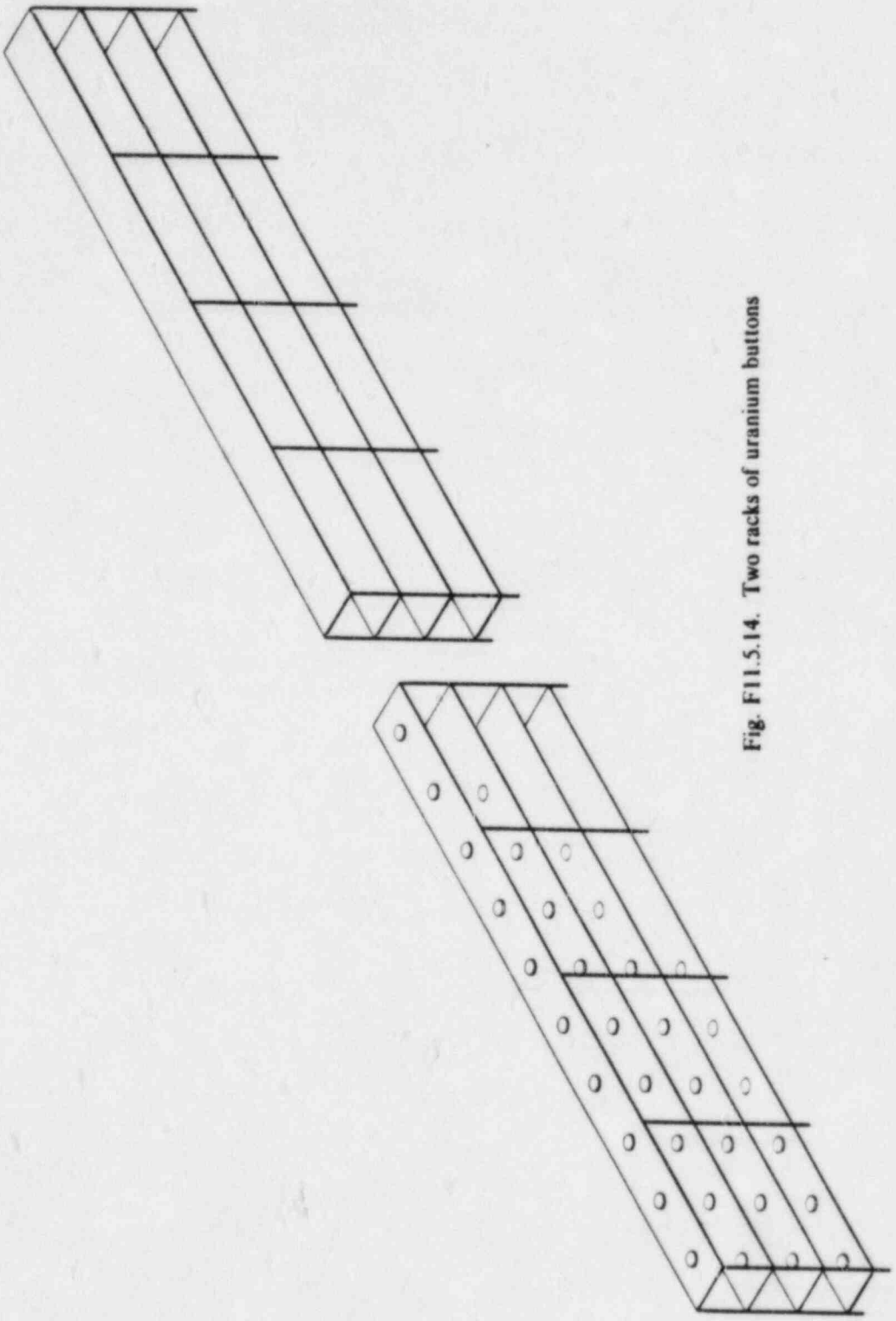
Stack four Unit 2's vertically to obtain one of the racks shown in Fig. F11.5.14. One rack is defined by array 2.

```
ARA=2  NUX=1  NUY=1  NUZ=4  FILL  F2  END FILL
```

Generate a Unit 3 that contains a rack of shelves and a Unit 4 that is the aisle between the ends of the two racks in the y direction.

```
UNIT 3
ARRAY 2 3*0.0
UNIT 4
CUBOID  0 1  2P22.86  2P45.72  185.42  0.0
```

Stack Units 3 and 4 together in the y direction to create Unit 5 which contains both racks in the y direction and the aisle between them. This is the configuration shown in Fig. F11.5.14.



ORNL–DWG 83-10099

Fig. F11.5.14. Two racks of uranium buttons

```
ARA=3  NUX=1  NUY=3  NUZ=1  FILL  3 4 3  END FILL

UNIT 5
ARRAY 3  3*0.0
```

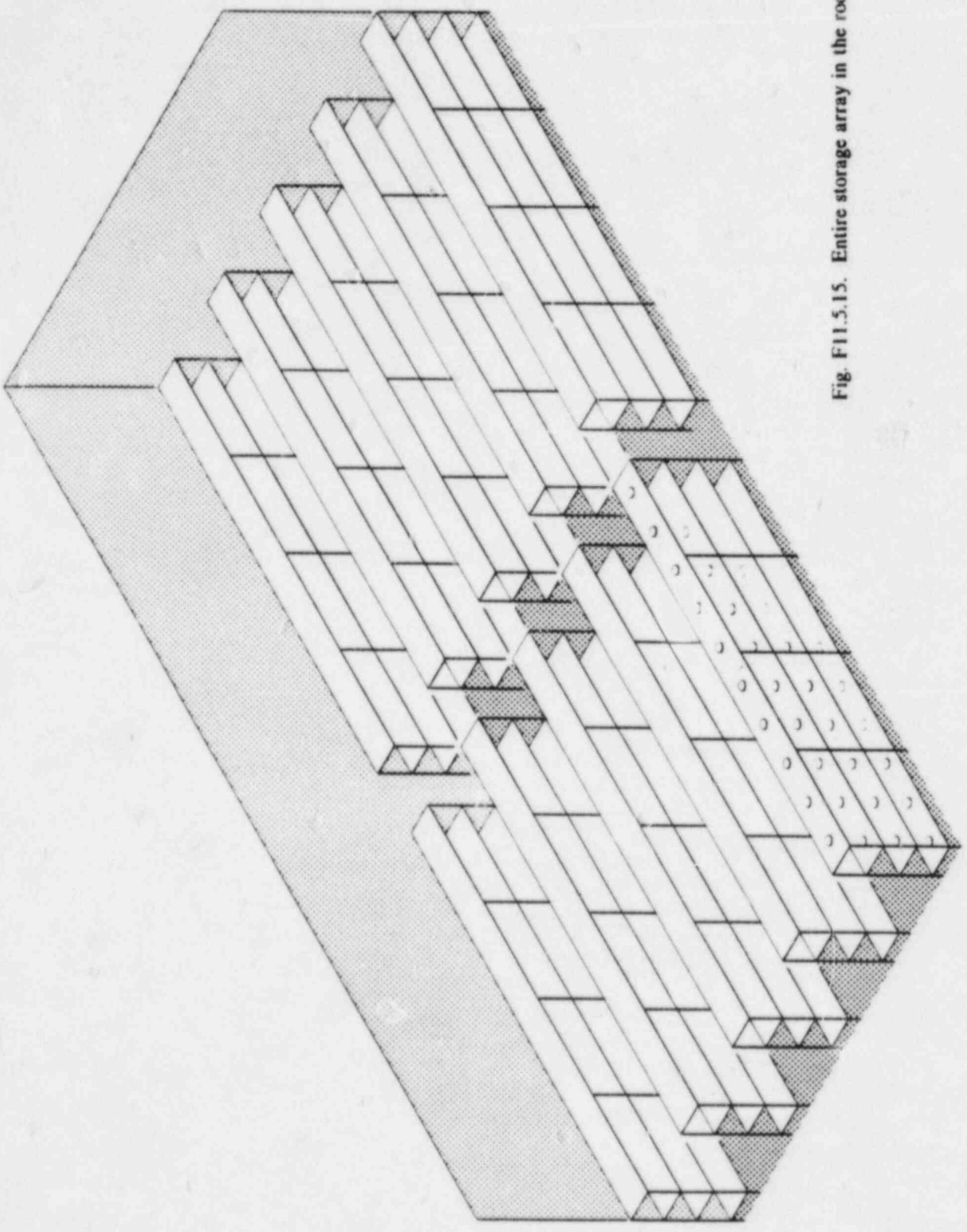
Create a Unit 6 which is an aisle 3 ft wide in the x direction and 43 ft in the y direction (full length of the room).

```
UNIT 6
CUBOID  0 1  91.44 0.0  1310.64 0.0  185.42 0.0
```

Stack Units 5 and 6 in the x direction to achieve the array of racks in the room. Then put the 6-in. spacing below the bottom of the racks, the spacing between the top of the top rack and the ceiling, and add the concrete floor, walls, and ceiling around the array. Array 4 describes the array of racks in the room. The core description encompasses this array, and the first reflector descriptions are used to add the spacing between the top rack and the ceiling. The last two reflector descriptions add the ceiling, walls and floor. A perspective of the room is shown in Fig. F11.5.15.

```
GBL=4  ARA=4  NUX=9  NUY=1  NUZ=1  FILL 5 6  3Q2  5  END FILL

CORE   4 1 3*0.0
REPLICATE 0 1  4*0.0  165.1  15.24  1
REPLICATE 3 2  6*5.0   6
REPLICATE 3 8  6*0.48  1
```

The final mock-up for this room is given below: The printer plots for this problem must be quite large in order to see all the detail because the array is sparse and the shelves are thin. Therefore, the printer plots for this system are not included as figures. The user can generate the printer plots if it is desirable to see them. The first two plots are two pages wide and the last one is only one page wide. The nuclide ID's used in this problem are for the 16-group Hansen Roach working format library.



ORNL-DWG 83 10100

Fig. F11.5.15. Entire storage array in the room

```
=KENO5
STORAGE ARRAY
READ PARAMETERS  TME=1.0 FDN=YES LIB=41
END PARAMETERS
READ MIXT  SCT=1  MIX=1 92500 4.48006-2  92800 2.6578-3  92400 4.827-4
92600 9.57-5  MIX=2 200 1.0 MIX=3 301 1 END MIXT
READ GEOMETRY
UNIT 1
CYLINDER 1 1 5.08 2.54 0.0
CUBOID  0 1 2P22.86 2P30.48 45.72 0.0
CUBOID  2 1 2P22.86 2P30.48 45.72 -0.635
UNIT 2
ARRAY 1 3*0.0
UNIT 3
ARRAY 2 3*0.0
UNIT 4
CUBOID 0 1 2P22.86 2P45.72 185.42 0.0
UNIT 5
ARRAY 3 3*0.0
UNIT 6
CUBOID 0 1 91.44 0.0 1310.64 0.0 185.42 0.0
CORE 4 1 3*0.0
REPLICATE 0 1 [illegible]0.0 165.1 15.24 1
REPLICATE 3 2 6*5.0 6
REPLICATE 3 8 6*0.48 1
END GEOMETRY
READ ARRAY
ARA=1 NUX=1 NUY=10 NUZ=1 FILL F1 END FILL
ARA=2 NUX=1 NUY=1 NUZ=4 FILL F2 END FILL
ARA=3 NUX=1 NUY=3 NUZ=1 FILL 3 4 3 END FILL
GBL=4 ARA=4 NUX=9 NUY=1 NUZ=1 FILL 5 6 3Q2 5 END FILL
END ARRAY
READ BIAS ID=301 2 8 END BIAS
READ START NST=4 TFX=0.0 TFY=0.0 TFZ=0.0 NBX=5 END START
READ PLT= NO TTL='X-Z SLICE AT Y=30.48 WITH Z ACROSS AND X DOWN'
XUL=594.8 YUL=30.48 ZUL=-1.0 XLR=-0.5 YLR=30.48 ZLR=186.0
WAX=1.0 UDN=-1.0 NAX=260 NCH=' *#.' END
TTL='Y-Z SLICE OF LEFT RACKS, X=22.86 WITH Z ACROSS AND Y DOWN'
XUL=22.86 YUL=1311.0 ZUL=-0.5 XLR=22.86 YLR=-3.0 ZLR=186.0
WAX=1.0 VDN=-1.0 NAX=260 END
TTL='X-Y SLICE OF ROOM THROUGH SHELF Z=0.3175 WITH X ACROSS AND Y DOWN'
XUL=-1.0 YUL=1312.0 ZUL=0.3175 XLR=596.0 YLR=-2.5 ZLR=0.3175
UAX=1.0 VDN=-1.0 NAX=130 END
END PLOT
END DATA
END
```

F11.5.6.4 *Arrays and Holes*

Sections F11.5.6.1 and 2 describe the use of holes and Sect. F11.5.6.3 describes multiple arrays and arrays of arrays. Holes can be used to place arrays at locations that would have been impossible with earlier geometry restrictions. This section contains examples to illustrate the combined use of arrays and holes.

EXAMPLE 1 A SIMPLE CASK

Consider a cylindrical mild steel container having an inside radius of 4.15 cm and a radial wall thickness of 0.45 cm. The thickness of the ends of the container is 1.27 cm and the inside height is 10.1 cm. Highly enriched uranium rods 1 cm in diameter and 10 cm long are banded together into square bundles of four. These bundles are then positioned in the mild steel container as shown in Fig. F11.5.16. The rods sit on the floor of the container and have a 0.1-cm gap between their tops and the top of the container.

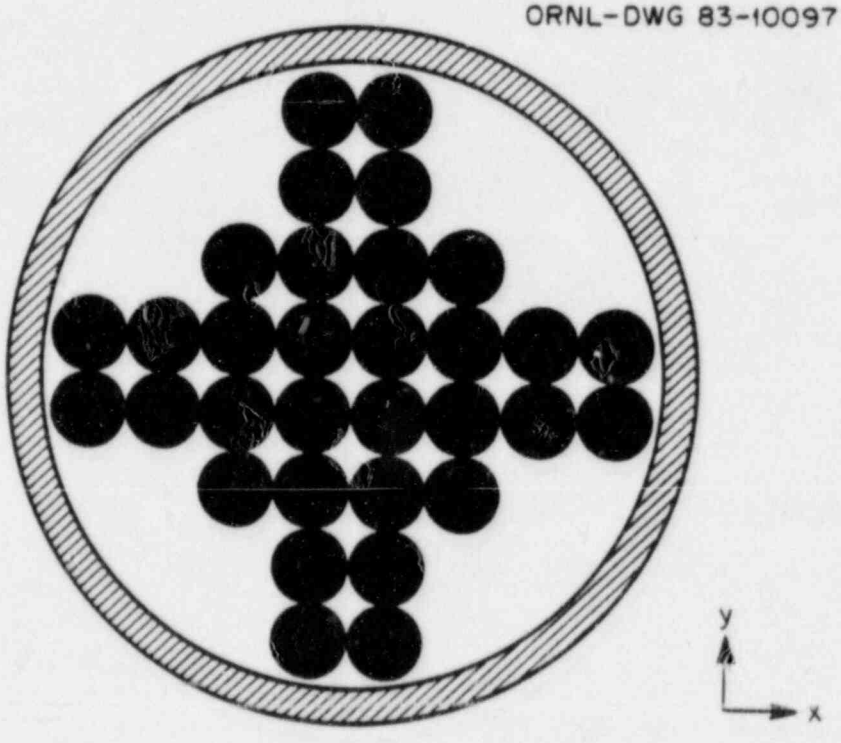


Fig. F11.5.16. Uranium rods in a cylindrical container

To generate the geometry description for this system, define Unit 1 to be one uranium rod and its associated square pitch close packed spacing region.

```
UNIT 1
CYLINDER 1 1 0.5 2P5.0
CUBOID   0 1 4P0.5 2P5.0
```

Define array 1 to be the central square array consisting of four bundles of rods.

ARA=1 NUX=4 NUY=4 NUZ=1 FILL F1 END FILL

Define array 2 to be a bundle of four rods.

ARA=2 NUX=2 NUY=2 NUZ=1 FILL F1 END FILL

Now place array 2 in Unit 2. This defines the outer boundaries of an imaginary cuboid that contains the array. It is convenient to have the origin of the array at its center, so the most negative point of the array will be (-1,-1,-5).

UNIT 2
ARRAY 2 -1.0 -1.0 -5.0

A core description is used to place array 1 in the global unit. Then the cylindrical container is described around it and holes are used to place the four outer bundles around the central array.

CORE 1 -2.0 -2.0 -5.0
CYLINDER 0 1 4.15 5.1 -5.0
HOLE 2 0.0 -3.0 0.0
HOLE 2 3.0 0.0 0.0
HOLE 2 0.0 3.0 0.0
HOLE 2 -3.0 0.0 0.0
CYLINDER 2 1 4.6 6.37 -6.27

The first hole places the bottom bundle of four rods, the second hole places the bundle of four rods at the right, the third hole places to top bundle of rods and the fourth hole places the left bundle of rods.

The overall problem description is shown below. Two of the printer plots used for verification of this mock-up are shown in Figs. F11.5.17 and F11.5.18.

```
=KENO5
CASK ARRAY
READ PARAMETERS  TME=1.0 FDN=YES LIB=41 GEN=10
END PARAMETERS
READ MIXT  SCT=1  MIX=1 92500 4.48006-2  92800 2.6578-3  92400 4.827-4
92600 9.57-5  MIX=2 100 1.0 END MIXT
READ GEOMETRY
UNIT 1
CYLINDER 1 1 0.5 2P5.0
CUBOID  0 1 4P0.5 2P5.0
UNIT 2
ARRAY 2 -1.0 -1.0 -5.0
CORE 1 1 -2.0 -2.0 -5.0
CYLINDER 0 1 4.15 5.1 -5.0
HOLE 2 0.0 -3.0 0.0
HOLE 2 3.0 0.0 0.0
HOLE 2 0.0 3.0 0.0
HOLE 2 -3.0 0.0 0.0
CYLINDER 2 1 4.6 6.37 -6.27
END GEOM
READ ARRAY
ARA=1 NUX=4 NUY=4 NUZ=1 FILL F1 END FILL
ARA=2 NUX=2 NUY=2 NUZ=1 FILL F1 END FILL
END ARRAY
READ PLOT TTL='X-Z SLICE AT Y=0.25 WITH X ACROSS AND Z DOWN'
XUL=-5.0 YUL=0.25 ZUL=6.5 XLR=5.0 YLR=0.25 ZLR=-6.5
UAX=1.0 WDN=-1.0 NAX=130 NCH=' *0' END
TTL='X-Y SLICE AT Z=0.0 WITH X ACROSS AND Y DOWN'
XUL=-5.0 YUL=5.0 ZUL=0.0 XLR=5.0 YLR=-5.0 ZLR=0.0
UAX=1.0 VDN=-1.0 NAX=130 END
END PLOT
END DATA
END
```

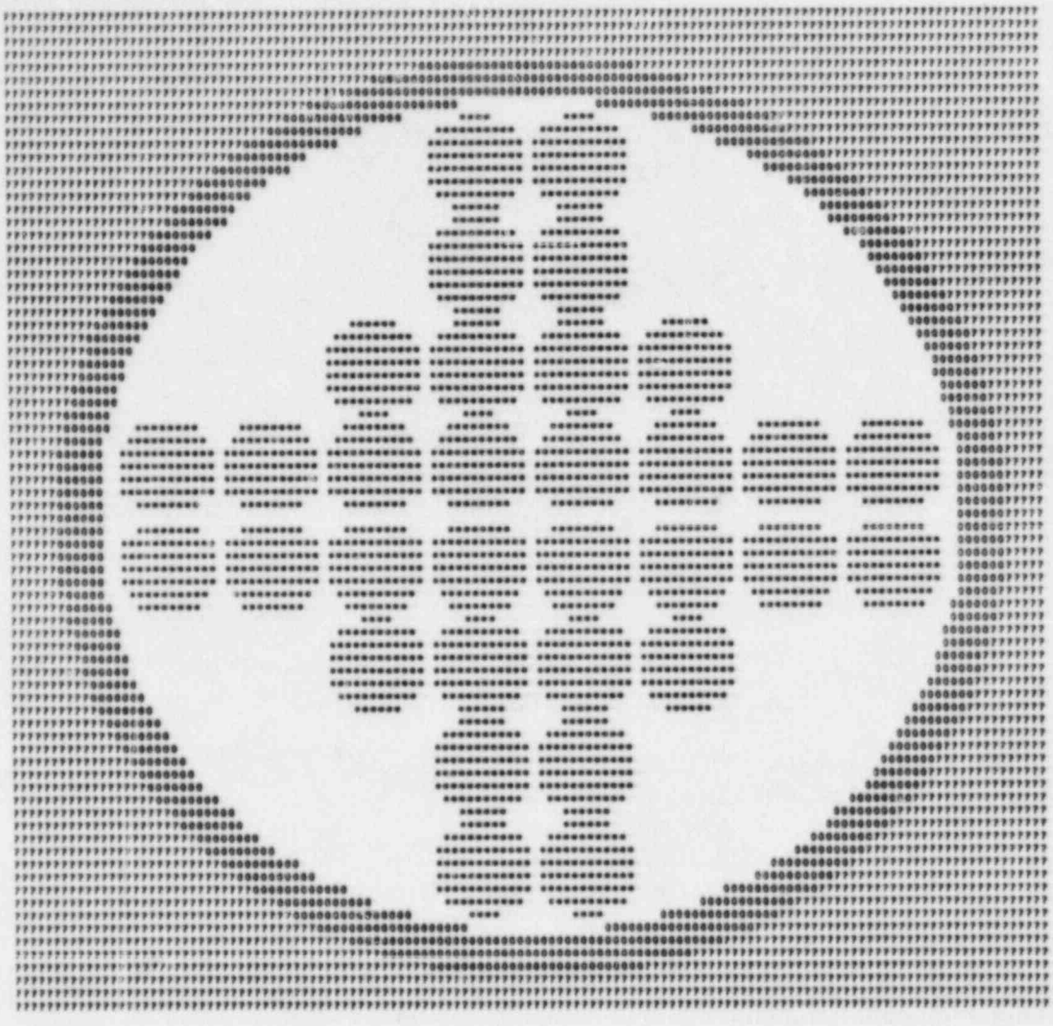


Fig. F11.5.17. x-y slice of uranium rods in a cylindrical container

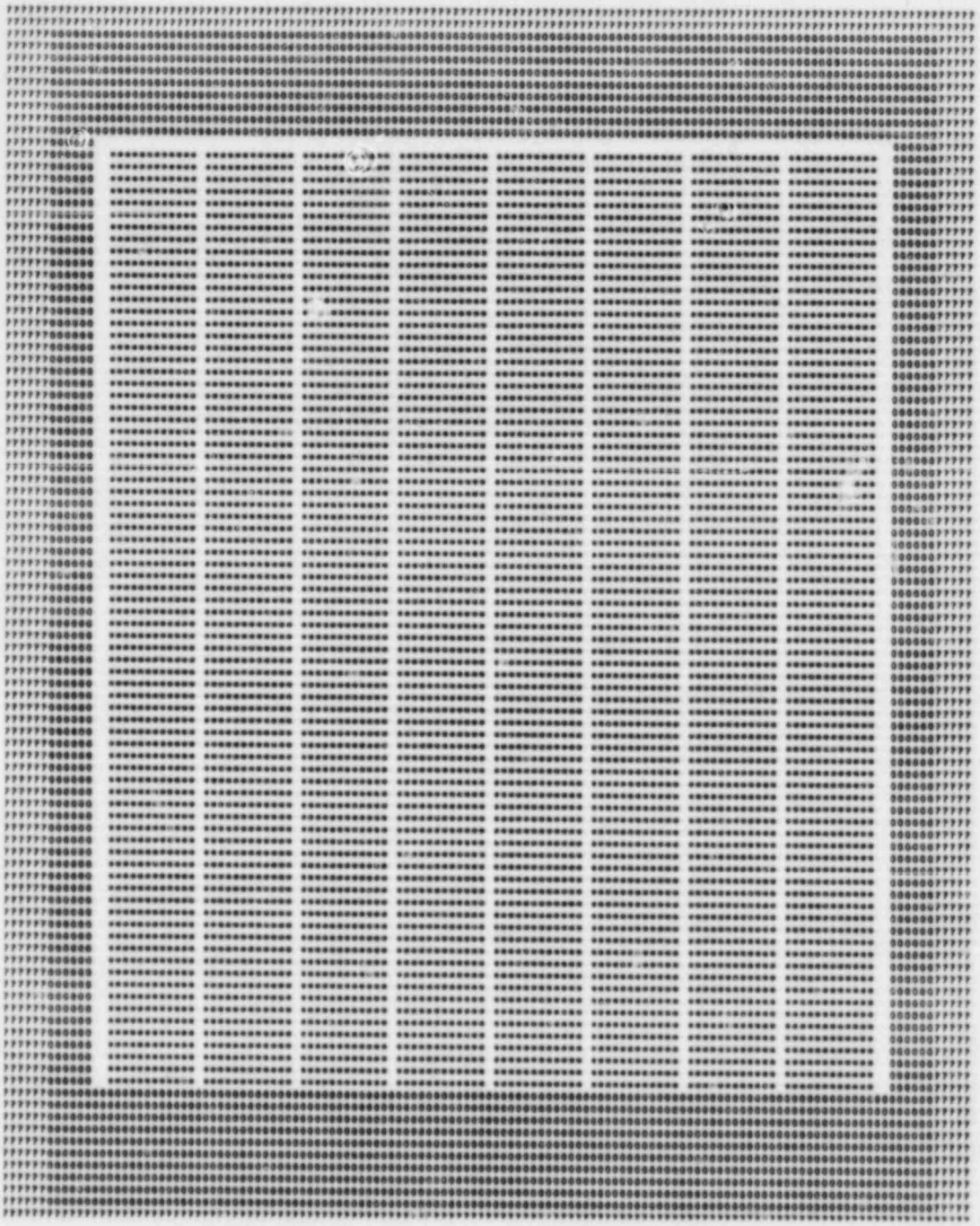
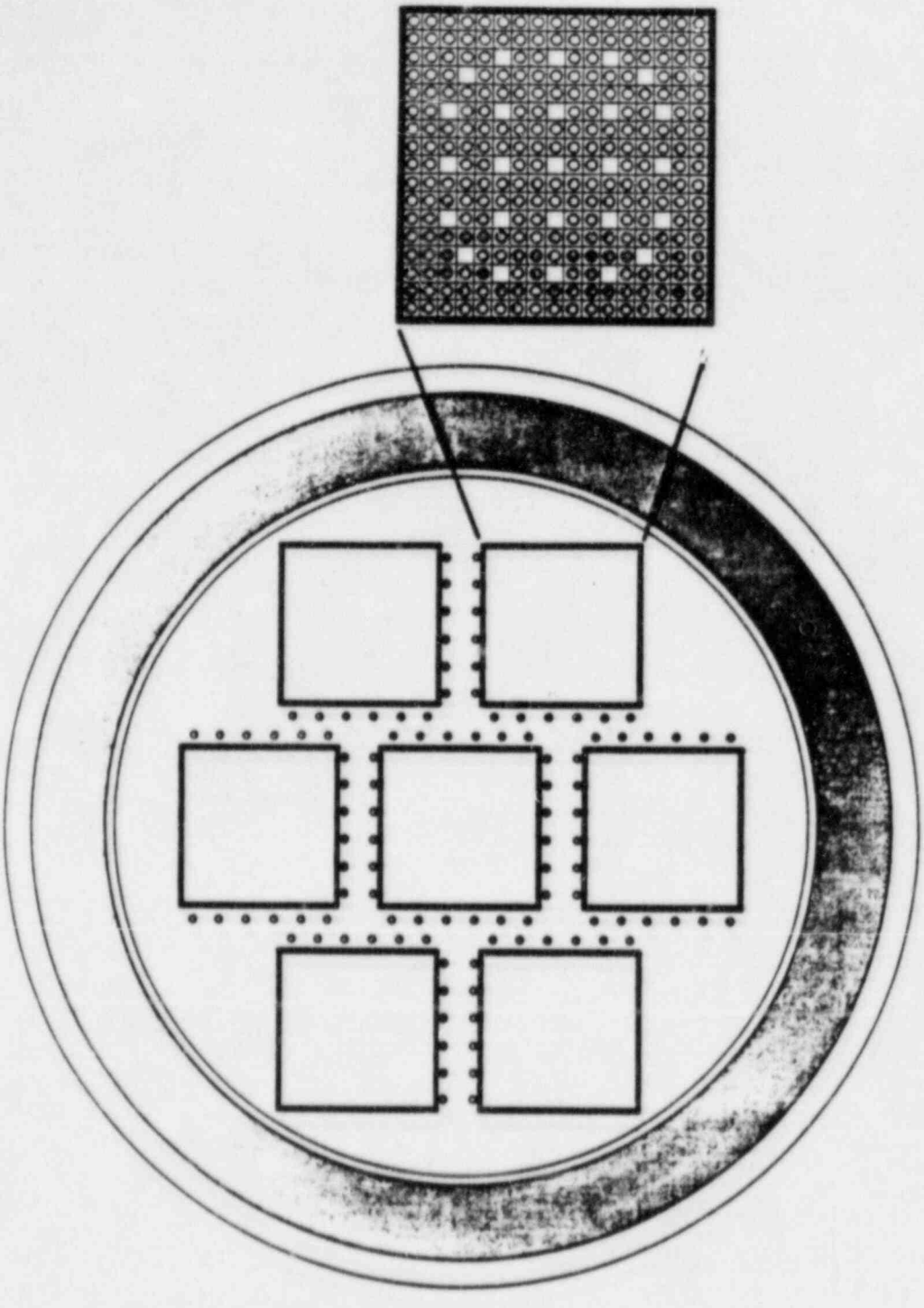


Fig. F11.5.18. x-z slice of uranium rods in a cylindrical container

EXAMPLE 2. A TYPICAL PWR SHIPPING CASK

Consider a typical PWR shipping cask, illustrated in Fig. F11.5.19. The interior and exterior of the cask is carbon steel and a depleted uranium gamma shield is present in the annulus. The shipping cask contains seven PWR fuel assemblies. Each assembly is a 17x17 array of fuel rods with water holes as shown. Each assembly is contained in stainless steel. Each fuel rod is clad with zircalloy and is composed of 4% enriched UO_2. Rods of B_4C clad with stainless steel are positioned between the fuel assemblies. The entire cask is filled with water.



Flg. F11.5.19. Typical PWR shipping cask

To describe the geometry of the cask, start by defining some simple units as shown in Fig. F11.5.20.

ORNL--DWG 83-14792

Fig. F11.5.20. Simple units

Unit 1 represents a fuel rod and its associated square pitch spacing region. Unit 2 represents a water hole in a fuel assembly.

```
UNIT 1
CYLINDER  1 1 .41148 365.76 0.0
CYLINDER  2 1 .48133 365.76 0.0
CUBOID    3 1 .63754 -.63754 .63754 -.63754 365.76 0.0

UNIT 2
CUBOID    3 1 .63754 -.63754 .63754 -.63754 365.76 0.0
```

Units 3, 4, and 6 represent the B_4C rods with their various spacings, and Unit 5 is a water hole that is used in association with some of the B_4C rods.

```
UNIT 3
CYLINDER  4 1 .584 365.76 0.0
CYLINDER  5 1 .635 365.76 0.0
CUBOID    3 1 .9912 -.9912 2.2352 -1.27 365.76 0.0


UNIT 4
CYLINDER  4 1 .584 365.76 0.0
CYLINDER  5 1 .635 365.76 0.0
CUBOID    3 1 .9912 -.9912 1.2702 -2.235 365.76 0.0


UNIT 5
CUBOID    3 1 .9912 -.9912 1.7526 -1.7526 365.76 0.0


UNIT 6
CYLINDER  4 1 .584 365.76 0.0
CYLINDER  5 1 .635 365.76 0.0
CUBOID    3 1 1.1875215 -1.1875215 1.883706 -1.883706 365.76 0.0
```

Units 1 and 2 are stacked together into array 1 to form the array of fuel pins and water holes in a fuel assembly as shown in Fig. F11.5.21. This array is then encompassed with a layer of water and a layer of stainless steel to complete a fuel assembly (Unit 7) as shown in Fig. F11.5.22.

```
ARA=1 NUX=17 NUY=17 NUZ=1 FILL
39R1 2 2Q3 8R1 2 9R1 2 22R1 2 4Q3 38R1 2 4Q3
Q51 22R1 2 Q10 Q9 2Q3 39R1                END FILL

UNIT 7   ARRAY 1 -10.83818 -10.83818 0.0
CUBOID   3 1 11.112495 -11.112495 11.112495 -11.112495 365.76 0.0
CUBOID   8 1 11.302238 -11.302238 11.302238 -11.302238 365.76 0.0
```

ORNL–DWG 83-14793

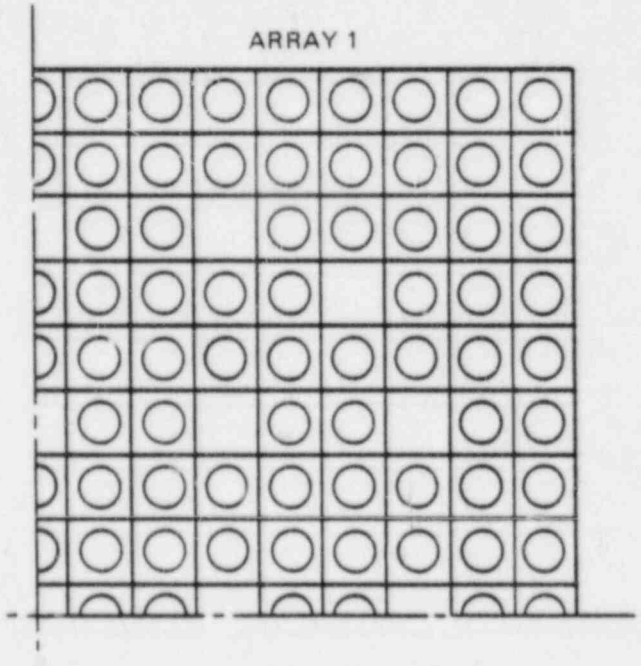


Fig. F11.5.21. Quarter section of fuel pin array

ORNL–DWG 83-14794

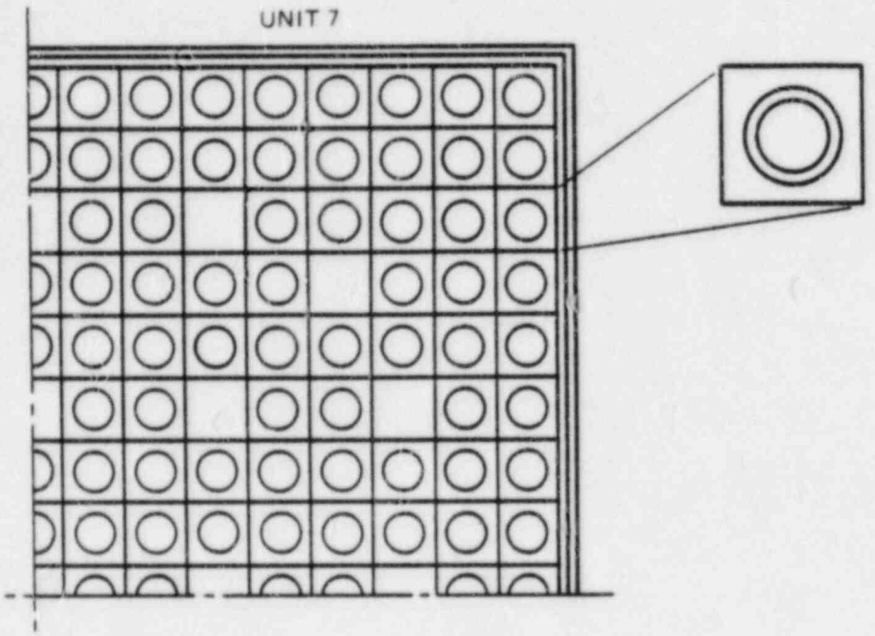


Fig. F11.5.22. Quarter section of fuel assembly.

An array of Unit 6's is created to represent the array of B_4C rods that is positioned between the fuel assemblies. This array of B_4C rods is contained in Unit 8 as shown in Fig. F11.5.23.

ARA=2 NUX=2 NUY=6 NUZ=1 FILL F6 END FILL

UNIT 8 ARRAY 2 0 0 0

The next step is to create the central array of three fuel assemblies with B_4C rods between them. This is done by stacking fuel assemblies (Unit 7) and B_4C rod arrays (Unit 8) into an array (array 3) and placing it in a unit (Unit 9). The resultant geometry is shown in Fig. F11.5.24.

ARA=3 NUX=5 NUY=1 NUZ=1 FILL 7 8 7 8 7 END FILL

UNIT 9 ARRAY 3 0 0 0

ORNL–DWG 83-14800

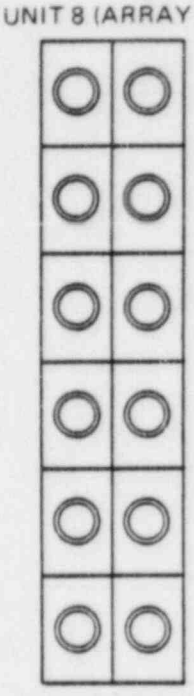


Fig. F11.5.23. 2 x 6 array of B_4C rods.

ORNL–DWG 83-14801

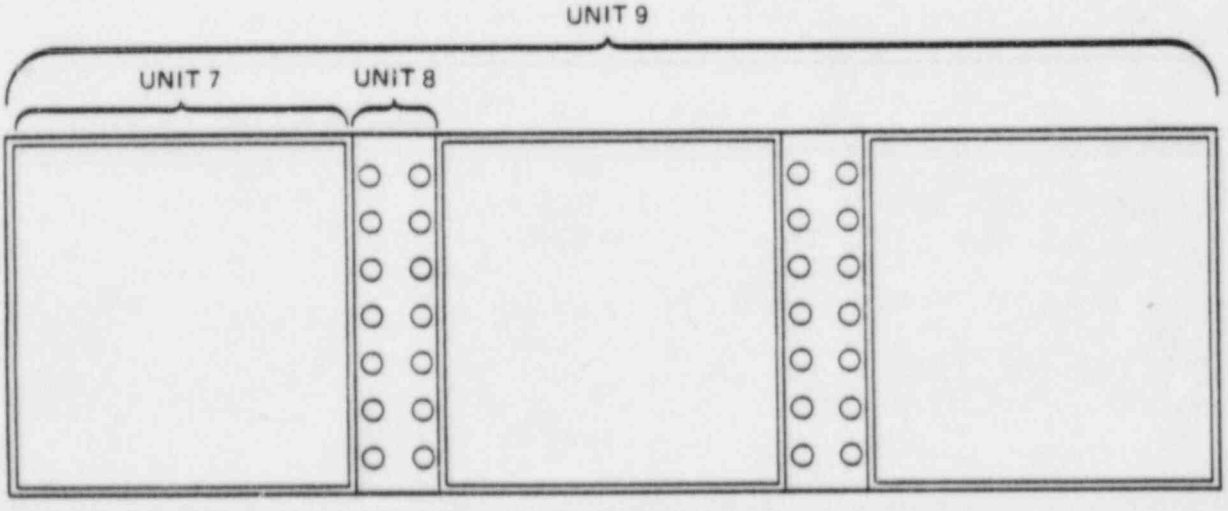


Fig. F11.5.24. Central array

Units 3, 4, and 5 are used to define the arrays of B_4C rods that fit above and below the central array as shown in Fig. F11.5.25.

```
ARA=4 NUX=39 NUY=1 NUZ=1   FILL 3 5 2Q2 3 4 2Q2 5 4 3 2Q2 5 3 4 2Q2 5 4 3 2Q2 5 2Q2 3
END FILL

UNIT 10   ARRAY 4 0 0 0

ARA=5 NUX=39 NUY=1 NUZ=1   FILL 4 5 2Q2 4 3 2Q2 5 3 4 2Q2 5 4 3 2Q2 5 3 4 2Q2 5 2Q2 4
END FILL

UNIT 11   ARRAY 5 0 0 0
```

ORNL–DWG 83-14802

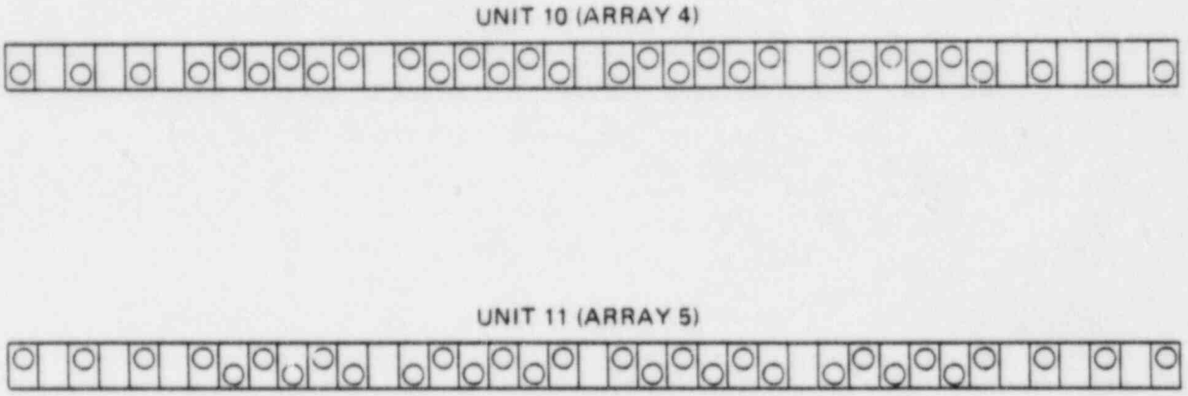


Fig. F11.5.25. Long B_4C rod arrays.

Units 9, 10, and 11 are stacked to form the central array with B_4C rods as shown in Fig. F11.5.26.

```
ARA=6 NUX=1 NUY=3 NUZ=1   FILL 11 9 10 END FILL
```

This completes the three central fuel assemblies and all the B_4C rods associated with them. Next, Units 7 and 8 are stacked together to form the array of two fuel assemblies separated by B_4C rods as shown in Fig. F11.5.27. This is designated as array 7 and Unit 12. The origin of Unit 12 is specified at the center of the array in the x and y directions and the bottom of the fuel assemblies are at z=27.94 cm.

```
ARA=7 NUX=3 NUY=1 NUZ=1   FILL 7 8 7   END FILL

UNIT 12   ARRAY 7 -24.979519 -11.302238 27.94
```

ORNL-DWG 83-14803

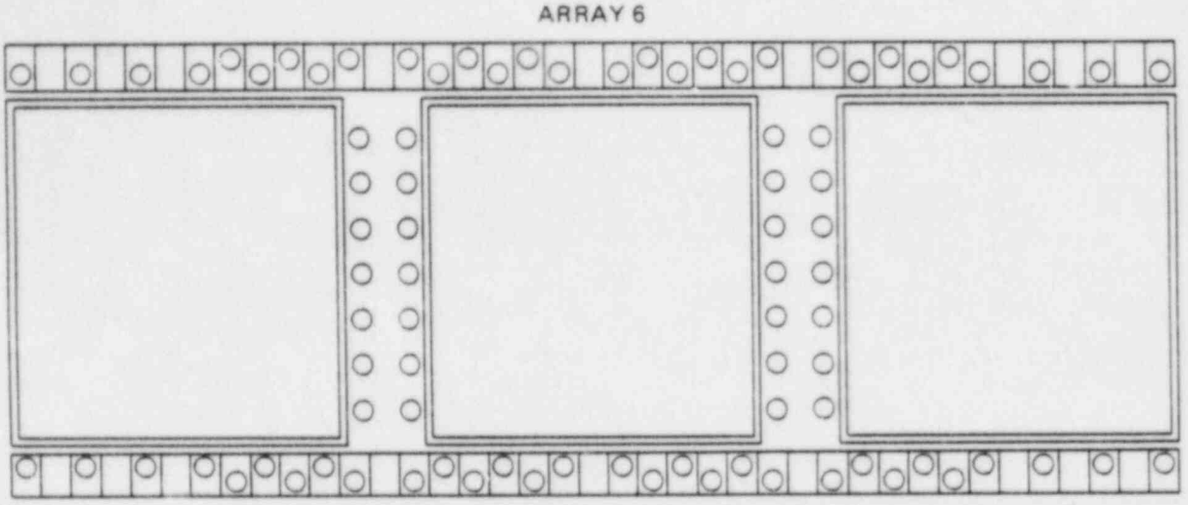


Fig. F11.5.26. Central array with long B_4C arrays

ORNL-DWG 83-14804

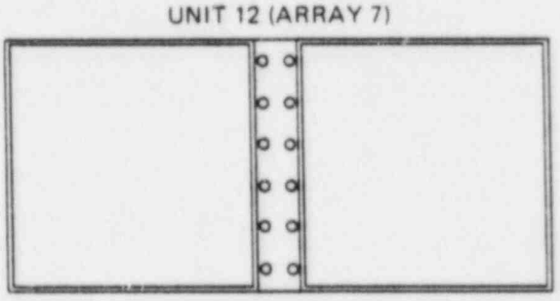


Fig. F11.5.27. Two fuel assemblies and B_4C rods

Unit 13 is simply a cylindrical lid that fits on top of the shipping cask. It is described relative to the origin of the shipping cask and is made of depleted uranium.

UNIT 13
CYLINDER 6 1 47.625 457.2 449.58

The shipping cask is completed by specifying the origin of array 6 (see Fig. F11.5.26) to be at the center of the array in x and y and the bottom of the array is at z=27.94 cm. Note that the three dimensions specified when an ARRAY is placed in a unit are the coordinates of the most negative point in the array with respect to the origin of the unit that contains the array. A cylinder of water defining the interior of the shipping cask is described around the array. A HOLE is used to place a Unit 12 (Fig. F11.5.27) below the array and a second HOLE is used to place another Unit 12 above the array. Then a cylinder of steel is placed around the water, which is in turn encased by depleted uranium. The

depleted uranium is then contained in the outer steel cylinder of the shipping cask. A third HOLE is used to place the depleted uranium lid (Unit 13) on the shipping cask. This completes the shipping cask description of Fig. F11.5.19.

```
ARRAY    6  -38.6568 -14.807438 27.94
CYLINDER 3 1 47.625 447.04 16.5!
HOLE     12   0.0 -26.1097 0.0
HOLE     12   0.0  26.1097 0.0
CYLINDER 7 1 48.895 447.04 13.335
CYLINDER 6 ! 59.06 447.04 3.81
CYLINDER 7 1 63.01 462.28 0.0
HOLE     13   0.0 0.0 0.0
```

Array 6 is the global array for this problem and is therefore not preceded by a unit definition. It defines the coordinate system for the overall problem. The geometry data for this shipping cask are shown below. The plot data have been included for verification of the geometry description. However the plot generated by this data is quite large and is therefore not included in this document.

```
READ GEOM
UNIT 1
CYLINDER  1 1 .41148 365.76 0.0
CYLINDER  2 1 .48133 365.76 0.0
CUBOID    3 1 .63754 -.63754 .63754 -.63754 365.76 0.0
UNIT 2
CUBOID    3 1 .63754 -.63754 .63754 -.63754 365.76 0.0
UNIT 3
CYLINDER  4 1 .584 365.76 0.0
CYLINDER  5 1 .635 365.76 0.0
CUBOID    3 1 .9912 -.9912 2.2352 -1.27 365.76 0.0
UNIT 4
CYLINDER  4 1 .584 365.76 0.0
CYLINDER  5 1 .635 365.76 0.0
CUBOID    3 1 .9912 -.9912 1.2702 -2.235 365.76 0.0
UNIT 5
CUBOID    3 1 .9912 -.9912 1.7526 -1.7526 365.76 0.0
UNIT 6
CYLINDER  4 1 .584 365.76 0.0
CYLINDER  5 1 .635 365.76 0.0
CUBOID    3 1 1.1875215 -1.1875215 1.883706 -1.883706 365.76 0.0
UNIT 7    ARRAY 1 -10.83818 -10.83818 0.0
CUBOID    3 1 11.112495 -11.112495 11.112495 -11.112495 365.76 0.0
CUBOID    8 1 11.302238 -11.302238 11.302238 -11.302238 365.76 0.0
UNIT 8    ARRAY 2 0 0 0
UNIT 9    ARRAY 3 0 0 0
UNIT 10   ARRAY 4 0 0 0
UNIT 11   ARRAY 5 0 0 0
UNIT 12   ARRAY 7 -24.979519 -11.302238 27.94
UNIT 13
CYLINDER  6 1 47.625 457.2 449.58
ARRAY     6  -38.6568 -14.807438 27.94
CYLINDER  3 1 47.625 447.04 16.51
HOLE      12   0.0 -26.1097 0.0
HOLE      12   0.0  26.1097 0.0
CYLINDER  7 1 48.895 447.04 13.335
CYLINDER  6 1 59.06 447.04 3.81
CYLINDER  7 1 63.01 462.28 0.0
HOLE      13   0.0 0.0 0.0
END GEOM
READ ARRAY
ARA=1 NUX=17 NUY=17 NUZ=1 FILL
39R1 2 2Q3 8R1 2 9R1 2 12R1 2 4Q3 38R1 2 4Q3
Q51 22R1 2 Q10 Q9 2Q3 39R1                END FILL
ARA=2 NUX=2 NUY=6 NUZ=1    FILL F6        END FILL
ARA=3 NUX=5 NUY=1 NUZ=1    FILL 7 8 7 8 7 END FILL
ARA=4 NUX=39 NUY=1 NUZ=1   FILL 3 5 2Q2 3 4 2Q2 5 4 3 2Q2 5 3 4 2Q2
5 4 3 2Q2 5 2Q2 3                         END FILL
ARA=5 NUX=39 NUY=1 NUZ=1   FILL 4 5 2Q2 4 3 2Q2 5 3 4 2Q2 5 4 3 2Q2
5 3 4 2Q2 5 2Q2 4                         END FILL
ARA=6 NUX=1 NUY=3 NUZ=1    FILL 11 9  10  END FILL
ARA=7 NUX=3 NUY=1 NUZ=1    FILL 7 8 7     END FILL
END ARRAY
READ PLOT
TTL=? SHIPPING CASK IF-300 X-Y SLICE  ?
XUL=-63 YUL=63 ZUL=180 XLR=63 YLR=-63 ZLR=180
UAX=1 VDN=-1 NAX=350
PLT=NO
END PLOT
```

F11.5.6.5 *Specification of the Global System*

KENO geometry data must be correlated to an absolute coordinate system. In the past, the code was responsible for determining this coordinate system. KENO V.a allows the user to define the absolute coordinate system by specifying the global unit. If the user does not exercise this option, the code will assign the default as before. The use of the word GLOBAL in the geometry region data is used to specify the global unit. A global array is required only when a problem consists of a bare array and thus has no global unit. In that event, the user can specify the global array by entering GBL= in the array data.

In the geometry region data for a problem, the word GLOBAL preceding a UNIT NUMBER DEFINITION or an ARRAY PLACEMENT DESCRIPTION specifies that unit to be the global unit. Only one global specification should be used in a problem. If GLOBAL is entered more than once in the geometry region data, the last entry defines the global system. The specification of a global unit, whether manually or by default, overrides the manual specification of the global array in the array data. Entering the word GLOBAL in the geometry region data always overrides the use of GBL= in the array data.

By default, the global unit is defined to be the last ARRAY PLACEMENT DESCRIPTION that does not immediately follow a UNIT NUMBER DEFINITION. The associated surrounding geometry, if any, is the external reflector. The default global array is the array referenced by the global unit or, in the absence of a global unit, the largest array number specified in the array data. If an array is referenced by the global unit, it is always the first geometry region in the global unit (i.e., an ARRAY PLACEMENT DESCRIPTION is the first geometry region in that unit.) The array number of the ARRAY PLACEMENT DESCRIPTION specifies the array that is contained in the unit. This array may contain other arrays, nested to any level. However, the global array is the one that is considered to be in the global unit, just as each subsequent array within the global array is considered to be within the unit that contains it. If the global array specified in the array data (GBL=) is inconsistent with the specification of the global array in the geometry region data, an error message will be written and the problem will not run. Some examples of global specification follow.

EXAMPLE 1. Consider the 2x2x2 array of sample problem 1 shown in Fig. F11.D.1. The array consists of four identical units in each of two layers. The geometry for this problem is listed below. By default, the global array is array 1. The most negative point of the array is located at the origin so the array extends from x=0.0 to x=27.48, y=0.0 to y=27.48 and z=0.0 to z=26.02.

```
SAMPLE PROBLEM 1  CASE 2C8 BARE
READ GEOMETRY
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
END GEOMETRY
READ ARRAY  NUX=2 NUY=2 NUZ=2  END ARRAY
```

EXAMPLE 2, Case 1. Consider the 2x2x2 reflected array of sample problem 3 shown in Fig. F11.D.2. The array consists of four identical units in each of two layers. The array is reflected by 15.24 cm of paraffin. The geometry for this problem is listed below. By default, the global unit is the unit whose first geometry region is CORE. Note that the array number following the word CORE is defaulted to 1 if a zero is entered. Array 1 is contained within this unit and is thus the global array. In this geometry description, the most negative point of array 1 is located at (-23.48,-23.48,-22.75) relative to the origin of the global unit. Thus the array extends from x=-23.48 to x=+23.48, y=-23.48 to y=+23.48, and z=-22.75 to z=+22.75. The overall system extends from x=-38.72 to x=+38.72, y=-38.72 to y=+38.72 and z=-37.99 to z=+37.99.

```
SAMPLE PROBLEM 3  2C8  15.24 CM PARAFFIN REFL
READ ARRAY NUX=2 NUY=2 NUZ=2 END ARRAY
READ GEOM
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 11.74 -11.74 11.74 -11.74 11.375 -11.375
CORE  0 1 -23.48 -23.48 -22.75
CUBOID  2 2 26.48 -26.48 26.48 -26.48 25.75 -25.75
CUBOID  2 3 29.48 -29.48 29.48 -29.48 28.75 -28.75
CUBOID  2 4 32.48 -32.48 32.48 -32.48 31.75 -31.75
CUBOID  2 5 35.48 -35.48 35.48 -35.48 34.75 -34.75
CUBOID  2 6 38.72 -38.72 38.72 -38.72 37.99 -37.99
END GEOM
```

EXAMPLE 2, Case 2. The 2x2x2 reflected array of sample problem 3 above could also have been described by explicitly specifying the global unit as shown below. The geometry remains unchanged and the results will be identical. A third method would be to omit the "UNIT 2." Again the geometry and results will be unchanged.

```
SAMPLE PROBLEM 3  2C8  15.24 CM PARAFFIN REFL
READ ARRAY NUX=2 NUY=2 NUZ=2 END ARRAY
READ GEOM
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 11.74 -11.74 11.74 -11.74 11.375 -11.375
GLOBAL
UNIT 2
CORE  0 1 -23.48 -23.48 -22.75
CUBOID  2 2 26.48 -26.48 26.48 -26.48 25.75 -25.75
CUBOID  2 3 29.48 -29.48 29.48 -29.48 28.75 -28.75
CUBOID  2 4 32.48 -32.48 32.48 -32.48 31.75 -31.75
CUBOID  2 5 35.48 -35.48 35.48 -35.48 34.75 -34.75
CUBOID  2 6 38.72 -38.72 38.72 -38.72 37.99 -37.99
END GEOM
```

EXAMPLE 3, Case 1. Consider sample problem 15 illustrated in Fig. F11.D.6. This problem requires the use of two different units to describe the geometry. Unit 1 describes the portion of the sphere that extends into the Plexiglas collar, the Plexiglas collar and a cuboid of water tight fitting around them. Unit 2 describes the portion of the sphere that is above the Plexiglas collar, surrounded by a cuboid of water having an edge dimension the same as the diameter of the Plexiglas collar and a height equal to the partial sphere. These two units are stacked with Unit 2 on top of Unit 1. The global unit is the geometry external to the array (the CORE, CYLINDER and REPLICATE). The global unit is cylindrical in shape and has its origin at the center of the array (0,0,0). The most negative point in the array is positioned at (-12.7,-12.7,-7.092175). Thus the array extends from x=-12.7 to x=+12.7, y=-12.7 to y=+12.7, and z=-7.092175 to z=+7.092175.

```
SAMPLE PROBLEM 15  SMALL WATER REFLECTED SPHERE ON PLEXIGLAS COLLAR
READ GEOM
UNIT 1
HEMISPHE-Z 1 1 6.5537 CHORD -5.09066
CYLINDER  3 1 4.1275 -5.09066 -7.63065
CYLINDER  2 1 12.7   -5.09066 -7.63065
CUBOID  3 1 4P12.7 -5.09066 -7.63065
UNIT 2
HEMISPHE+Z 1 1 6.5537 CHORD 5.09066
CUBOID  3 1 4P12.7 6.5537 -5.09066
CORE  0 1 -12.7 -12.7 -7.092175
CYLINDER  3 1 17.97 2P7.0922
REPLICATE 3 2 3*3.0 5
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=2  FILL 1 2 END ARRAY
```

EXAMPLE 3, Case 2. Consider sample problem 15 described above. Again define Unit 1 to contain the portion of the sphere that extends into the Plexiglas collar, the Plexiglas collar and a cuboid of water snugly surrounding them. Unit 2 contains the portion of the sphere that is above the Plexiglas collar, surrounded by a cuboid of water having an edge dimension equal to the diameter of the Plexiglas collar and a height equal to the partial sphere. These two units are stacked with Unit 2 on top of Unit 1. The global unit is the geometry external to the array (the CORE, CYLINDER and REPLICATE). The global unit is cylindrical in shape and has its origin at the center of the sphere (0,0,0). The most negative point in the array is positioned at (-12.7,-12.7,-7.63065). Thus the array extends from x=-12.7 to x=+12.7, y=-12.7 to y=+12.7, and z=-7.63065 to z=+6.5537.

```
SAMPLE PROBLEM 15  SMALL WATER REFLECTED SPHERE ON PLEXIGLAS COLLAR
READ GEOM
UNIT 1
HEMISPHE-Z 1 1 6.5537 CHORD -5.09066
CYLINDER  3 1 4.1275 -5.09066 -7.63065
CYLINDER  2 1 12.7   -5.09066 -7.63065
CUBOID  3 1 4P12.7 -5.09066 -7.63065
UNIT 2
HEMISPHE+Z 1 1 6.5537 CHORD 5.09066
CUBOID  3 1 4P12.7 6.5537 -5.09066
CORE  0 1 -12.7 -12.7 -7.63065
CYLINDER  3 1 17.97 6.5537 -7.63065
REPLICATE 3 2 3*3.0 5
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=2  FILL 1 2 END ARRAY
```

EXAMPLE 3, Case 3. Consider sample problem 15 as described above. The problem geometry can remain basically unchanged except the global unit is explicitly specified as shown below. The problem is identical to that of EXAMPLE 3, Case 2, and the results will be identical. The same problem could be run by omitting the "UNIT 3" and/or replacing "CORE 0" with "ARRAY 1."

```
SAMPLE PROBLEM 15  SMALL WATER REFLECTED SPHERE ON PLEXIGLAS COLLAR
READ GEOM
UNIT 1
HEMISPHE-Z 1 1 6.5537 CHORD -5.09066
CYLINDER  3 1 4.1275 -5.09066 -7.63065
CYLINDER  2 1 12.7   -5.09066 -7.63065
CUBOID  3 1 4P12.7 -5.09066 -7.63065
UNIT 2
HEMISPHE+Z 1 1 6.5537 CHORD 5.09066
CUBOID  3 1 4P12.7 6.5537 -5.09066
GLOBAL
UNIT 3
CORE  0 1 -12.7 -12.7 -7.63065
CYLINDER  3 1 17.97 6.5537 -7.63065
REPLICATE 3 2 3*3.0 5
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=2  FILL 1 2 END ARRAY
```

EXAMPLE 4, Case 1. Consider sample problem 19, illustrated in Fig. F11.D.3. The array of arrays geometry for this problem is shown below. Unit 1 defines a Plexiglas cylinder filled with uranyl nitrate solution. Unit 2 defines a uranium metal cylinder. Array 1 defines the 2x2 array of solution units at the back of the figure. Array 2 defines the 2x2 array of metal units at the front of the figure. Unit 3 contains array 1, and Unit 4 contains array 2. The global array is defined to be array 3 by specifying GBL=3 in the ARRAY DATA. Array 3 is composed of Unit 3 and Unit 4 and defines the overall system. The most negative point in the global array is at the origin of the global unit. The global array extends from x=0.0 to x=34.93, y=0.0 to y=43.5, z=0.0 to z=46.96. This is also the overall dimensions of the implied global array.

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ GEOM
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
UNIT 2
CYLINDER  1 1 5.748 2P5.3825
CUBOID  0 1 4P6.59 2P6.225
UNIT 3
ARRAY 1 3*0.0
UNIT 4
ARRAY 2 3*0.0
REFLECTOR 0 1 2*0.0 2*8.57 2*8.03 1
END GEOM
READ ARRAY  ARA=1 NUX=1 NUY=2 NUZ=2  FILL F1 END FILL
ARA=2 NUX=1 NUY=2 NUZ=2 FILL F2 END FILL GBL=3 ARA=3 NUX=2 NUY=1 NUZ=1
FILL 4 3 END FILL
END ARRAY
```

EXAMPLE 4, Case 2. Sample problem 19 could also have been described by omitting GBL=3. The global array would have defaulted to array 3 because it is the largest array number in the array data.

EXAMPLE 4, Case 3. Still another way of specifying the data for sample problem 19 would be to omit GBL=3 in the array data and specify array 3 as the global array in the geometry data as shown below. The problem would still be identical to the previous descriptions although the geometry is specified in a slightly different manner.

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ GEOM
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
UNIT 2
CYLINDER  1 1 5.748 2P5.3825
CUBOID  0 1 4P6.59 2P6.225
UNIT 3
ARRAY 1 3*0.0
UNIT 4
ARRAY 2 3*0.0
REFLECTOR 0 1 2*0.0 2*8.57 2*8.03 1
GLOBAL
ARRAY 3 3*0.0
END GEOM
READ ARRAY  ARA=1 NUX=1 NUY=2 NUZ=2  FILL F1 END FILL
ARA=2 NUX=1 NUY=2 NUZ=2 FILL F2 END FILL ARA=3 NUX=2 NUY=1 NUZ=1
FILL 4 3 END FILL
END ARRAY
```

EXAMPLE 4, Case 4. Consider the data specified in EXAMPLE 4, Case 1. It is possible to specify a portion of that data to be the global unit. This causes the code to ignore all of the data that are not referenced by the global unit. The 2x2 array of uranyl nitrate units can be run by specifying Unit 3 to be the global unit as shown below. Only the 2x2 array of solution units will be utilized by the problem.

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ GEOM
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
UNIT 2
CYLINDER  1 1 5.748 2P5.3825
CUBOID  0 1 4P6.59 2P6.225
GLOBAL
UNIT 3
ARRAY 1 3*0.0
UNIT 4
ARRAY 2 3*0.0
REFLECTOR 0 1 2*0.0 2*8.57 2*8.03 1
ARRAY 3 3*0.0
END GEOM
READ ARRAY  ARA=1 NUX=1 NUY=2 NUZ=2  FILL F1 END FILL
ARA=2 NUX=1 NUY=2 NUZ=2 FILL F2 END FILL ARA=3 NUX=2 NUY=1 NUZ=1
FILL 4 3 END FILL
END ARRAY
```

EXAMPLE 4, Case 5. The problem of EXAMPLE 4, Case 4 can also be specified by omitting the UNIT 3 card and placing the global specification before ARRAY 1 as shown below.

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ GEOM
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
UNIT 2
CYLINDER  1 1 5.748 2P5.3825
CUBOID  0 1 4P6.59 2P6.225
GLOBAL
ARRAY 1 3*0.0
UNIT 4
ARRAY 2 3*0.0
REFLECTOR 0 1 2*0.0 2*8.57 2*8.03 1
ARRAY 3 3*0.0
END GEOM
READ ARRAY  ARA=1 NUX=1 NUY=2 NUZ=2  FILL F1 END FILL
ARA=2 NUX=1 NUY=2 NUZ=2 FILL F2 END FILL ARA=3 NUX=2 NUY=1 NUZ=1
FILL 4 3 END FILL
END ARRAY
```

Both case 5 and case 4 of EXAMPLE 4 are equivalent to the following geometry data.

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ GEOM
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
END GEOM
READ ARRAY  ARA=1 NUX=1 NUY=2 NUZ=2  FILL F1 END FILL
END ARRAY
```

EXAMPLE 4, Case 6. If it is desirable to determine the keff of one of the uranyl nitrate units of sample problem 19, simply specify the unit as the global system as shown below.

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ GEOM
GLOBAL
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
UNIT 2
CYLINDER  1 1 5.748 2P5.3825
CUBOID  0 1 4P6.59 2P6.225
UNIT 3
ARRAY 1 3*0.0
UNIT 4
ARRAY 2 3*0.0
REFLECTOR 0 1 2*0.0 2*8.57 2*8.03 1
ARRAY 3 3*0.0
END GEOM
READ ARRAY  ARA=1 NUX=1 NUY=2 NUZ=2  FILL F1 END FILL
ARA=2 NUX=1 NUY=2 NUZ=2 FILL F2 END FILL ARA=3 NUX=2 NUY=1 NUZ=1
FILL 4 3 END FILL
END ARRAY
```

Case 6 of EXAMPLE 4 is equivalent to the data shown below.

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS
READ GEOM
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
END GEOM
```

F11.5.7 ALTERNATIVE SAMPLE PROBLEM MOCK-UPS

The geometry data for KENO V can often be described correctly in several ways. Some alternative geometry descriptions are given here for sample problem 12 and sample problem 13. (See Sect. F11.D.)

F11.5.7.1 *Sample Problem 12, First Alternative*

This mock-up maintains the same overall unit dimensions that were used in sample problem 12. In sample problem 12, the origin of Unit 1, the solution cylinder, is at the center of the unit; the origin of Units 2, 3, and 4, the metal cylinders, are at the center of the cylinders. In this mock-up, the unit numbers remain the same and the origin of each unit is at the center of the unit. In each unit the cylinder is offset by specifying the position of its centerline relative to the origin of the unit.

```
READ GEOM
UNIT 1
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 9.525 -9.525
CUBOID    0 1 10.875 -10.875 10.875 -10.875 10.24 -10.24
UNIT 2
CYLINDER  1 1 5.748 9.3975 -1.3975 ORIG 4.285 4.285
CUBOID    0 1 10.875 -10.875 10.875 -10.875 10.24 -10.24
UNIT 3
CYLINDER  1 1 5.748 9.3975 -1.3675 ORIG 4.285 -4.285
CUBOID    0 1 10.875 -10.875 10.875 -10.875 10.24 -10.24
UNIT 4
CYLINDER  1 1 5.748 1.3675 -9.3975 ORIG 4.285 4.285
CUBOID    0 1 10.875 -10.875 10.875 -10.875 10.24 -10.24
UNIT 5
CYLINDER  1 1 5.748 1.3675 -9.3975 ORIG 4.285 -4.285
CUBOID    0 1 10.875 -10.875 10.875 -10.875 10.24 -10.24
END GEOM
READ ARRAY  NUX=2 NUY=2 NUZ=2 FILL 2 1 3 1 4 1 5 1 END ARRAY
```

F11.5.7.2 *Sample Problem 12, Second Alternative*

In this mock-up, the outer boundaries of the system are made as close fitting as possible on all six faces. The origin of each unit is located at the center of the cylinder. Units 1, 3, 5, and 7 contain the metal cylinders. Units 2, 4, 6, and 8 contain the solution cylinders.

```
READ GEOM
UNIT 1
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID   0 1 6.59 -5.748 6.59 -14.445 6.225 -13.54
UNIT 2
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 9.525 -9.525
CUBOID    0 1 10.16 -10.875 10.875 -10.16 10.24 -9.525
UNIT 3
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID    0 1 6.59 -5.748 14.444 -6.59 6.225 -13.54
UNIT 4
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 9.525 -9.525
CUBOID    0 1 10.16 -10.875 10.16 -10.875 10.24 -9.525
UNIT 5
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID    0 1 6.59 -5.748 6.59 -14.445 13.54 -6.225
UNIT 6
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 9.525 -9.525
CUBOID    0 1 10.16 -10.875 10.875 -10.16 9.525 -10.24
UNIT 7
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID    0 1 6.59 -5.748 14.445 -6.59 13.54 -6.225
UNIT 8
CYLINDER  2 1 9.525 8.89 -8.89
CYLINDER  3 1 10.16 9.525 -9.525
CUBOID    0 1 10.16 -10.875 10.16 -10.875 9.525 -10.24
READ ARRAY  NUX=2 NUY=2 NUZ=2 FILL 6I1 8 END FILL END ARRAY
```

F11.5.7.3 *Sample Problem 13, Alternative*

This mock-up maintains the same overall unit dimensions that were used in sample problem 13, Sect. F11.D. In sample problem 13, the origin of Units 1, 2, and 3 is located at the center of the base of the uranium metal cuboid. In this mock-up, the origin of Units 1 and 2 is located at the center of the cylinder. In Unit 3, the origin is at the center of the unit.

```
READ GEOM
UNIT 1
CUBOID  1 1 0.2566 -12.4434 6.35 -6.35 3.81 -3.81
CYLINDER  0 1 13.97 3.81 -3.81
CYLINDER  1 1 19.05 3.81 -3.81
CUBOID  0 1 19.05 -19.05 19.05 -19.05 3.81 -3.81
UNIT 2
CUBOID  1 1 12.4434 -0.2566 6.35 -6.35 4.28 -4.28
CYLINDER  0 1 13.97 4.28 -4.28
CYLINDER  1 1 19.05 4.28 -4.28
CUBOID  0 1 19.05 -19.05 19.05 -19.05 4.28 -4.28
UNIT 3
CUBOID  1 1 12.4434 -0.2566 6.35 -6.35 1.308 -1.308
CUBOID  0 1 19.05 -19.05 19.05 -19.05 1.308 -1.308
END GEOM
READ ARRAY  NUX=1 NUY=1 NUZ=3  FILL 1 2 3 END ARRAY
```

F11.5.8 BIASING OR WEIGHTING DATA

Section F11.2.3 discusses the basis of weighting or biasing. The use of biasing data in reflected problems has been illustrated in Examples 9, 10, and 11 of Sect. F11.5.6. Section F11.4.7 discusses the input directions for entering biasing data.

Every geometry card requires a bias ID to associate that geometry region with a biasing or weighting function. A biasing or weighting function is a set of energy-dependent values of the average weight that are applicable in a given region. The default function for all bias ID's is constant through all energy groups, and is defined to be the default value of weight average which can be specified in the parameter data. A bias ID can be associated with a biasing function, other than default, by specifying it in the biasing input data. This function can be chosen from the weighting library or can be input from cards. Table F11.4.5 lists the materials and energy group structures for biasing functions available from the weighting library.

In general, the use of biasing should be restricted to external reflectors unless the user has generated correct biasing functions for other applications. Improper use of biasing functions can result in erroneous answers without giving any indication that they are invalid. Caution should be exercised in the generation and use of biasing functions.

Biasing functions are most applicable to thick external reflectors. Their use can significantly reduce the amount of computer time required to obtain answers in KENO V. If the user wishes to use a biasing function for a concrete reflector, for example, the following steps must be included in preparing the input data:

(1) The geometry region data must define the shape and dimensions of the reflector using the mixture ID for concrete and a sequence of bias ID's that associate the geometry region with the appropriate interval of the concrete weighting function. CAUTION: THE THICKNESS OF EACH REGION UTILIZING BIASING FUNCTIONS MUST MATCH OR VERY NEARLY MATCH THE INCREMENT THICKNESS OF THE WEIGHTING DATA. NO CHECK IS MADE ON THE REQUIREMENT. IT IS THE USER'S RESPONSIBILITY TO ASSURE CONSISTENCY.

(2) Biasing data must be entered. This must include the material ID for the reflector material (from Table F11.4.5 or as specified on cards) and a beginning and ending bias ID. The beginning bias ID is used to select the first set of energy-dependent average weights, and the subsequent sets of energy-dependent average weights are assigned consecutive ID's until the ending bias ID is reached.

Small deviations in reflector region thickness are allowed, such as using three generated regions with a thickness per region of 5.08 cm to generate a 15.24-cm-thick reflector of concrete, or using five generated regions with a thickness per region of 3.048 cm to generate a 15.24-cm-thick reflector of water. See Table F11.4.5 for a list of the increment thickness for each material in the weighting library. It is acceptable for the thickness of the last reflector region to be significantly different than the increment thickness. For example, a reflector card specifying five generated regions with a thicknesss per region of 3.0 cm could be followed by a reflector card specifying one region with a thickness per region of 0.24 cm. Assume material 2 is water and a 15.24-cm-thick cuboidal reflector of water is desired. The required reflector description and biasing data could be entered as follows:

```
REFLECTOR 2 2 6*3.0  5
REFLECTOR 2 7 6*0.24 1
READ BIAS ID=500 2 7 END BIAS
```

The same 15.24-cm-thick reflector can be described by including the extra 0.24 cm in the last region as shown below:

REFLECTOR 2 2 6*3.0 4
REFLECTOR 2 6 6*3.24 1
READ BIAS ID=500 2 6 END BIAS

Here the weighting functions associated with bias ID's 2, 3, 4, and 5 are defined by the first reflector card and each generated region has a thickness of 3.0 cm, corresponding exactly to the increment thickness for water in Table F11.4.5. Bias ID 6 is used for the last generated region which is 3.24 cm thick.

The following examples illustrate the use of biasing data. Suppose the user wishes to use the weighting function for water from Table F11.4.5 for bias ID's 2 through 6. The biasing input data would then be:

READ BIAS ID=500 2 6 END BIAS

The energy-dependent values of weight average for the first 3-cm interval of water will be used for weighting the geometry regions that specify a bias of ID of 2. The energy-dependent values of weight average for the second 3-cm interval of water will be used for geometry regions that specify a bias ID of 3, etc. Thus, the energy-dependent values of weight average for the fifth 3-cm interval of water will be used for geometry regions that specify a bias ID of 6. Geometry regions that use bias ID's other than 2, 3, 4, 5, and 6 will use the default value of weight average that is constant for all energies as a biasing function.

Several sets of biasing data can be entered at once. Assume the user wishes to use the weighting function for concrete from Table F11.4.5 for bias ID's 2 through 4 and the weighting function for water for bias ID's 5 through 7. The appropriate input data block is:

READ BIAS ID=301 2 4 ID=500 5 7 END BIAS

The energy-dependent values of weight average for the first 5-cm interval of concrete will be used for the geometry regions that specify a bias ID of 2, the energy-dependent values of weight average for the second 5-cm interval of concrete will be used for the geometry regions that specify a bias ID of 3, and the energy-dependent values of weight average for the third 5-cm interval of concrete will be used for the geometry regions that specify a bias ID of 4. The energy-dependent values of weight average for the first 3-cm interval of water will be used for geometry regions that specify a bias ID of 5, the values for the second 3-cm interval of water will be used for geometry regions that specify a bias ID of 6, and the values for the third 3-cm interval of water will be used for geometry regions that specify a bias ID of 7. The default value of weight average will be used for all bias ID's outside the range of 2 through 7.

If the biasing data block defines the same bias ID more than once, the value that is entered last supersedes previous entries. Assume the following data block is entered.

READ BIAS ID=400 2 7 ID=500 5 7 END BIAS

Then the data for paraffin (ID=400) will be used for bias ID's 2, 3, and 4, and the data for water (ID=500) will be used for bias ID's 5, 6, and 7.

EXAMPLE 1. Use of biasing data with a reflector card.

Assume a 5-cm-radius sphere of material 2 is reflected by a 20-cm-thickness of material 1 (concrete). The concrete reflector is spherical and close fitting upon the sphere of material 2. The mixing table must specify material 1 and material 2. Material 1 must be defined as concrete. The geometry and biasing data should be entered as follows:

```
READ GEOM
SPHERE 2 1 5.0
REPLICATE 1 2 5.0 4
END GEOM
READ BIAS ID=301 2 5 END BIAS
```

In the above example, the replicate card will generate four spherical geometry regions, each 5.0 cm thick. The bias ID for the first generated region is 2; the second, 3; the third, 4; and the fourth, 5. The biasing data block specifies that the biasing function for material ID 301 (concrete) will be used from the weighting library. The bias ID to which the energy-dependent weighting function for the first 5.0-cm interval of concrete is applied is 2; the energy-dependent weighting function for the fourth 5-cm interval of concrete is applied to the fourth generated geometry region. This generated region has a bias ID of 5.

Example 1 can be described without using a reflector card as shown below. The cards that are generated by the reflector card in the previous set of data are identical to the last four spheres in this mock-up.

EXAMPLE 1. Use of biasing without a reflector card.

```
READ GEOM
SPHERE 2 1 5.0
SPHERE 1 2 10.0
SPHERE 1 3 15.0
SPHERE 1 4 20.0
SPHERE 1 5 25.0
END GEOM
READ BIAS  ID=301 2 5 END BIAS
```

F11.5.9 PRINTER PLOTS

Printer plots are generated only if a plot data block has been entered for the problem and PLT=NO has not been entered in the parameter data or the plot data. When a printer plot is to be made, the user MUST correctly specify the upper left-hand corner of the plot with respect to the origin of the plot. The origin of a plot is defined as follows:

(1) SINGLE UNIT - The origin of the plot coincides with the origin of the geometry description.

(2) UNREFLECTED ARRAY - This is an array problem that does not have a CORE or ARRAY description in the EXTENDED GEOMETRY DESCRIPTION of the global array. The origin of the plot is located at the most negative point of the global array. This occurs at the lower left-hand back corner of the global array.

(3) REFLECTED ARRAY - The origin of the plot coincides with the origin of the CORE or ARRAY description in the EXTENDED GEOMETRY DESCRIPTION of the global array.

Printer plots can represent mixture numbers, unit numbers or bias ID numbers. A title can be entered for each plot. If plot titles are omitted, the title of the KENO case will be printed for each plot title until a plot title is entered. If a plot title is entered and a subsequent plot title is omitted, the last plot title prior to the omitted one will be used for the omitted one.

The upper left and lower right coordinates define the area (i.e., the slice and its location) for which the plot is to be made. The direction cosines across the page and the direction cosines down the page define the direction of the vector across the page and the vector down the page with respect to the geometry coordinate system. One of the simplest ways of generating a plot is to specify the desired coordinates of the upper left and lower right corners of the plot. Determine which plot axis is to be across the page and which is to be down. The sign of the direction cosine should be consistent with the direction of that component when moving from the upper left to lower right corner. For example, to draw a plot of an x-z slice at y=5.0 with x across the page and z down the page for a system whose x coordinates ranges from 0.0 to 10.0 and whose z coordinates range from 0.0 to 20.0, the upper left coordinate could be XUL=0.0 YUL=5.0 ZUL=20.0 and the lower right coordinates could be XLR=10.0 YLR=5.0 ZLR=0.0. Since x is to be plotted across the page with x=0.0 at the left and x=10.0 at the right, only the x component of the direction cosines across the page need be entered. It should be positive because going from 0.0 to 10.0 is moving in the positive direction. Thus, UAX=1.0 would be entered for the direction cosines across the page. VAX and WAX could be omitted. Z is to be plotted down the page with z=20.0 at the top and z=0.0 at the bottom. Therefore, only the z component of the direction cosines down the page need to be defined. It should be negative because moving from 20.0 to 0.0 is moving in the negative direction. Thus, WDN=-1.0 would be entered for the direction cosines down the page. UDN and VDN could be omitted. The sign of the direction cosines should be consistent with the coordinates of the upper left and lower right corners in order to get a plot.

It is not necessary that the plot be made for a slice orthogonal to one of the axes. Plots can be made of slices cut at any desired angle, but the user should exercise caution and be well aware of the distortion of shapes that can be introduced. (Nonorthogonal slices through cylinders plot as ellipses.)

The user can specify the horizontal and vertical spacing between points on the plot. It is usually advisable to enter one or the other. Entering both of them can cause distortion of the plot. DLX= is used to specify the horizontal spacing between points and DLD= is used to specify the vertical spacing between points. When only one of them is specified, the code calculates the correct value of the other such that the plot will not be distorted. The spacing for an undistorted plot is defined as DLD*LPI=DLX*CPI where LPI is the number of lines per inch printed vertically and CPI is the number of characters per inch horizontally. These parameters are a function of the printer or print chain. The vertical distortion factor, VDIS, is then defined by VDIS= (DLD*LPI)/(DLX*CPI). The code then calculates the number of intervals that will be printed across the page and down the page if both the upper left and lower right coordinates of the plot were entered. In some instances, it can be desirable to distort a plot by entering both DLX and DLD. It might be desirable to compress one-dimension relative to another. If DLX=0.5 and DLD=5.0 are entered as data for a printer that prints 10 characters per inch across and 8 lines per inch down, the portion of the plot that is printed in the vertical direction will be reduced by a factor of 8 relative to the portion printed in the horizontal direction ((5.0*8)/(0.5*10)). That is, if the coordinates specify a perfect square, the plot will be a rectangle that is about 8 times as wide as it is tall.

DLX or DLD can be specified by the user to be small enough to show the desirable detail in the plot. The plot is generated by starting at the upper left corner of the plot and generating a point every DLX across the page; then moving down DLD and repeating the generation of the points across the page.

NAX specifies the number of intervals that will be printed across the page. It may be convenient for the user to specify the number of characters that can be printed across the page on the printer that will be used. Larger plots can be created by specifying multiples of this number. In that case, the plot must be taped together to see the overall plot. The plot will print one page wide and full length. Then the next page width and full length will be printed, etc., until the entire plot is completed.

NDN specifies the number of intervals that will be printed down the page. If both NAX and NDN are entered, the plot may be distorted. If one of them is entered, the value of the other will be calculated so the plot will not be distorted.

LPI, the number of lines per inch that are printed on a page, can be entered to be consistent with the printer that will be used. The default value is 8 lines per inch. This parameter need be entered only once for a problem. It should be entered in the data for the first plot so all the plots will be printed in the same manner.

When a plot is being made, the first character represents the coordinates of the upper left corner. The value of DELV is added to the coordinate that is to be printed across the page and the next character is printed. DELV is added to that value to determine the location of the next character. That is, a point is determined every DELV across the page and a character is printed for each point. When a line has been completed, a new line is begun DELU from the first line. This procedure is repeated until the plot is complete. Some examples of printer plots are shown in Sects. F11.5.6.1, F11.5.6.2, F11.5.6.3 and F11.5.6.4. Further examples are shown below.

EXAMPLE 1. SINGLE UNIT WITH CENTERED ORIGIN

Consider two concentric cylinders in a cuboid. The inner cylinder is 5.2 cm in diameter. The outer cylinder has an inside diameter of 7.2 cm and an outside diameter of 7.6 cm. Both cylinders are 30 cm high. They are contained in a tight-fitting box whose wall thickness is 0.5 cm and whose top and bottom are each 1.0 cm thick. The inner cylinder is composed of mixture 1, the outer cylinder is made of mixture 4, and the box is made of mixture 2. The problem can be described with its origin at the center of the inner cylinder. The problem description for this arrangement is shown below:

```
=KENO5
SINGLE UNIT CONCENTRIC CYLINDERS IN CUBOID WITH ORIGIN AT CENTER
READ PARAM  RUN=NO LIB=41 TME=0.5 END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2 MIX=2 200 1.0 MIX=3 502 0.1
MIX=4 200 1.0
END MIXT
READ GEOM
CYLINDER 1 1 2.6 2P15.0
CYLINDER 0 1 3.6 2P15.0
CYLINDER 4 1 3.8 2P15.0
CUBOID 0 1 4P3.8 2P15.0
CUBOID 2 1 4P4.3 2P16.0
END GEOM
READ PLOT TTL='X-Y SLICE AT Z MIDPOINT.  SINGLE UNIT CONCENTRIC CYLS.'
XUL=-4.6 YUL=4.6 ZUL=0.0 XLR=4.6 YLR=-4.6 ZLR=0.0
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *-.X' END
PIC=UNIT NCH='01' END
END PLOT
END DATA
END
```

The plot data block included above is set up to draw a mixture map of an x-y slice taken at the half height (z=0.0) and a unit map for the same slice. The code will print question marks for points outside the range of the problem geometry description. By setting the plot dimensions slightly larger than the geometry dimensions, a border of question marks will be printed around the specified plot. This verifies that the outer boundaries of the geometry are contained within the plot dimensions. In the above example, the geometry dimensions extend from x=-4.3 to x=4.3, from y=-4.3 to y=4.3, and from z=-8.0 to z=8.0. An x-y slice is to be printed at the half height (z=0.0). The desired plot data sets the upper left-hand corner of the plot to be x=-4.6 and y=4.6. The lower right-hand corner of the plot is specified as x=4.6 and y=-4.6. These data are entered by specifying the upper left-hand corner as XUL=-4.6 YUL=4.6 ZUL=0.0 and the lower right-hand corner as XLR=4.6 YLR=-4.6 ZLR=0.0. It is desired to print x across the page and y down the page. Therefore, the x direction cosine is specified across the page, in the direction from x=-4.6 to x=4.6 as UAX=1.0. The y direction cosine is specified down the page, from y=4.6 to y=-4.6 as VDN=-1.0. It was desirable for the plot to be one page wide (130 characters) so the number of characters across the page was specified as NAX=130. An arbitrary choice was made to print a blank for a void, a * for mixture 1, a - for mixture 2, and a . for mixture 4. Mixture 3 was not used in the problem, so a character did not have to be entered for it in the character string. Thus, a character string of NCH=' *-.' would have been sufficient but a string of NCH=' *-.x' was entered. Since only three mixtures were used, only the first four characters were utilized. The blank represents a void, the * represents the smallest mixture number used in the problem (mixture 1), the - represents the next smallest mixture number used in the geometry description (mixture 2) and the . represents the largest mixture number used in the geometry (mixture 4). The resultant printer plot and associated data are shown in Figs. F11.5.28 and F11.5.29. A second printer plot covering the same area shows a unit map rather than a mixture map. This unit map and associated data are shown in Figs. F11.5.30 and F11.5.31.

```
X-Y SLICE AT Z MIDPOINT.  SINGLE UNIT CONCENTRIC CYLS.

                                                MIXTURE MAP

MIXTURE    0   1   2   4
 SYMBOL        *   -   .

      UPPER LEFT          LOWER RIGHT
      COORDINATES         COORDINATES

X     -0.46000E+01         0.46000E+01
Y      0.46000E+01        -0.46000E+01
Z      0.0                 0.0

          U AXIS      V AXIS
          (DOWN)      (ACROSS)

X          0.0         1.00000
Y         -1.00000     0.0
Z          0.0         0.0

NU=   78   NV=  130     DELU= 1.17958E-01     DELV= 7.07658E-02
```

Fig. F11.5.28. Associated data for single unit mixture map

Figure F11.5.31 shows a block of 1's surrounded by a border of ?'s. This indicates that the entire slice specified in the plot data was part of Unit 1. For this problem the entire volume is Unit 1.

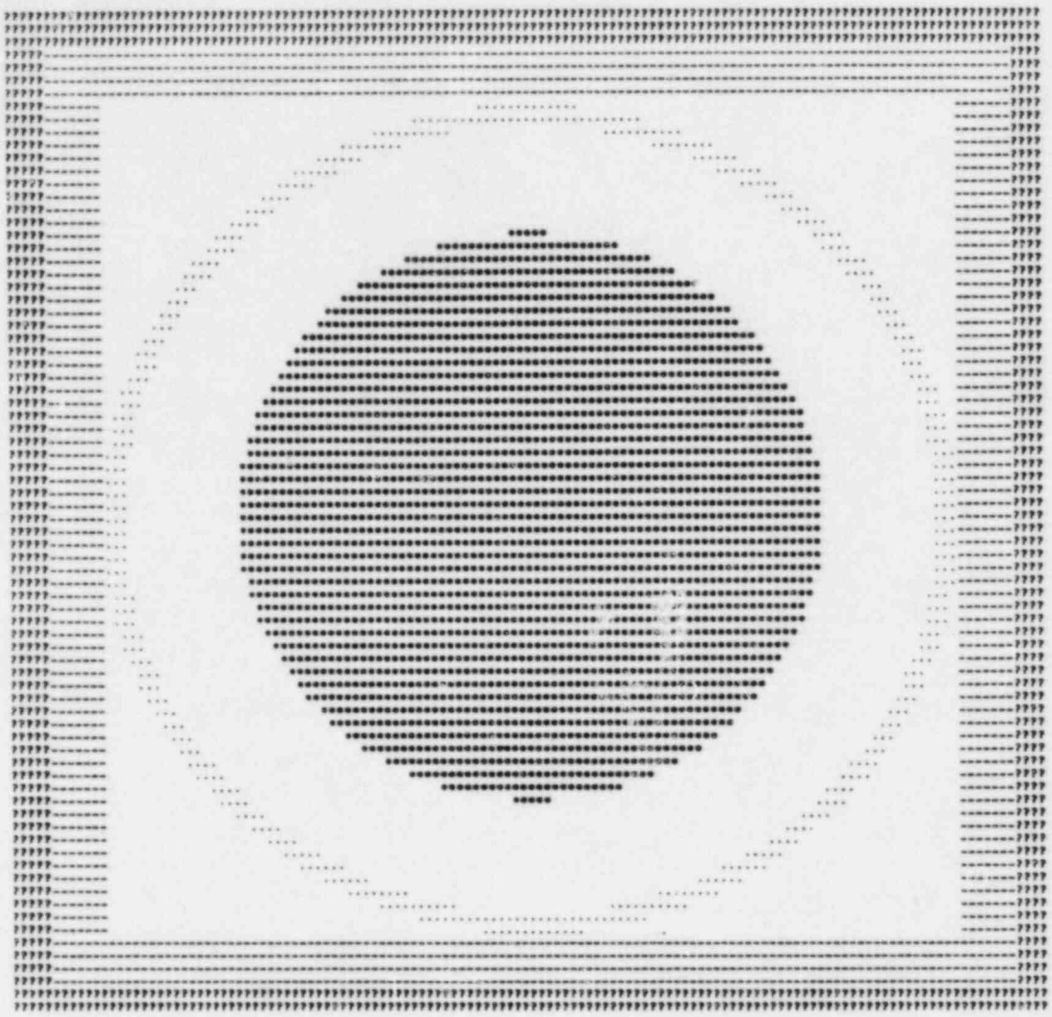


Fig. F11.5.29. Mixture map of single unit with centered origin

```
X-Y SLICE AT Z MIDPOINT.  SINGLE UNIT CONCENTRIC CYLS.
                                                  UNIT MAP

   UNIT    1
 SYMBOL    1

        UPPER  LEFT          LOWER RIGHT
        COORDINATES          COORDINATES

X      -0.14000E+01           0.14000E+01
Y       0.14000E+01          -0.14000E+01
Z       0.0                   0.0

             U AXIS       V AXIS
             (DOWN)       (ACROSS)

X            0.0           1.00000
Y           -1.00000       0.0
Z            0.0           0.0

NU=   70   NV=  130     DELU= 1.17955E-01     DELV= 7.07692E-02
```

Fig. F11.5.30. Associated data for single unit unit map

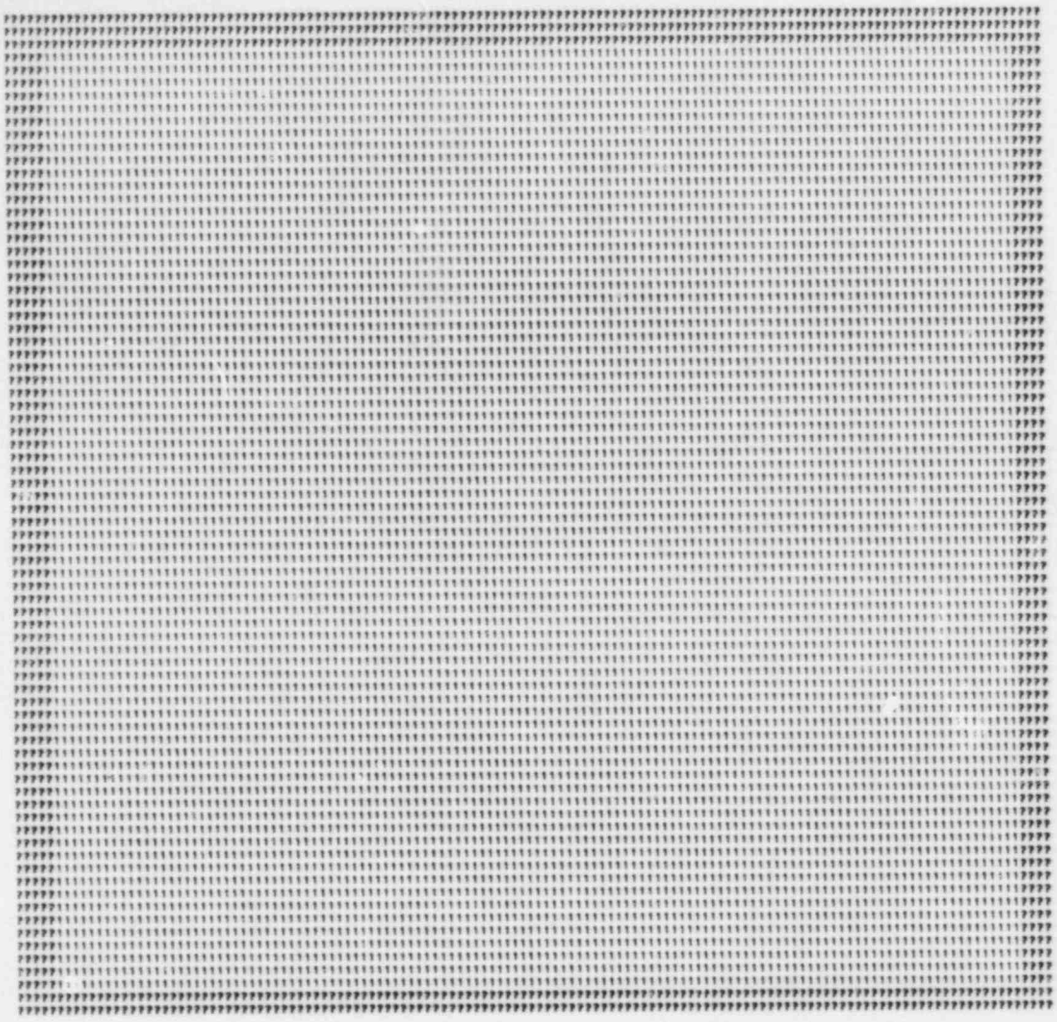


Fig. F11.5.31. Unit map printer plot for single unit with centered origin

EXAMPLE 2. SINGLE UNIT WITH OFFSET ORIGIN

The physical problem is the same as that described in Example 1, two concentric cylinders in a cuboid. The dimensions are exactly the same. The difference is in the choice of the origin. In this geometry description, the origin was specified as the most negative point of the unit, the lower left back corner. Thus, the cylinders have to have an origin specified to center them in the cuboid and the cuboid extends from 0.0 to 8.6 in x and y and from 0.0 to 32 in z as shown in the problem description below.

```
=KENO5
SINGLE UNIT CONCENTRIC CYLINDERS IN CUBOID WITH ORIGIN AT CORNER
READ PARAM  RUN=NO LIB=41 TME=0.5 END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2 MIX=2 200 1.0 MIX=3 502 0.1
MIX=4 200 1.0
END MIXT
READ GEOM
CYLINDER 1 1 2.6 31.0 1.0 ORIGIN 4.3 4.3
CYLINDER 0 1 3.6 31.0 1.0 ORIGIN 4.3 4.3
CYLINDER 4 1 3.8 31.0 1.0 ORIGIN 4.3 4.3
CUBOID 0 1 8.1 0.5 8.1 0.5 31.0 1.0
CUBOID 2 1 8.6 0.0 8.6 0.0 32.0 0.0
END GEOM
READ PLOT TTL='X-Y SLICE AT Z MIDPOINT.  SINGLE UNIT CONCENTRIC CYLS.'
XUL=-0.3 YUL=8.9 ZUL=16.0 XLR=8.9 YLR=-0.3 ZLR=16.0
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *-.X' END
PIC=UNIT NCH='01' END
END PLOT
END DATA
END
```

The plot data included above will draw a mixture map of an x-y slice taken at the half height (z=16.0). It will also draw a unit map of the same slice. The plot dimensions extend 0.3 cm beyond the problem dimensions to provide a border of question marks around the plot. The associated plot data specification for the mixture map is shown in Fig. F11.5.32, the mixture map is shown in Fig. F11.5.33, and the associated plot data for the unit map is shown in Fig. F11.5.34. The unit map is identical to Fig. F11.5.31 and is not included. Note that Fig. F11.5.33 is identical to Fig. F11.5.29.

```
X-Y SLICE AT Z MIDPOINT.  SINGLE UNIT CONCENTRIC CYLS.
                                        MIXTURE MAP

MIXTURE    0    1    2    4
 SYMBOL         *    -    .

        UPPER  LEFT          LOWER RIGHT
        COORDINATES          COORDINATES

X     -0.30000E+00          0.89000E+01
Y      0.89000E+01         -0.30000E+00
Z      0.16000E+02          0.16000E+02

             U AXIS       V AXIS
             (DOWN)       (ACROSS)

X             0.0          1.00000
Y            -1.00000      0.0
Z             0.0          0.0

/U=   78   NV=  130     DELU= 1.1795E-01     DELV= 7.8769E-02
```

Fig. F11.5.32. Associated data for mixture map of single unit with offset origin

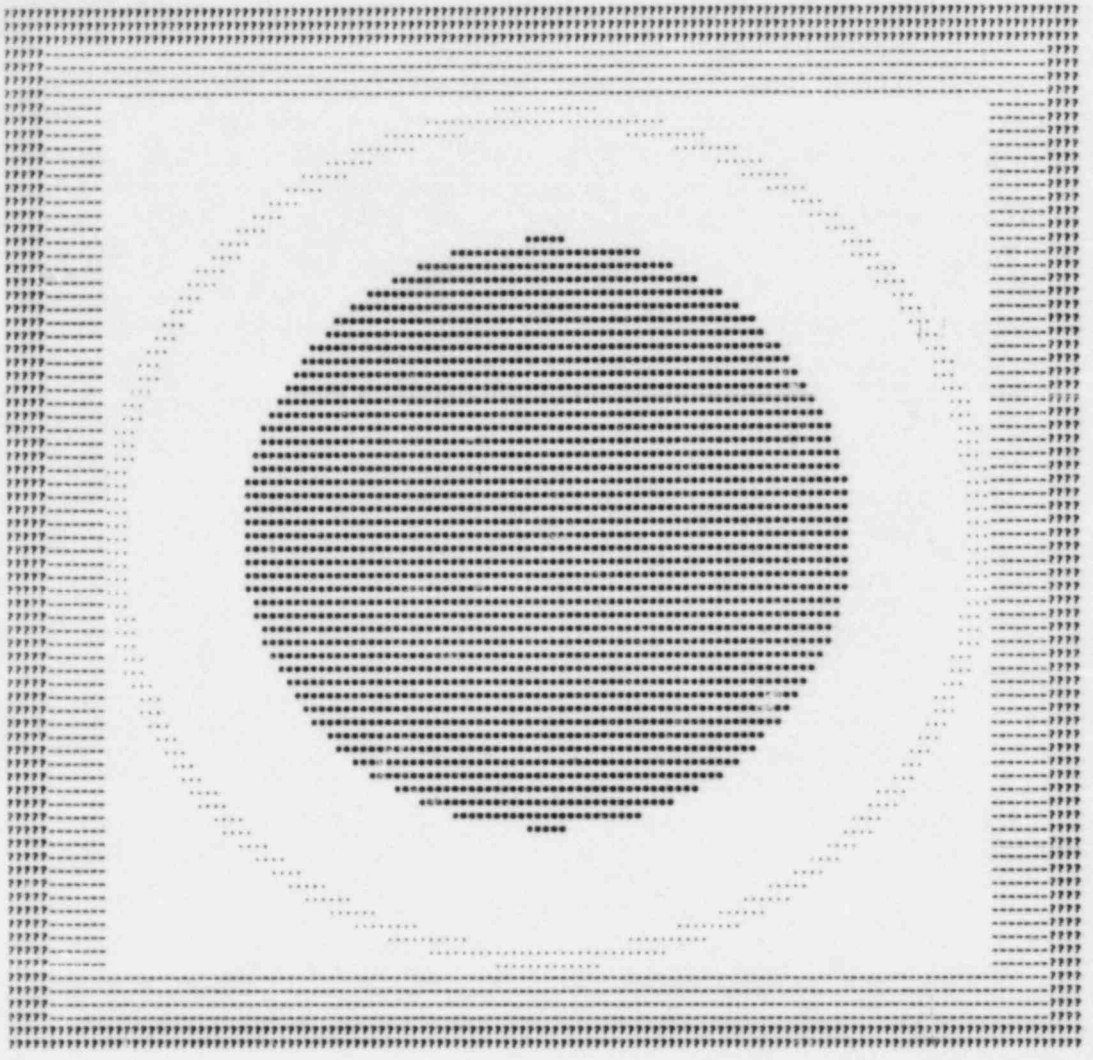
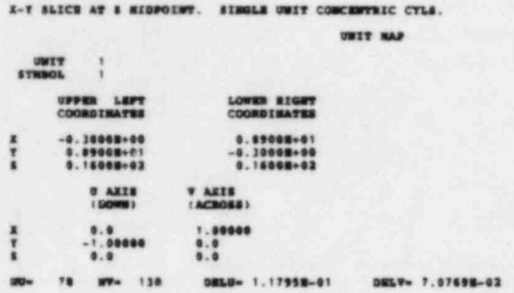


Fig. F11.5.33. Mixture map of single unit with offset origin



```
X-Y SLICE AT Z MIDPOINT.  SINGLE UNIT CONCENTRIC CYLS.
                                                UNIT MAP

   UNIT     1
 SYMBOL     1

        UPPER  LEFT          LOWER RIGHT
        COORDINATES          COORDINATES

X      -0.1000E+00           0.8900E+01
Y       0.8900E+01          -0.1000E+00
Z       0.1600E+02           0.1600E+02

             U AXIS       V AXIS
             (DOWN)       (ACROSS)

X             0.0          1.00000
Y            -1.00000      0.0
Z             0.0          0.0

NU=   78   NV=  130     DELU= 1.1795E-01     DELV= 7.0769E-02
```

Fig. F11.5.34. Associated data for unit map of single unit with offset origin

EXAMPLE 3. A 2x2x2 UNREFLECTED ARRAY OF CONCENTRIC CYLINDERS IN CUBOIDS

The physical representation of this example is a 2x2x2 array of the configuration described in Example 1 of this section. The input data description for this array is given below:

```
2x2x2 BARE ARRAY OF CONCENTRIC CYLINDERS IN CUBOID
READ PARAM  RUN=NO LIB=41 TME=8.5 END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2 MIX=2 200 1.0 MIX=3 502 0.1
MIX=4 200 1.0
END MIXT
READ GEOM
CYLINDER 1 1 2.6 2P15.0
CYLINDER 0 1 3.6 2P15.0
CYLINDER 4 1 3.8 2P15.0
CUBOID 0 1 4P3.8 2P15.0
CUBOID 2 1 4P4.3 2P16.0
END GEOM
READ ARRAY NUX=2 NUY=2 NUZ=2 END ARRAY
READ PLOT TTL='X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER.'
XUL=-0.3 YUL=17.5 ZUL=16.0 XLR=17.5 YLR=-0.3 ZLR=16.0
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *-.X' END
TTL='X-Z SLICE THROUGH FRONT ROW, Y=12.9.'
XUL=-1.0 YUL=12.9 ZUL=65.0 XLR=18.2 YLR=12.9 ZLR=-1.0
UAX=1.0 WDN=-1.0 NAX=60 END
END PLOT
END DATA
END
```

As stated at the beginning of Sect. F11.5.10, the origin of the plot is located at the lower left back corner of the array. Each individual unit in the array is 8.6 cm wide in x and y and is 32 cm high in z. Since the array has two units stacked in each direction, the array is 17.2 cm wide in x and y and is 64 cm high. Therefore, the array exists from x=0.0 to x=17.2, from y=0.0 to y=17.2 and from z=0.0 to z=64.0.

The first printer plot is to generate an x-y slice through the array at the half height of the first layer as shown in Fig. F11.5.35. This occurs at z=16.0 cm. It is desirable to define the outer boundaries of the array. This is achieved by setting the boundaries of the plot larger than the array. In this case, the boundaries were arbitrarily set 0.3 cm larger than the array, resulting in a border of question marks around the array. If the plot were to exclude everything external to the array, the following coordinates could have been entered XUL=0.0 YUL=17.2 ZUL=16.0 XLR=17.2 YLR=0.0 ZLR=16.0. This would have eliminated all the question marks. The existing picture was made using XUL=-0.3 YUL=17.5 ZUL=16.0 XLR=17.5 YLR=-0.3 ZLR=16.0.

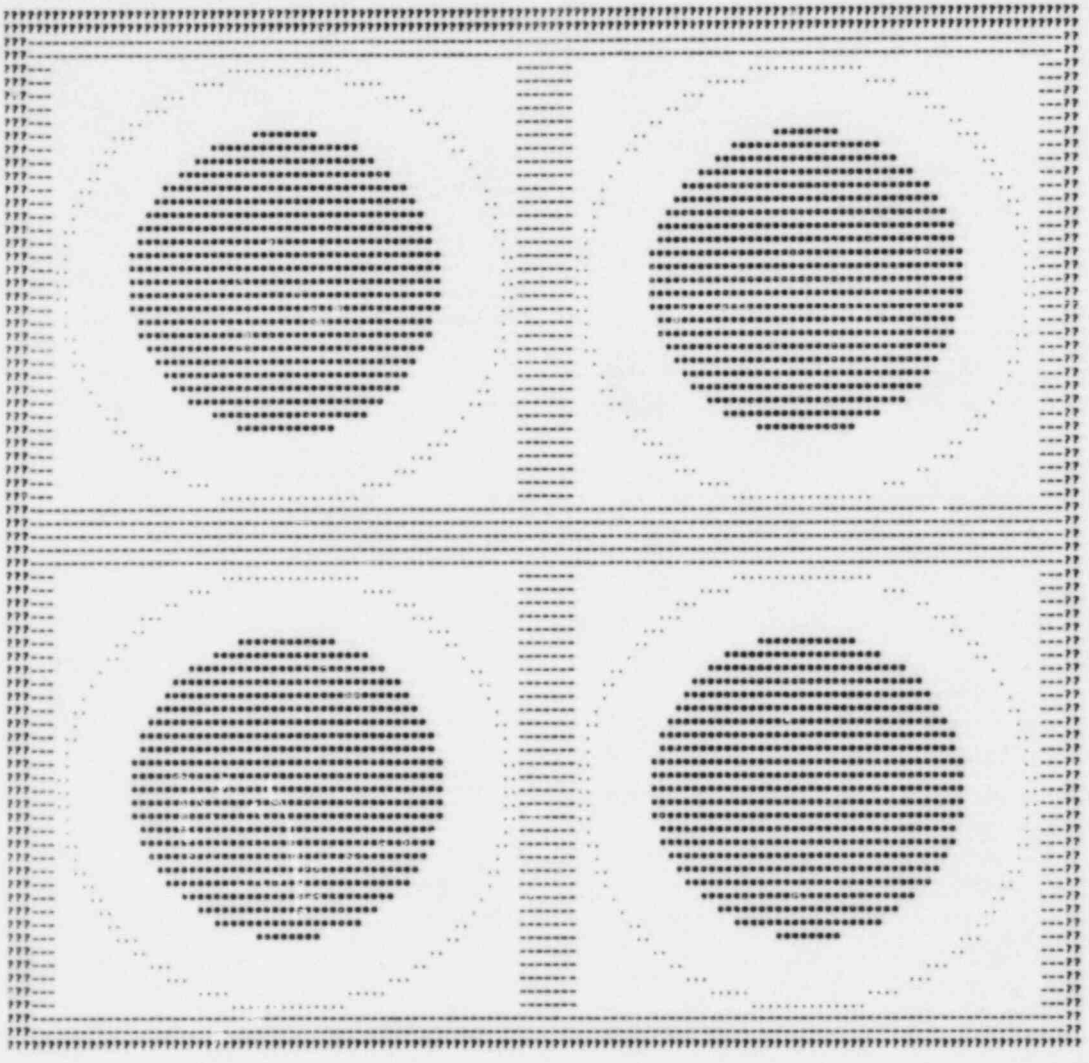


Fig. F11.5.35. x-y plot of 2x2x2 bare array

The second printer plot is to generate an x-z slice through the center of the front row of the array. In order to obtain a boundary of question marks, the coordinates of x and z were arbitrarily set 1.0 cm larger than the boundaries of the array. The center of the front row occurs at y=12.9. The coordinates of the plot were: XUL=-1.0 ZUL=65.0 YUL=12.9 XLR=18.2 ZLR=-1.0 YLR=12.9. The plot was defined to be half a page wide by entering NAX=60. The resultant mixture map is shown in Fig. F11.5.36.

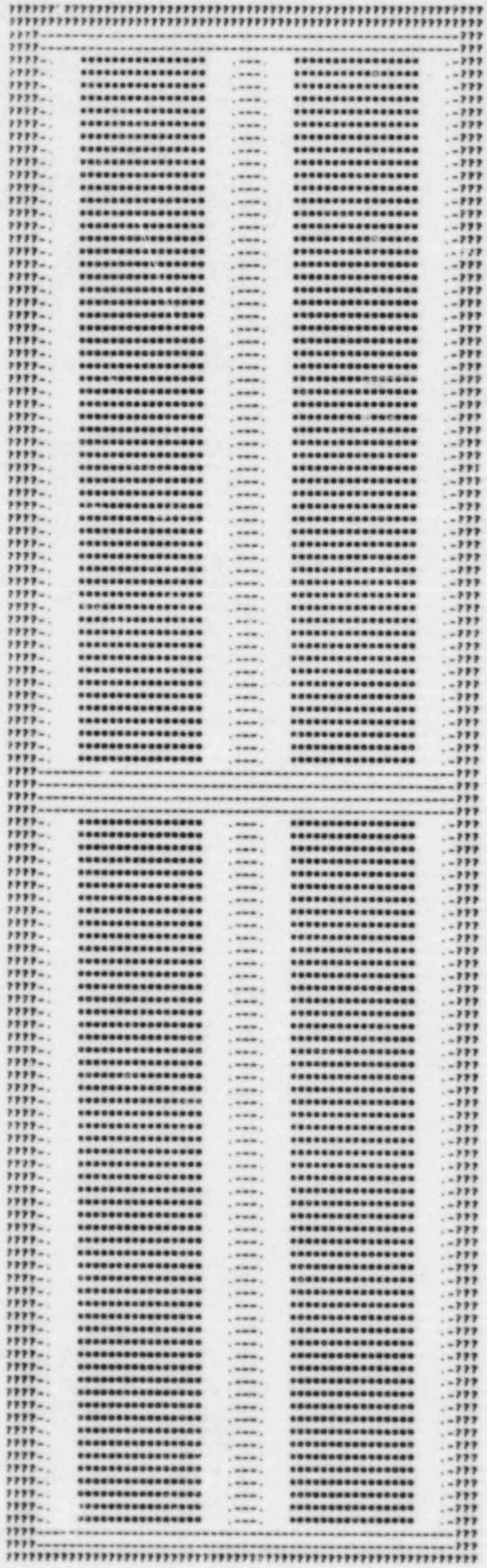


Fig. F11.5.36. x-z plot of 2x2x2 bare array

EXAMPLE 4. A 2x2x2 REFLECTED ARRAY WITH THE ORIGIN AT THE LOWER LEFT-HAND BACK CORNER OF THE ARRAY

The array is the array described in Example 3 of this section with a 6-in. concrete reflector on all faces. The input data description for this array is given below.

```
2x2x2 REFLECTED ARRAY OF CONCENTRIC CYLINDERS IN CUBOID
READ PARAM  RUN=NO LIB=41 TME=0.5 END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2 MIX=2 200 1.0 MIX=3 301 1.0
MIX=4 200 1.0
END MIXT
READ GEOM
CYLINDER 1 1 2.6 2P15.0
CYLINDER 0 1 3.6 2P15.0
CYLINDER 4 1 3.8 2P15.0
CUBOID 0 1 4P3.8 2P15.0
CUBOID 2 1 4P4.3 2P16.0
CORE 1 1 3*0.0
REFLECTOR 3 2 6*5.0 3
REFLECTOR 3 5 6*0.24 1
END GEOM
READ BIAS ID=301 2 5 END BIAS
READ ARRAY NUX=2 NUY=2 NUZ=2 END ARRAY
READ PLOT TTL='X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER.INCLUDES REFL.'
XUL=-16.24 YUL=33.44 ZUL=16.0 XLR=33.44 YLR=-16.24 ZLR=16.0
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *-.X' END
TTL='X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER, INCLUDE 3 CM OF REFL.'
XUL=-3.0 YUL=20.2 ZUL=16.0 XLR=20.2 YLR=-3.0 ZLR=16.0
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *-.X' END
TTL='X-Z SLICE THROUGH FRONT ROW, Y=12.9. INCLUDE REFLECTOR'
XUL=-16.24 YUL=12.9 ZUL=80.24 XLR=33.44 YLR=12.9 ZLR=-16.24
UAX=1.0 WDN=-1.0 NAX=130 END
TTL='X-Z SLICE THROUGH FRONT ROW, Y=12.9. INCLUDE 3 CM OF REFLECTOR'
XUL=-3.0 YUL=12.9 ZUL=67.0 XLR=20.2 YLR=12.9 ZLR=-3.0
UAX=1.0 WDN=-1.0 NAX=130 END
END PLOT
END DATA
END
```

The CORE card specifies the coordinates of the lower left back corner of the array to be (0.0,0.0,0.0). Thus the reflected array extends from -15.24 cm to +32.44 cm in x and y and from -15.24 to +79.24 in z.

The first printer plot for this example is to show an x-y slice through the array and reflector at the half height of the bottom layer. A border of question marks is used to verify that the entire reflector has been shown. This is accomplished by arbitrarily setting the picture boundaries 1 cm beyond the reflector boundaries. The coordinates used for this plot are: XUL=-16.24 YUL=33.44 ZUL=16.0 XLR=33.44 YLR=-16.24 ZLR=16.0. The plot data description is shown in Fig. F11.5.37, and the plot is shown in Fig. F11.5.38.

```
X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER.INCLUDES REFL.

                                        MIXTURE MAP

MIXTURE     0   1   2   3   4
 SYMBOL         *   -   .   X

        UPPER  LEFT          LOWER RIGHT
        COORDINATES          COORDINATES

X      -0.16248E+02          0.33448E+02
Y       0.33448E+02         -0.16248E+02
Z       0.16008E+02          0.16008E+02

            U AXIS       V AXIS
            (DOWN)       (ACROSS)

X           0.0          1.00000
Y          -1.00000      0.0
Z           0.0          0.0

NU=   76   NV=  130     DELU= 6.34928E-01     DELV= 3.82158E-01
```

Fig. F11.5.37. Plot data for x-y slice of example 4

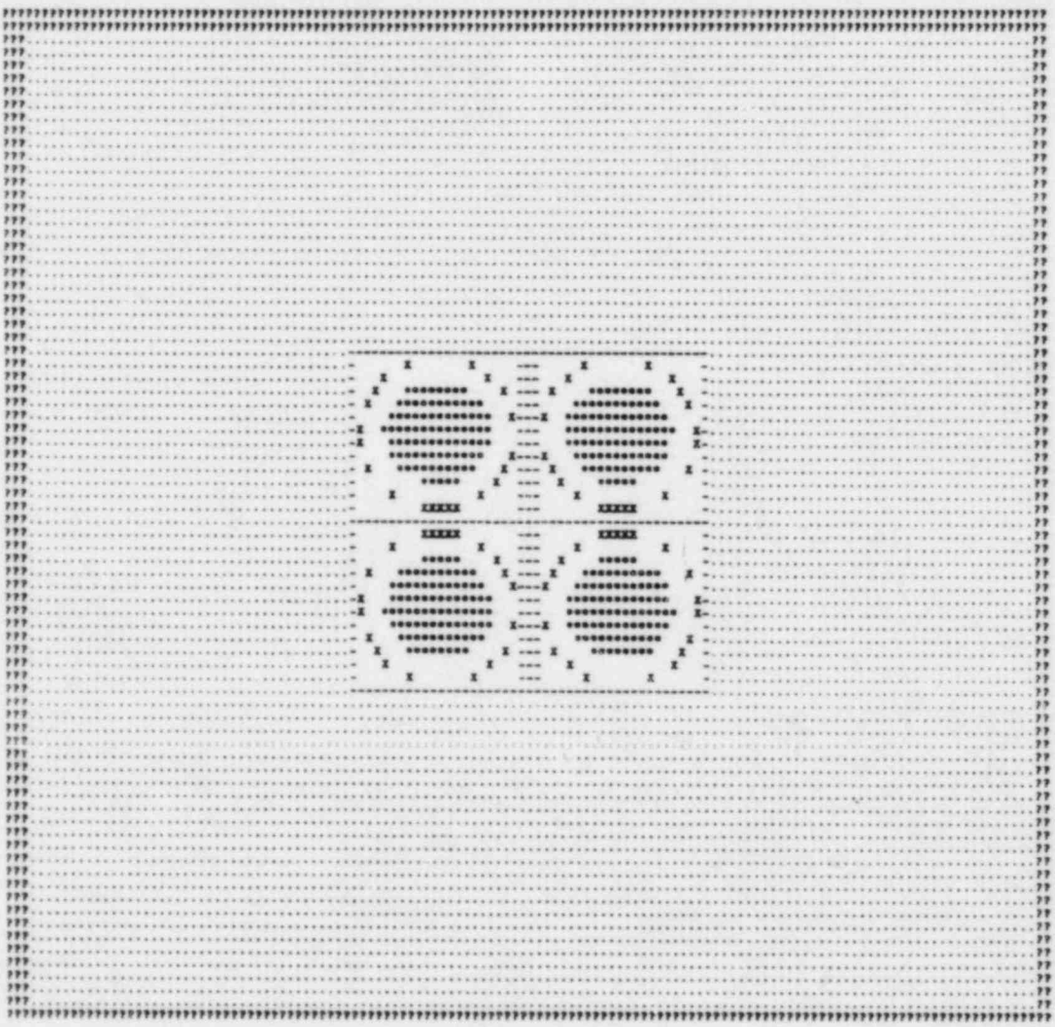


Fig. F11.5.38. x-y plot of 2x2x2 reflected array

The next printer plot is the same as the previous plot except the plot includes only the first 3 cm of the reflector. This results in the picture being large enough to show more detail. The coordinates used for this plot are: XUL=-3.0 YUL=20.2 ZUL=16.0 XLR=20.2 YLR=-3.0 ZLR=16.0. This plot data description is given in Fig. F11.5.39, and the plot is shown in Fig. F11.5.40.

```
X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER. INCLUDE 3 CM OF REFL.

                                MIXTURE MAP

MIXTURE    0   1   2   3   4
 SYMBOL        *   -   .   X

        UPPER LEFT              LOWER RIGHT
        COORDINATES             COORDINATES

X      -0.3000E+01               0.2020E+01
Y       0.2020E+02              -0.3000E+01
Z       0.1600E+02               0.1600E+02

             U AXIS      V AXIS
             (DOWN)      (ACROSS)

X             0.0         1.00000
Y            -1.00000     0.0
Z             0.0         0.0

NU=   78   NV=  130     DELU= 2.9744E-01     DELV= 1.7846E-01
```

Fig. F11.5.39. Plot data for enlarged x-y slice of example 4

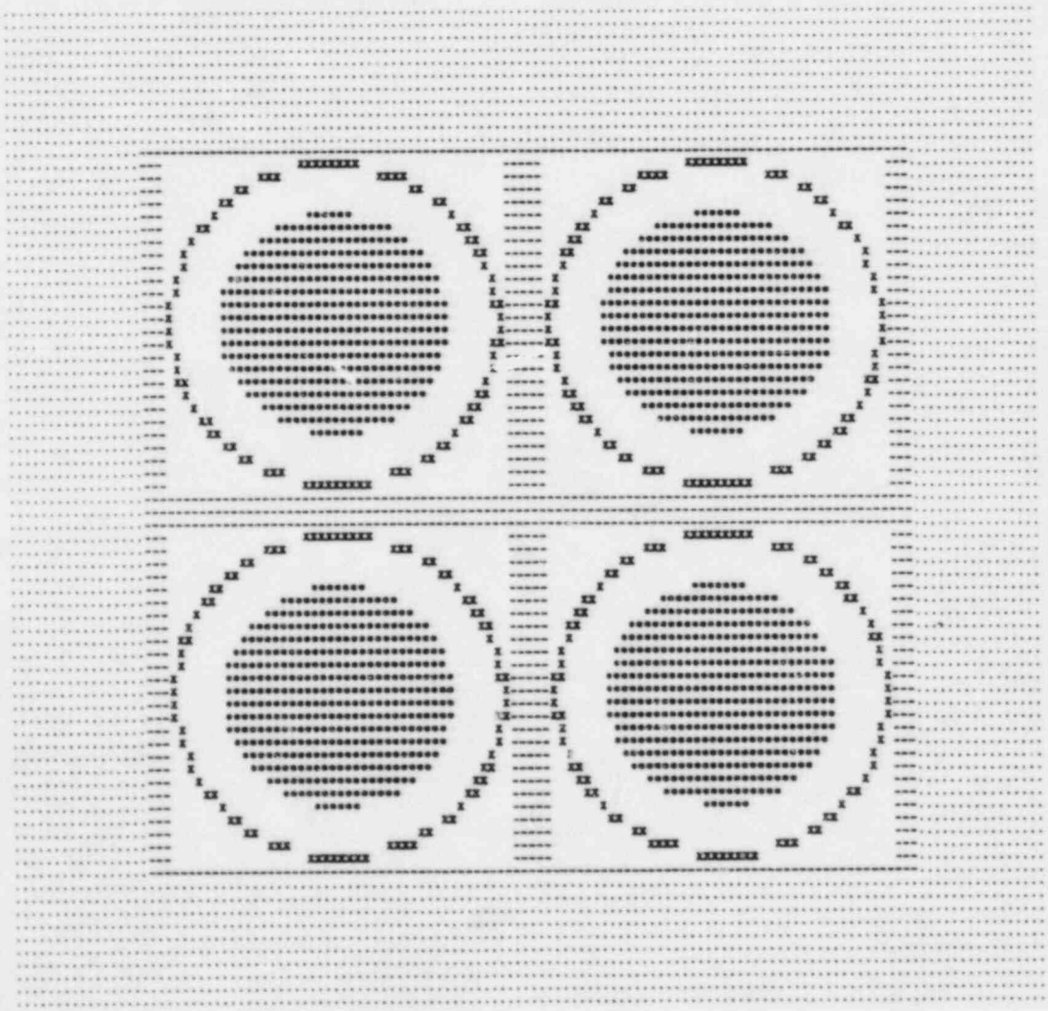


Fig. F11.5.40. Enlarged x-y plot of 2x2x2 reflected array

The third printer plot for this example is an x-z slice through the center of the front row. An extra 1 cm is included in the coordinates to provide a border of question marks around the plot. The coordinates are: XUL=-16.24 YUL=12.9 ZUL=80.24 XLR=33.44 YLR=12.9 ZLR=-16.24. The resultant plot data and plot are shown in Figs. F11.5.41 and F11.5.42.

```
X-Z SLICE THROUGH FRONT ROW, Y=12.9. INCLUDE REFLECTOR

                                        MIXTURE MAP

MIXTURE     0   1   2   3   4
 SYMBOL         *   -   .   X

       UPPER  LEFT              LOWER RIGHT
       COORDINATES              COORDINATES

X     -0.1624E+02                0.3344E+02
Y      0.1290E+02                0.1290E+02
Z      0.8024E+02               -0.1624E+02

            U AXIS          V AXIS
            (DOWN)          (ACROSS)

X           0.0              1.00000
Y           0.0              0.0
Z          -1.00000          0.0

NU=  151   NV=  136     DELU= 6.3692E-01     DELV= 3.8215E-01
```

Fig. F11.5.41. Plot data for x-z siice of example 4

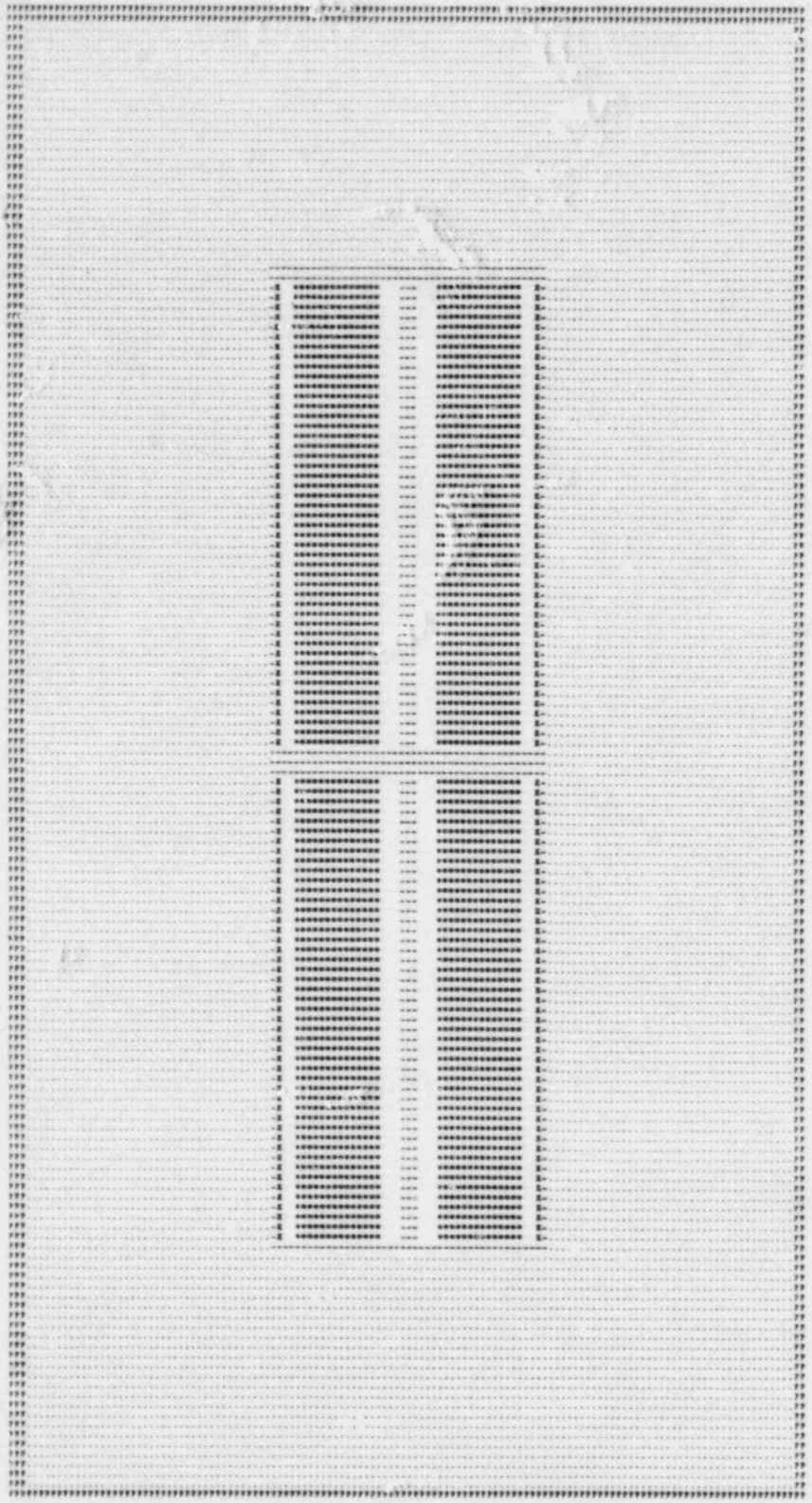


Fig. F11.5.42. x-z plot for 2x2x2 reflected array

The last printer plot for this example is the same as the previous one except only 3 cm of the reflector is included in the plot. The plot data and associated plot are shown in Figs. F11.5.43 and F11.5.44.

```
X-Z SLICE THROUGH FRONT ROW, Y=12.9. INCLUDE 3 CM OF REFLECTOR

                                    MIXTURE MAP

MIXTURE     0   1   2   3   4
 SYMBOL         +   -   .   X

      UPPER  LEFT          LOWER RIGHT
      COORDINATES          COORDINATES

X     -0.30000E+01          0.20300E+02
Y      0.12900E+02          0.12900E+02
Z      0.67000E+02         -0.30000E+01

          U AXIS       V AXIS
          (DOWN)       (ACROSS)

X          0.0          1.00000
Y          0.0          0.0
Z         -1.00000      0.0

MU=  235   MV=  130     DELU= 2.97448E-01     DELV= 1.78468E-01
```

Fig. F11.5.43. Plot data for enlarged x-z slice of example 4

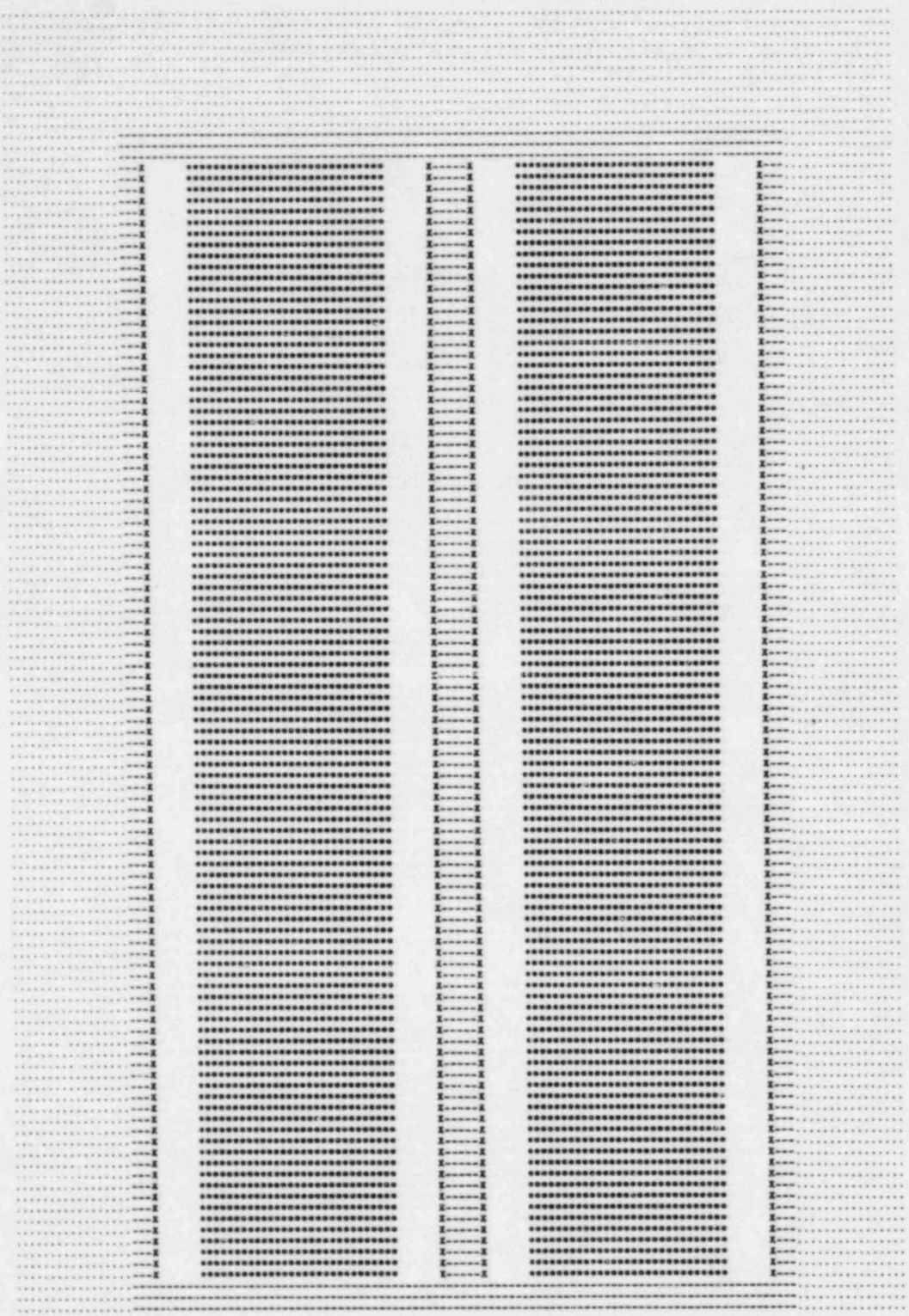


Fig. F11.5.44. Enlarged x-z plot of 2x2x2 reflected array

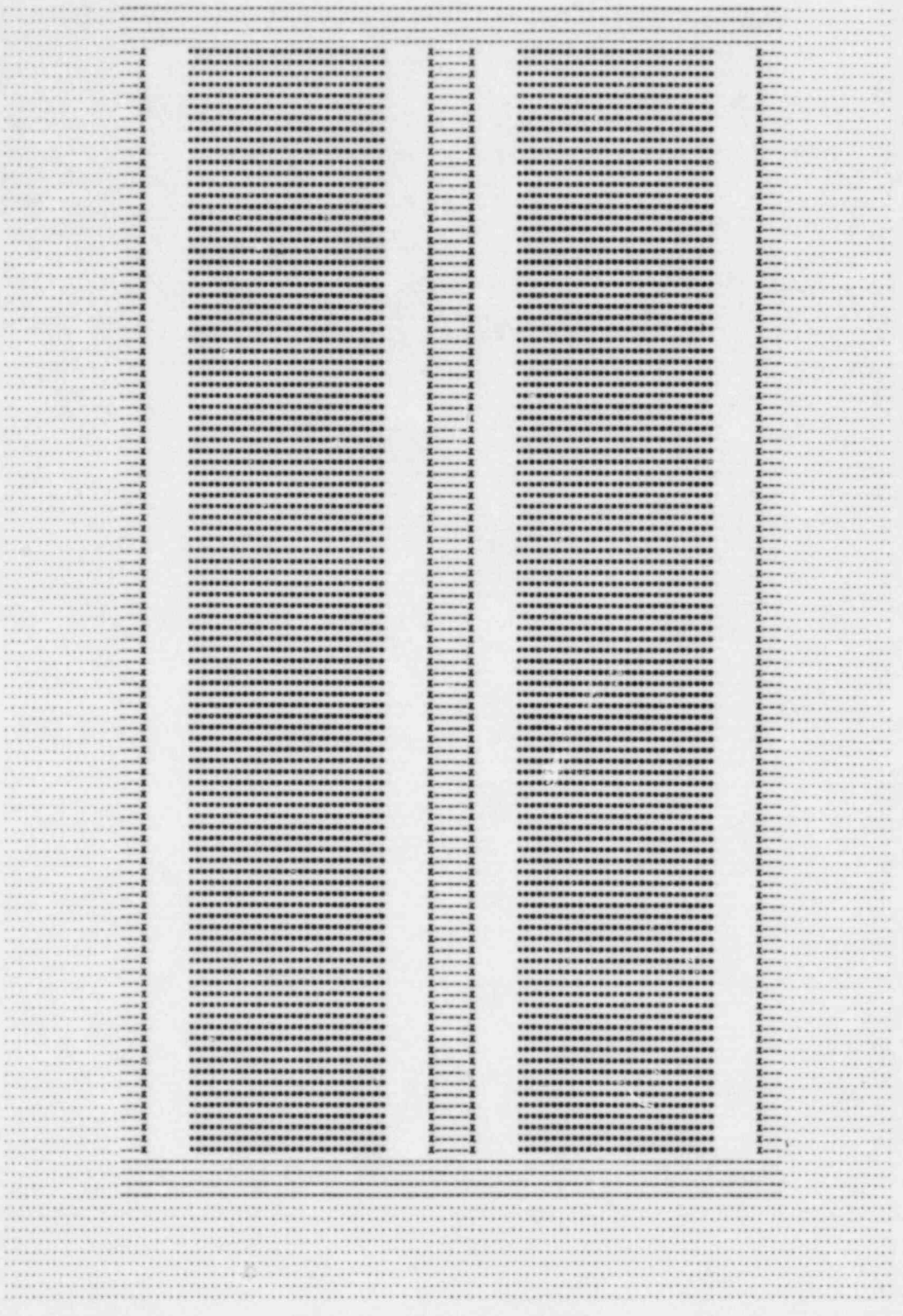


Fig. F11.5.44 (cont.)

EXAMPLE 5. A 2x2x2 REFLECTED ARRAY WITH THE ORIGIN CENTERED IN THE ARRAY

This example is physically identical to Example 4. The difference is in the specification of the origin. The bare array is 17.2 cm wide in x and y and 64 cm high. The origin (0,0,0) can be placed at the exact center of the array by specifying the lower left back corner of the array as x=-8.6, y=-8.6 and z=-32.0. This is done using the CORE description. Because the origin is located at a different position, the coordinates of the plots will also be different. The input data description for this example is given below.

```
2x2x2 REFLECTED ARRAY OF CONCENTRIC CYLINDERS IN CUBOID
READ PARAM  RUN=NO LIB=41 TME=0.5 END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2 MIX=2 200 1.0 MIX=3 301 1.0
MIX=4 200 1.0
END MIXT
READ GEOM
CYLINDER 1 1 2.6 2P15.0
CYLINDER 0 1 3.6 2P15.0
CYLINDER 4 1 3.8 2P15.0
CUBOID 0 1 4P3.8 2P15.0
CUBOID 2 1 4P4.3 2P16.0
CORE 1 1 2*-8.6 -32.0
REFLECTOR 3 2 6*5.0 3
REFLECTOR 3 5 6*0.24 1
END GEOM
READ BIAS ID=301 2 5 END BIAS
READ ARRAY NUX=2 NUY=2 NUZ=2 END ARRAY
READ PLOT TTL='X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER.INCLUDES REFL.'
XUL=-24.84 YUL=24.84 ZUL=-8.0 XLR=24.84 YLR=-24.84 ZLR=-8.0
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *=.X' END
TTL='X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER, INCLUDE 3 CM OF REFL.'
XUL=-11.6 YUL=11.6 ZUL=-8.0 XLR=11.6 YLR=-11.6 ZLR=-8.0
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *-.X' END
TTL='X-Z SLICE THROUGH FRONT ROW. Y=4.3  INCLUDE REFLECTOR'
XUL=-24.84 YUL=4.3 ZUL=48.24 XLR=24.84 YLR=4.3 ZLR=-48.24
UAX=1.0 WDN=-1.0 NAX=130 END
TTL='X-Z SLICE THROUGH FRONT ROW, Y=4.3  INCLUDE 3 CM OF REFLECTOR'
XUL=-11.6 YUL=4.3 ZUL=35.0 XLR=11.6 YLR=4.3 ZLR=-35.0
UAX=1.0 WDN=-1.0 NAX=130 END
END PLOT
END DATA
END
```

The first printer plot for this example covers identically the same area as the first printer plot for Example 4. The plot data for this plot and the actual plot are given in Figs. F11.5.45 and F11.5.46.

```
X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER. INCLUDES REFL.

                                    MIXTURE MAP

MIXTURE    0   1   2   3   4
 SYMBOL        +   -   .   X

      UPPER LEFT          LOWER RIGHT
      COORDINATES         COORDINATES

X     -0.24848E+02         0.24848E+02
Y      0.24848E+02        -0.24848E+02
Z     -0.80008E+01        -0.80008E+01

          U AXIS      V AXIS
          (DOWN)      (ACROSS)

X          0.0         1.00000
Y         -1.00000      0.0
Z          0.0          0.0

NU=   78   NV=  130     DELU= 6.36928E-01     DELV= 3.82158E-01
```

Fig. F11.5.45. Plot data for x-y slice of example 5

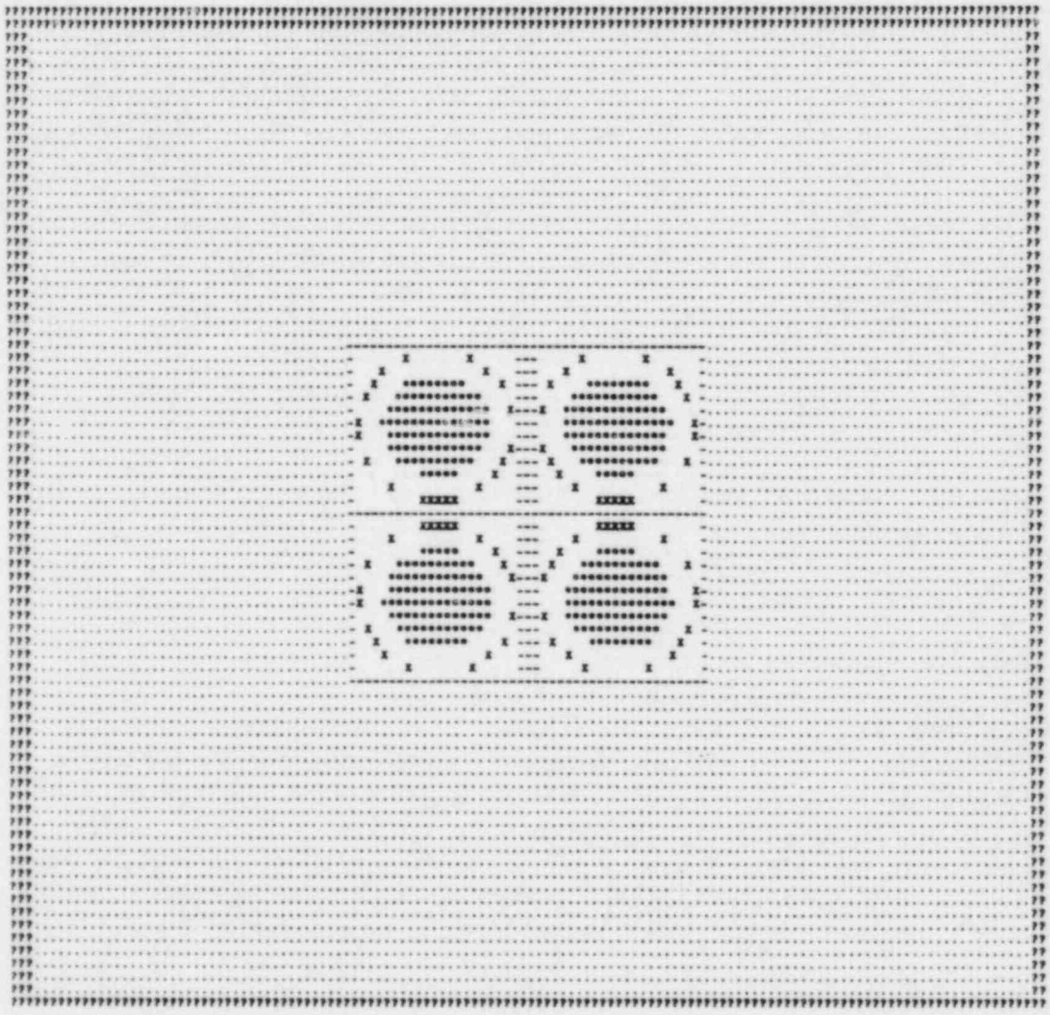


Fig. F11.5.46. x-y plot of 2x2x2 reflected array with centered origin

The Example 5 plot data and associated plots for an enlarged x-y plot, an x-z plot and an enlarged x-z plot are given in Figs. F11.5.47 through F11.5.52.

```
X-Y SLICE AT HALF HEIGHT OF BOTTOM LAYER. INCLUDE 1 CM OF REFL.

                                    MIXTURE MAP

MIXTURE    0   1   2   3   4
 SYMBOL        +   -   .   X

      UPPER  LEFT          LOWER RIGHT
      COORDINATES          COORDINATES

X     -0.11608E+02          0.11608E+02
Y      0.11608E+02         -0.11608E+02
Z     -0.80000E+01         -0.80000E+01

         U AXIS      V AXIS
         (DOWN)      (ACROSS)

X         0.0        1.00000
Y        -1.00000    0.0
Z         0.0        0.0

MU=   78   MV=  130     DELU= 2.97448E-01     DELV= 1.78468E-01
```

Fig. F11.5.47. Plot data for an enlarged x-y slice of example 5

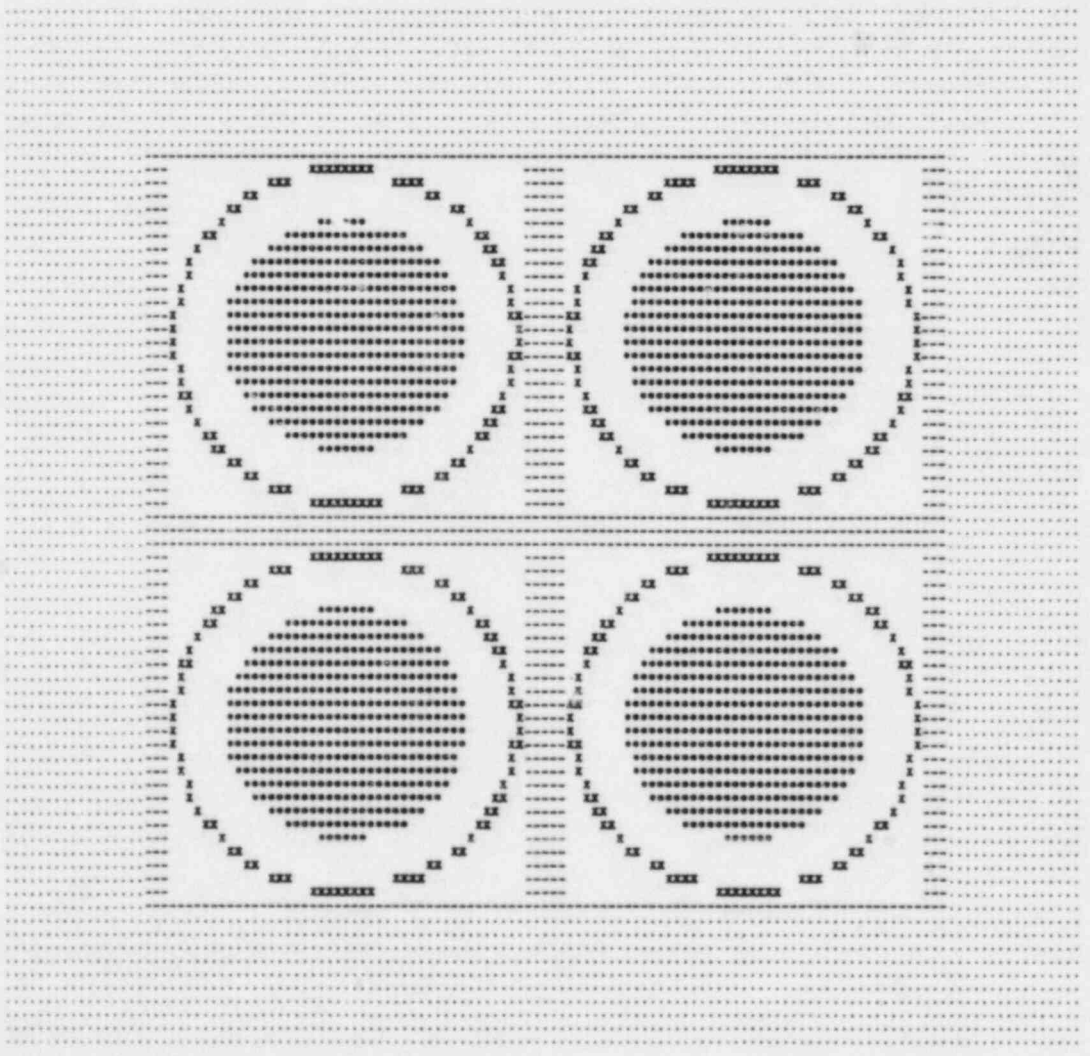


Fig. F11.5.48. Enlarged x-y plot of 2x2x2 reflected array with centered origin

```
X-Z SLICE THROUGH FRONT ROW, Y=4.3  INCLUDE REFLECTOR

                                        MIXTURE MAP

MIXTURE     0   1   2   3   4
 SYMBOL         +   -   .   X

      UPPER LEFT            LOWER RIGHT
      COORDINATES           COORDINATES

X     -0.2484E+02            0.2484E+02
Y      0.4300E+01            0.4300E+01
Z      0.4824E+02           -0.4824E+02

          U AXIS          V AXIS
          (DOWN)          (ACROSS)

X          0.0             1.00000
Y          0.0             0.0
Z         -1.00000         0.0

###=  151   ###=  130     DELU= 6.369##-01     DELV= 3.8215E-01
```

Fig. F11.5.49. Plot data for x-z slice of example 5

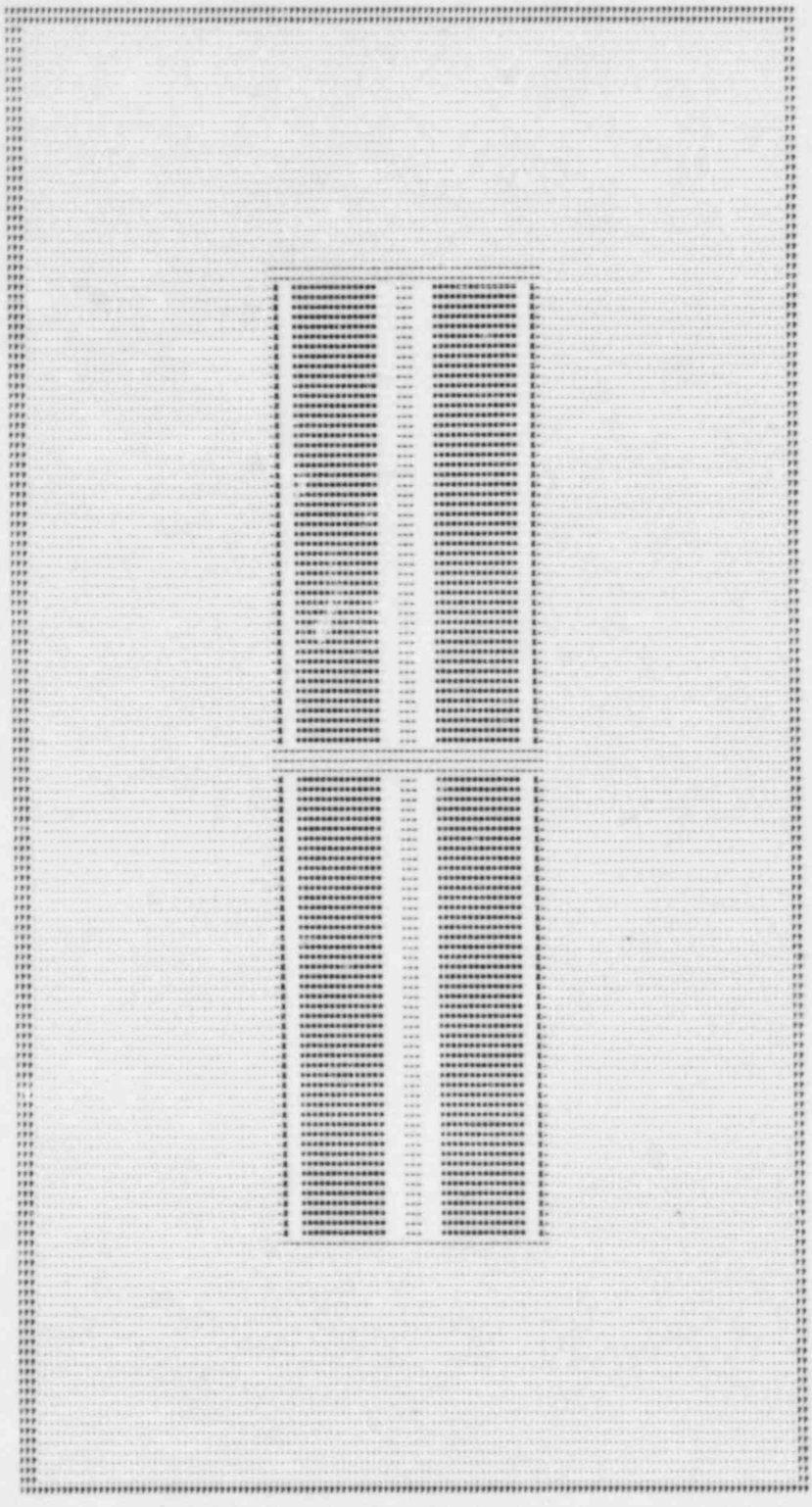


Fig. F11.5.50. x-z plot of reflected 2x2x2 array with centered origin

```
X-Z SLICE THROUGH FRONT ROW, Y=4.3  INCLUDE 1 CM OF REFLECTOR

                                    MIXTURE MAP

MIXTURE     0   1   2   3   4
 SYMBOL         *   -   .   X

      UPPER LEFT              LOWER RIGHT
      COORDINATES             COORDINATES

X     -0.11608E+02             0.11608E+02
Y      0.43008E+01             0.43008E+01
Z      0.35008E+02            -0.35008E+02

          U AXIS          V AXIS
          (DOWN)          (ACROSS)

X          0.0             1.00000
Y          0.0             0.0
Z         -1.00000         0.0

MU=  235   MV=  130     DELU= 2.97448E-01     DELV= 1.78468E-01
```

Fig. F11.5.51. Plot data for enlarged x-z slice of example 5

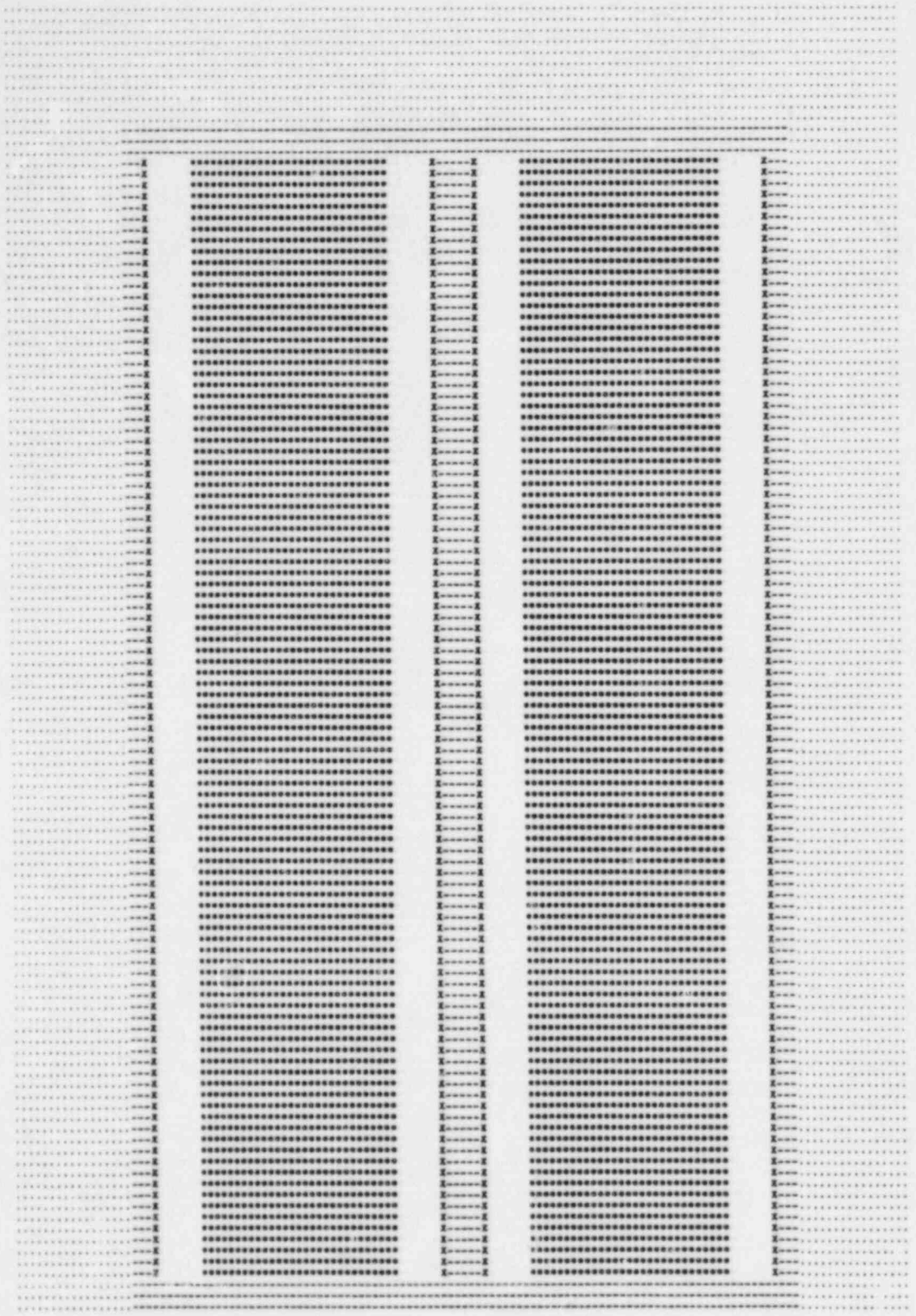


Fig. F11.5.52. Enlarged x-z plot of reflected 2x2x2 array with centered origin

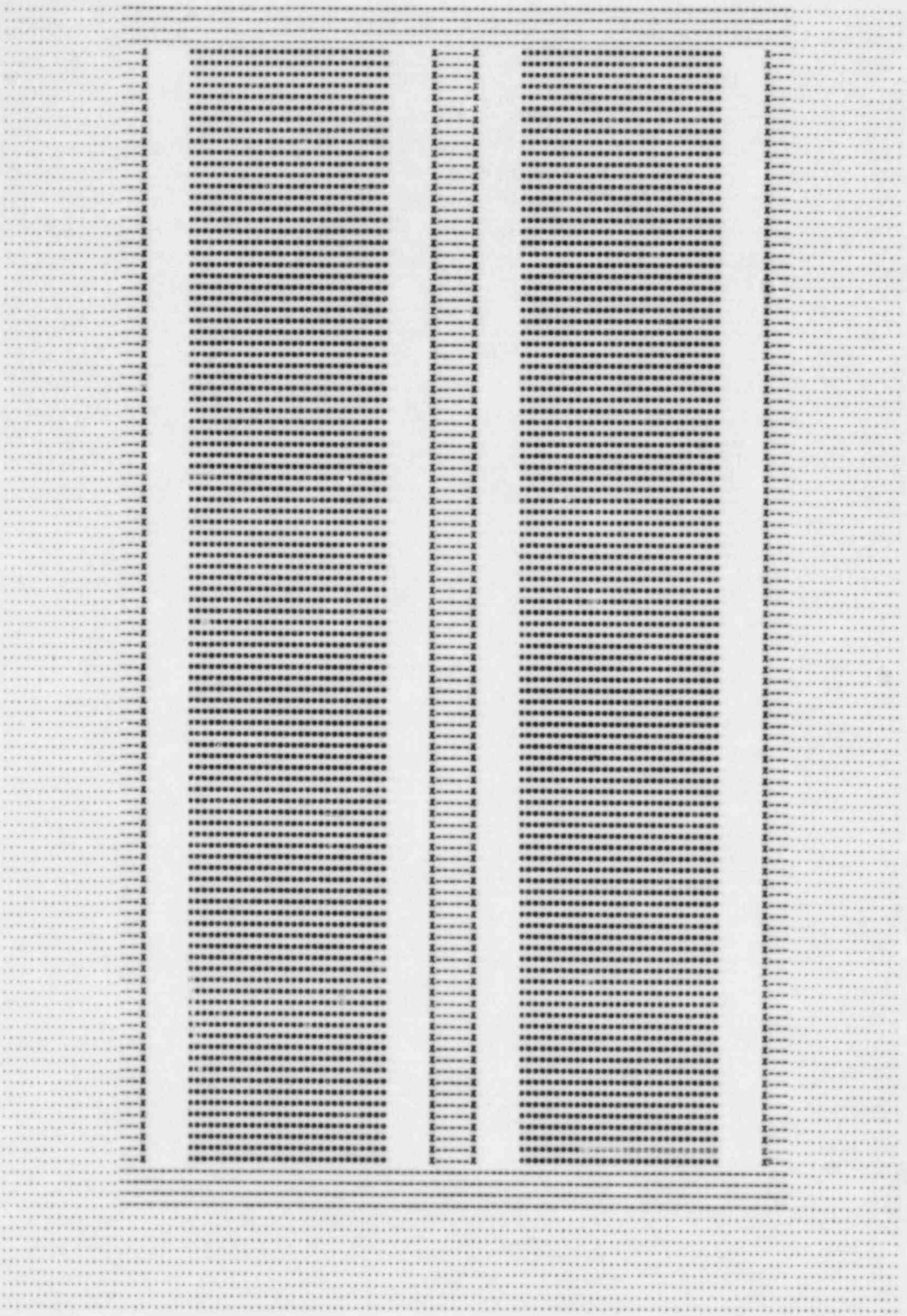


Fig. F11.5.52 (cont.)

EXAMPLE 6. NESTED HOLES

Refer to the nested hole description of Sect. F11.5.6.2. The mixture map for this problem is shown in two parts in Figs. F11.5.11 and F11.5.12. This example is one that involves a reasonably complicated placement of units or box types. Therefore, it might be useful to the user to generate both a mixture map and a unit map for the problem. Because an enlarged printer plot of the mixture map was already presented in Figs. F11.5.11 and F11.5.12, the mixture map in this section was reduced to fit on one page width by changing NAX from 260 to 130. The resultant mixture map is shown in Fig. F11.5.53 and the unit map is shown in Fig. F11.5.54. The data description for Example 6 follows.

```
=KENO5
NESTED HOLES SAMPLE
READ PARAM  RUN=NO LIB=41 TME=0.5 END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2 MIX=2 200 1.0 MIX=3 502 0.1
MIX=4 200 1.0
END MIXT
READ GEOM
UNIT 1
CYLINDER 1 1 0.1 2P15.0
UNIT 2
CUBOID 2 1 2P0.1 2P0.05 2P15.0
UNIT 3
CUBOID 2 1 2P0.05 2P0.1 2P15.0
UNIT 4
CYLINDER 1 1 0.1 2P15.0
CYLINDER 3 1 0.5 2P15.0
HOLE 1  0.0 -0.4 0.0
HOLE 1  0.4  0.0 0.0
HOLE 1  0.0  0.4 0.0
HOLE 1 -0.4  0.0 0.0
HOLE 2 -0.2  0.0 0.0
HOLE 2  0.2  0.0 0.0
HOLE 3  0.0 -0.2 0.0
HOLE 3  0.0  0.2 0.0
UNIT 5
CYLINDER 1 1 0.5 2P15.0
UNIT 6
CYLINDER 2 1 0.2 2P15.0
UNIT 7
CYLINDER 2 1 0.2 2P15.0
CYLINDER 0 1 1.3 2P15.0
HOLE 5 0.7071068 2*0.0
HOLE 6 0.7071068 0.7071068 0.0
HOLE 4 0.0 0.7071068 0.0
HOLE 6 -0.7071068 0.7071068 0.0
HOLE 5 -0.7071068 0.0 0.0
HOLE 6 -0.7071068 -0.7071068 0.0
HOLE 4 0.0 -0.7071068 0.0
HOLE 6 0.7071068 -0.7071068 0.0
CYLINDER 4 1 1.4 2P15.0
UNIT 8
CYLINDER 2 1 0.6 2P15.0
UNIT 9
CYLINDER 2 1 0.6 2P15.0
CYLINDER 0 1 3.6 2P15.0
HOLE 7 2.0 0.0 0.0
HOLE 8 2*2.0 0.0
HOLE 7 0.0 2.0 0.0
HOLE 8 -2.0 2.0 0.0
HOLE 7 -2.0 2*0.0
HOLE 8 2*-2.0 0.0
HOLE 7 0.0 -2.0 0.0
HOLE 8 2P2.0 0.0
CYLINDER 4 1 3.8 2P15.0
CUBOID 0 1 4P4.0 2P16.0
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=1  FILL 9 END ARRAY
READ PLOT TTL='X-Y SLICE AT Z MIDPOINT.  NESTED HOLES'
XUL=-0.1 YUL=8.1 ZUL=16.0 XLR=8.1 YLR=-0.1 ZLR=16
UAX=1.0 VDN=-1.0 NAX=130 NCH=' *-.X' END
TTL='X-Y SLICE AT Z MIDPOINT.  NESTED HOLES.  UNIT MAP'
NCH='*1-.X5678 '
PIC=UNIT  END
END PLOT
END DATA
END
```

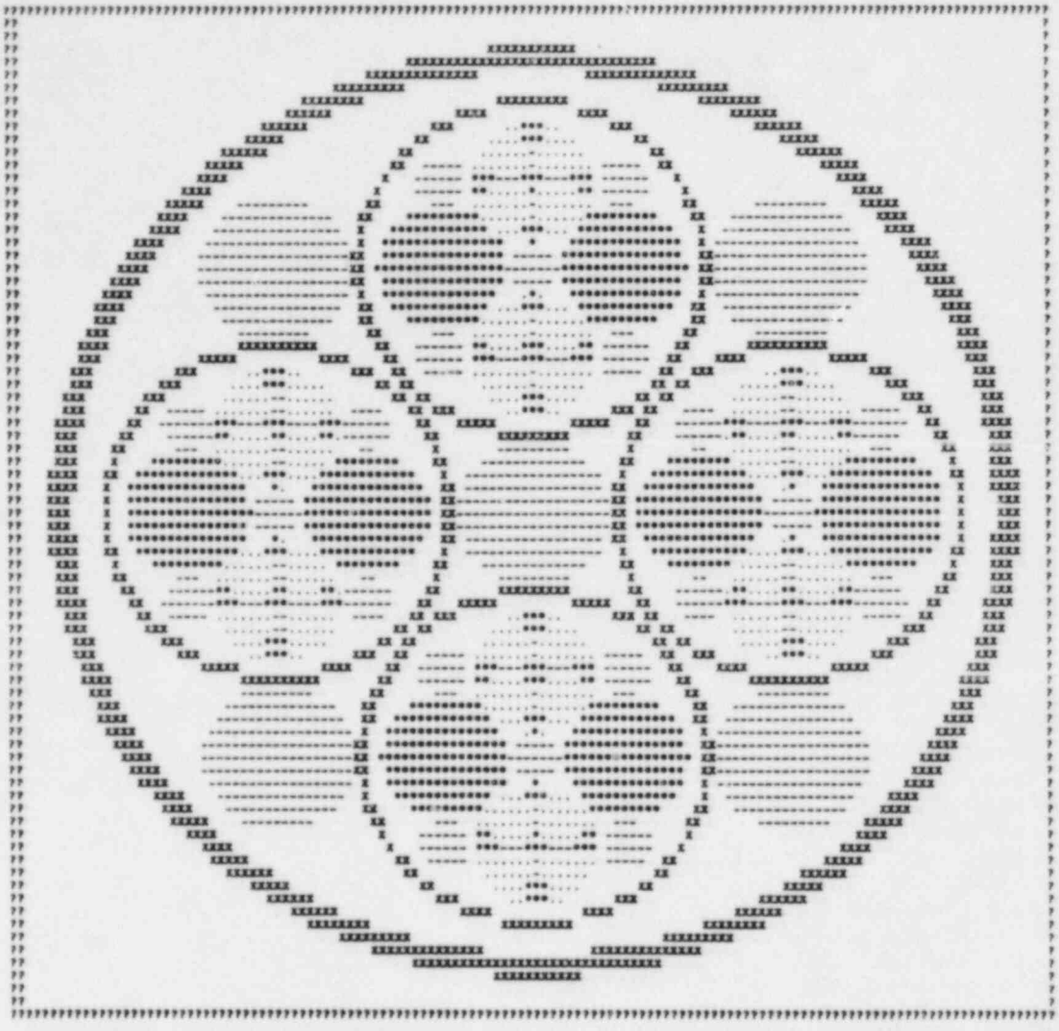


Fig. F11.5.53. Mixture map of nested holes problem

```
X-Y SLICE AT K MIDPOINT.  NESTED HOLES.  UNIT MAP

                                         UNIT MAP

  UNIT     1   2   3   4   5   6   7   8   9
SYMBOL     1   -   .   X   5   6   7   8

      OVERALL SYSTEM COORDINATES:
      XMIN= 0.0            XMAX= 0.000000E+00   YMIN= 0.0        YMAX= 0.000000E+00   ZMIN= 0.0          ZMAX= 1.200000E+01

      UPPER  LEFT          LOWER RIGHT
      COORDINATES          COORDINATES

X     -0.10000E+00          0.81000E+01
Y      0.81000E+01         -0.10000E+00
Z      0.16000E+02          0.16000E+02

          U AXIS       V AXIS
          (DOWN)       (ACROSS)

X          0.0          1.00000
Y         -1.00000      0.0
Z          0.0          0.0

NU=   78   NV=  130     DELU= 1.05113E-01     DELV= 6.30775E-02
```

Fig. F11.5.54. Plot data for unit map of nested holes

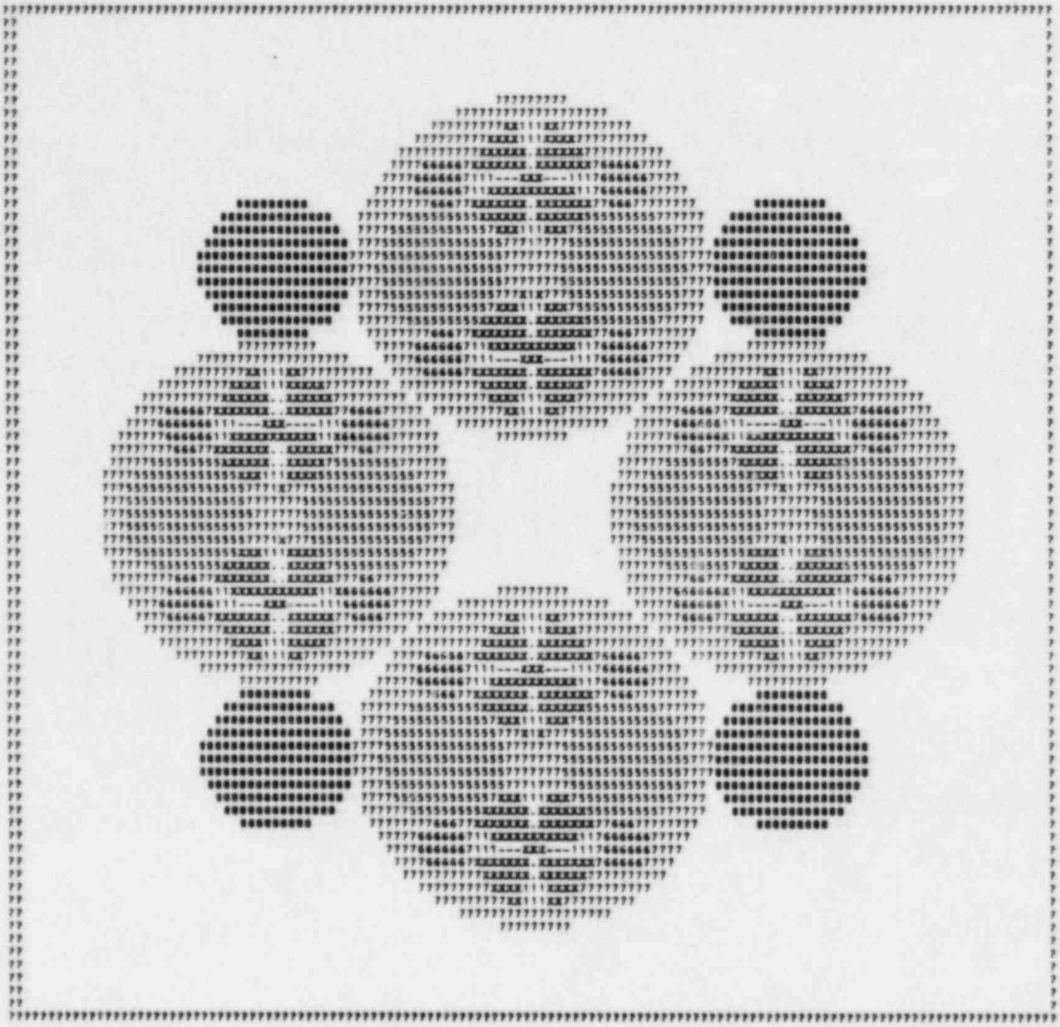


Fig. F11.5.55. Unit map of nested holes problem

The plot data description shown in Fig. F11.5.54 shows the symbols that represent each unit in the geometry mock-up. The unit map is shown in Fig. F11.5.55. The user can utilize this map to verify the correct placement of the units. Note that the unit map prints the units that are present at the deepest nesting level for the two-dimensional slice. It does not show any detail within a unit. The apparent detail within Unit 7 are Units 1, 2, 3, 4, 5, and 6 that were placed there via the hole option. Unit 1 is represented by "1", Unit 2 by "-", Unit 3 by ".", Unit 4 by "x", Unit 5 by "5", Unit 6 by "6", Unit 7 by "7", Unit 8 by "8", and Unit 9 by blanks.

EXAMPLE 7. LARGE STORAGE ARRAY

The storage array described in Sect. F11.5.6.3 and Fig. F11.5.15 is such a sparse array that the mixture map had to be very large in order to show the detail of the shelves and uranium buttons. The mixture maps for this configuration were not presented in Sect. F11.5.6.3 but the data description was listed so the user could generate them. It may be useful to generate a unit map for this kind of problem. The data description for generating unit maps for this storage array is given below. The unit maps have been intentionally distorted by restricting the x direction which extends from 624.84 cm to -30.48 cm to 60 lines. The z direction which extends from -10.0 cm to 196.0 cm is printed in 60 characters. Thus the x direction is considerably compressed.

The plot data and unit map for an x-z slice through the array at y=30.48 cm is given in Figs. F11.5.56 and F11.5.57. This unit map was created with z across the page and x down the page. Thus, the unit map shows five rows of shelves in the x direction represented by 1's for Unit 1 with the aisles represented by 6's for Unit 6. The borders of Unit 7's are part of the concrete floor, walls, and ceilings.

```
X-Z SLICE AT Y=30.48 WITH Z ACROSS AND X DOWN

                                                    UNIT MAP

   UNIT     1   2   3   4   5   6   7
 SYMBOL     1   2   3   4   5   6   7

        UPPER  LEFT            LOWER RIGHT
        COORDINATES            COORDINATES

X        0.6095E+03             -0.1500E+02
Y        0.3048E+02              0.3048E+02
Z       -0.1500E+02              0.2010E+03

             U AXIS          V AXIS
             (DOWN)          (ACROSS)

X          -1.00000          0.0
Y           0.0              0.0
Z           0.0              1.00000

NU=   60   NV=   60     DELU= 1.0000E+01     DELV= 3.4000E+00
```

Fig. F11.5.56. Plot data for x-z slice of storage array

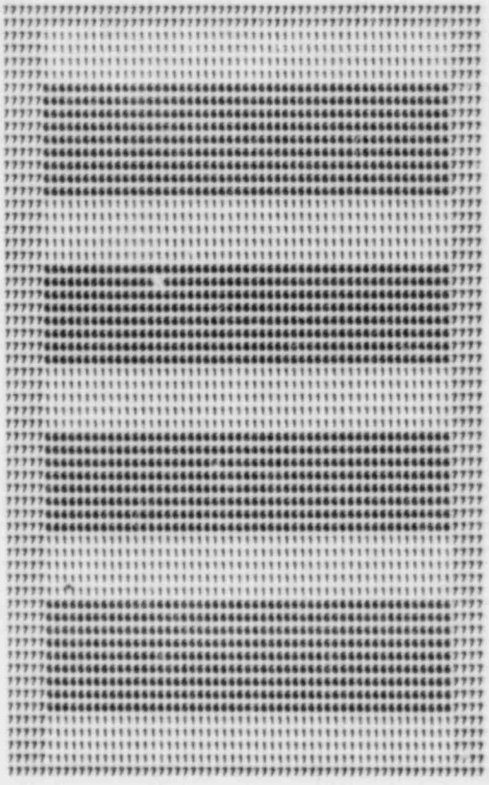


Fig. F11.5.57. x-z plot of storage array

The plot data and unit map for an x-y slice through the shelf are given in Figs. F11.5.58 and F11.5.59. This unit map was created with x across the page and y down the page. This shows 5 rows of shelves in the x direction represented by 1's for Unit 1 and separated by aisles of 6's for Unit 6. There are two rows of shelves in the y direction represented by 1's for Unit 1 and separated by aisles of 4's for Unit 4. The border of 7's are part of the concrete walls. The x direction (down the page) is considerably compressed.

```
X-Y SLICE OF ROCK THROUGH SHELF Z=0.3175 WITH X ACROSS AND Y DOWN

                                                    UNIT MAP

  UNIT     1   2   3   4   5   6   7
SYMBOL     1   2   3   4   5   6   7

        UPPER  LEFT              LOWER RIGHT
        COORDINATES              COORDINATES

X      -0.1500E+02               0.6110E+01
Y       0.1381E+00              -0.3048E+02
Z       0.3175E+00               0.3175E+00

             U AXIS          V AXIS
             (DOWN)          (ACROSS)

U            0.0              1.00000
V           -1.00000          0.0
W            0.0              0.0

NU=   60   NV=   60     DELU= 2.2860E+01     DELV= 1.0433E+01
```

Fig. F11.5.58. Plot data for x-y slice of storage array

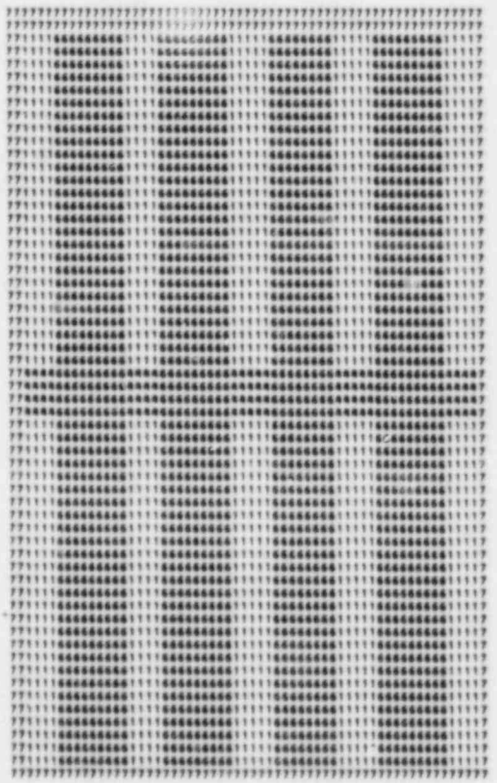


Fig. F11.5.59. x-y plot of storage array

F11.5.10 RESTART CAPABILITIES

Restart data can be written and used for restarting a problem. This type of data is saved by specifying a file definition card in the job control language on the unit associated with parameter WRS= when it is written, and RST= when it is read. Most input data can be changed *only* if the problem is restarted with the first generation. However, certain parameter data can be changed if the problem is restarted at a generation greater than one.

Parameters that can be changed when a problem is restarted at a generation greater than one include: RND=, TME=, TBA=, GEN=, RES=, LNG=, BEG=, AMX=, XAP=, XS1=, XS2=, PKI=, PID=, CKU=, CKP=, CKH=, CKA=, FMU=, FMP=, FMH=, FMA=, BUG=, TRK=, PWT=, PGM=, RUN=, PLT=, NB8=, NL8=, and PAX=. All the logical unit numbers can be changed.

If NSK= is changed, it will cause the fluxes, fission densities, leakage, absorptions, and fissions to be incorrect.

FLX=, FDN=, FAR=, MKU=, MKP=, MKH= and MKA= can be changed subject to the following restrictions:

1. If the original problem specified YES, and the restarted problem specifies NO, the data will be calculated but not printed. A warning message will be printed.
2. If the original problem specified NO and the restarted problem specifies YES, an error message is printed and the problem is terminated.

The parameters RUN= and PLT= differ from other parameters because they can be set in the parameter data and overridden in the plot data. Their values are stored in a common block that is written out if a problem is to be restarted. Therefore, when a problem is run, the value entered in the plot data will override the value that was entered in the parameter data. However, if the problem is restarted, the value from the restart file will be used unless it is overridden by entering additional data.

If WRS= is defined in the parameter data and RES= is not entered, a full restart data file is not written. The input data are written on the restart data file but the calculated data are not. When the restart data file is written in this manner, it can only be used to restart a problem at the first generation. Any desired data can be overridden when a problem is restarted at the first generation. This is accomplished by reading in the desired data.

To write a complete restart data file, both WRS= and RES= must be specified in the parameter data. However, WRS= is defaulted to 35 if a value greater then zero is entered for RES=. In this case, a file definition card for Unit 35 must be supplied by the user in order to save the restart data.

A problem that is restarted can also write a restart data file. These data are written on the same data file if no entry for WRS= is made in the parameter data. It can be written on a different unit if WRS= is so specified in the parameter data and the proper file definition card is included in the job control language. If a restarted problem does not have RES=0 specified in the parameter data, it will continue writing calculated restart data. Therefore, the user can run a long problem a little bit at a time by allowing the restarted problem to write restart data and then restarting the problem with those data. This can be done in sequence until the desired number of generations have been completed.

For example, consider a problem that is to run 500,000 histories (500 generations of 1000 histories per generation) and the amount of computer time available at any given time is quite limited. On the first pass, GEN=500 NPG=1000 RES=500 WRS=35 should be included in the parameter data and a unit specification for Unit 35 should be included in the job control language when the problem is run. Note that 35 is an arbitrary number chosen by the user. KENO V will automatically pull the job before it runs out of time or I/O's. A restart data file will be written on Unit 35 for the last generation that was completed. If the user wishes to restart the problem by reading the data from Unit 35 and writing a new restart file on the same unit, the parameter data for the second and all subsequent passes would be: READ PARAM BEG=500 RST=35 END PARAM.

The random sequence can be changed when a problem is restarted by entering a different random number in the parameter data. The starting random number is acceptable for changing the random sequence of a problem that is to be restarted at a generation greater than one.

Parameter data can be changed by entering new values for the desired options. This has the effect of inputting additional data within the parameter data block as explained in Sect. F11.5.3. However, all other data blocks that are to be changed must be entered as an entire data block and will completely replace that data block from the restart problem. When a problem is restarted and some of the data are overridden, the original restart data blocks are retained on the unit defined by WRS=, if it is the same as the unit defined by RST=. If the unit number associated with WRS= is different from the unit number associated with RST=, the resultant data blocks are written on the unit number associated with WRS=. For example, if the original problem is a single unit problem with RST=95 and WRS=95, and an array data block is entered to change it to an array problem, the original single unit problem data blocks remain on Unit 95. However, if the single unit problem is restarted with RST=95 and WRS=96, and an array data block is entered to change the problem to an array problem, the original single unit data blocks remain on Unit 95 and the resultant array problem data blocks are retained on Unit 96.

When a problem is restarted, the only way to change data in a data block is to reenter a new data block. For example, if in the original problem an array data block was entered as: READ ARRAY NUX=11 NUY=5 NUZ=3 END ARRAY and the problem is to be restarted at the first generation as an 11x5x6 array, the following data block must be entered: READ ARRAY NUX=11 NUY=5 NUZ=6 END ARRAY.

Section F11.2.6 contains some information pertaining to the restart capability. Other information may be found in Sect. F11.4.3. The structure of the restart file is listed in Table F11.5.2, and the variables referenced in that table are listed in Table F11.5.3.

F11.5.11 RANDOM SEQUENCE

The random number package utilized by KENO V always starts with the same seed and thus always reproduces the same sequence of random numbers. The current random number is printed at various places in the KENO V printout, and any of them except the one printed in the parameter table can be used to activate a different random sequence. The user can rerun a problem with a different random sequence by simply entering a hexidecimal random number, other than the starting random number, in the parameter data. The last digit of the random number should always be an odd number. For example, by entering RND=A10C1893E6D5 in the parameter data, the problem will be run with a different random sequence.

Table F11.5.2. Structure of RESTART file

Record number	Contents
(1)	TITLE(20), NBA, NPB, NSKIP, NRSTRT, NBANK, NFBNK, NXBNK, NXFBK, NUMX1D, TMAX, TBTCH, RNDNUM, LOG*1(36) (This record contains the title and parameter data.)
(2)	MT(LMT) LMT = 5 + NUMX1D (This record contains the identifiers of the 1-D cross-section arrays.)
(3 to NUMPT+2)	NDX = 1, NUMPT NUMPT = 11 (a) NDX,NREC (This record contains the index for each type of data and the number of records of data associated with each type of data.) (b) NREC - 1 records of data (These records contain the data associated with the specified type of data.) NDX = 1 (geometry data) (b1) LLNGTH, KMAX, KREFM, NGBLU, NBOXT,MAXMIX, MAXIMP, NUMHOL, NUCOM, EXRFL, LFIL, LFIL, LFIL (b2) MAT(KREFM), IMP(KREFM), FGEOM(KREFM), IGEOM(KREFM), XX(7,KREFM), KBNDS1(NBOXT), KBNDS2(NBOXT), IFH(KREFM), ILH(KREFM), KHOLE(NUMHOL), LHOLU(NUMHOL), HOLX(NUMHOL), HOLY(NUMHOL), HOLZ(NUMHOL), ICOMC(NBOXT), UCOMNT(33,NUCOM) NDX = 2 (array or unit orientation data) (b1) LLNGTH, NGLOBL, MAXARA, NACOM, LSGUN, MBOX, LFIL, LFIL (b2) NBXMAX(MAXARA), NBYMAX(MAXARA), NBZMAX(MAXARA), ITYPE(MAXARA), LPT(MAXARA), LNG(MAXARA), ICOMA(MAXARA), ACOMNT(33,NACOM) I = 1 MAXARA (The following records are written for each array if multiple units are entered in the geometry data.) (b3) IF (LPT(I).GT.0) LBA(NBXMAX(I),NBYMAX(I),NBZMAX(I)) NDX = 3 (mixing table data) (b1) LLNGTH, NMIX, NSCT, MIX, MIXT, NPL (b2) MIXTUR(NMIX), NUC(NMIX), DEN(NMIX) NDX = 4 (extra data)

Table F11.5.2 (continued)

Record number	Contents
	(b1) LLNGTH
	(b2) D(LLNGTH)
	NDX = 5 (weighting function by energy group and importance reg.)
	(b1) LLNGTH, NIMP
	(b2) WTAVG(NGP,NIMP)
	NDX = 6 (start data)
	(b1) LLNGTH, NTYPST, TFX, TFY,TFZ, NBXS, NBYS, NBZS KFIS, LFIN, NBOXST, FRACT, FISVOL, XSM, XSP, YSM, YSP, ZSM, ZSP, RFLKEY, LPRT6, LPSTP, LFIL
	(b2) IF (NTYPST .EQ. 6) X(NPB), Y(NPB), Z(NPB), NBX(NPB), NBY(NPB), NBZ(NPB)
	NDX = 7 (albedo data)
	(b1) LLNGTH, NALB, NANG, NG, INTR(6) ,RNAMES(2,6), IDALB(6), NBXL(6), LNXX
	IF (NALB .GT. 0)
	(b2) NABS(3,NALB), LABS(3,NALB)
	(b3) MAL(3,NANG,NG)
	(b4) EALB(NG+1) (group boundaries)
	I = 1,NALB
	(b5(i)) PLIM(NANG), CPOL(NANG), SPOL(NANG)
	(b6(i)) ALB(LENG) (LENG is the length of the albedos for a given angle.)
	(b7(i)) A(LENG,NANG)
	NDX = 8 (mixed cross sections)
	(b1) LLNGTH, MATT, NGP, NSCT
	(b2) LXS(MANG,MATT) MANG = 2*NSCT + 4

Table F11.5.2 (continued)

Record number	Contents
	I = 1, MATT for each existing mixture
	(b3(i)) ID(50) the mixture information record
	(b4(i)) X1D(NN1D,NGP+1) NN1D = ID(28) no. of 1-D's
	(b5(i)) MWA(3,NGP)
	(b6(i)) P0(LNG) LNG = MWA(2,NGP)
	J = 1, NSCT
	(b7(i,j)) ANG(LNG)
	(b8(i,j)) PRB(LNG)
	NDX = 9 (energies and inverse velocities)
	(b1) LLNGTH
	(b2) E(NGP+1), VINV(NGP)
	NDX = 10 (plot data)
	(b1) LLNGTH, NUMPLT
	I = 1, NUMPLT
	(b2(i)) XL, YL, ZL, XR, YR, ZR, VX, VY, VZ, UX, UY, UZ, DELV, DELU, NV, NU, LPIC, PTITL(33), TABLE(59)
	NDX = 11 (biasing data)
	(b1) LLNGTH, NUMIDS, NCS, NTSETS
	IF ((NUMIDS + NCS).GT.0),
	(b2) ID(NUMIDS), IBGN(NUMIDS), IEND(NUMIDS), WTTITL(3,NUMIDS), NCID(NCS), NCSETS(NCS), CRDTTL(3,NCS), NCTHK(NTSETS), NUMINC(NTSETS), NGPWTS(NTSETS), IPTWT(NTSETS)
	I =1, NCS
	J =1, NSETS(I)
	(b3(i,j)) WTAVG(NGPWTS(j,i),NUMINC(j,i))
(14 to END)	calculated data as listed below

Table F11.5.2 (continued)

Record number	Contents
	(a) IGEN, RND, NPB, NGP, KMAX, LBANK, NBANK, LIF(50), LOJIC*1(36), EFFK(IGEN) (This record contains parameter data, COMMON /LIFETM/ part of COMMON /LOGIC/ and k-effectives by generation.)
	(b) NUBANK(LBANK,NBANK)
	(c) IF(LOJIC(4)) FISDEN(KMAX) (If fission densities are calculated, the fission densities are written on the restart file.)
	(d) IF(LOJIC(10)) TP(MATDIM,MATDIM,3), SNP(MATDIM), SP(MATDIM) (If matrix data by position are calculated, the matrix by position data are written on the restart file.)
	(e) IF(LOJIC(7)) TU(NBOXT, NBOXT,3), SNU(NBOXT), SU(NBOXT) (If matrix data by unit are calculated, the matrix by unit data are written on the restart file.)
	(f) IF(LOJIC(13)) TH(NUMHOL,NUMHOL,3), SNH(NUMHOL), SH(NUMHOL) (If matrix data by hole are calculated, the matrix by hole data are written on the restart file.)
	(g) IF(LOJIC(17)) TA(NUMARA,NUMARA,3), SNA(NUMARA), SA(NUMARA) (If matrix data by array are calculated, the matrix by array data are written on the restart file.)
	(h) 1 to NGP (the following records are written for each energy group.
	(h1) IGROUP, FLEAK(3)
	(h2) FMABS(LREG,3), FMFIS(LREG,3)
	(h3) IF(LOJIC(3)) FLUX(KMAX,3)
	Repeat (a through h) until LOJIC(34) is TRUE, i.e., until the last generation is completed.

Table F11.5.3. Key of RESTART file variables

TITLE	80 Character KENO problem title
NBA	Number of generations in the KENO problem
NPB	Number of histories per generation
NSKIP	Number of generations to be skipped in averaging k-effective
NRSTRT	Number of generations between writing restart data
NBANK	Number of positions in the neutron bank
NFBNK	Number of positions in the fission bank
NXBNK	Number of extra entries in the neutron bank
NXFBK	Number of extra entries in the fission bank
NUMX1D	Number of extra 1-D cross sections
TMAX	Time allowed to execute the problem
TBTCH	Time allowed for each generation
RNDNUM	Random number with which the problem will be started
LOG	COMMON /LOGIC/
LMT	Number of 1-D cross-section identifiers. LMT = 5 + NUMX1D
MT	Array containing the 1-D cross-section identifiers
NUMPT	Number of kinds of data that are written on the restart file
NDX	Index for the type of data
NREC	Number of records for the specified type of data
NDX = 1	geometry data
LLNGTH	The length of record b2
KMAX	Number of geometry regions used
KREFM	Number of geometry cards read in the geometry data
NBOX	Largest unit or boxtype number in the geometry data
NBOXT	NBOX + the number extra units generated by KENO
MAXMIX	Largest mixture number encountered in the geometry data
MAXIMP	Largest biasing number encountered in the geometry data
NUMHOL	Number of holes in the geometry data
EXRFL	Logical flag. Value is TRUE if a reflector is present
MBOX	Logical flag. Value is TRUE if multiple units are entered
LFIL	Logical*1 blank variable used to pad to full word boundary
MAT	Array containing the mixtures used in the geometry
IMP	Array containing the bias ID's used in the geometry
FGEOM	Array containing the geometry words used in the geometry
IGEOM	Array containing the shape identifiers used in the geometry
XX	Array of geometry dimensions
KBNDS1	Array of the first geometry region number in each unit
KBNDS2	Array of the last geometry region number in each unit
IFH	First hole number encountered in each geometry region
ILH	Last hole number encountered in each geometry region
KHOLE	Region number that contains the hole
LHOLU	Unit that is placed in the hole
HOLX	X coordinate of the origin of the unit in the hole with respect to the unit that contains the hole
HOLY	Y coordinate of the origin of the unit in the hole with respect to the unit that contains the hole
HOLZ	Z coordinate of the origin of the unit in the hole with respect to the unit that contains the hole
NUCOM	Number of geometry comments
UCOMNT	Comments

Table F11.5.3 (continued)

NDX = 2	array or unit orientation data
LLNGTH	length of the b2 record
NGLOBL	Global array number
MAXARA	Largest array number encountered in the array data
LSGUN	Logical flag. Value is TRUE if there is no array present
LFIL	Logical*1 blank variable used to pad to full word boundary
NBXMAX	Number of units in the X direction of each array
NBYMAX	Number of units in the Y direction of each array
NBZMAX	Number of units in the Z direction of each array
ITYPE	Pitch indicator, 1 for square pitched arrays
LPT	Pointer to locate the beginning of each unit orientation array
LNG	Length of each unit orientation array
LBA	Contains the unit orientation arrays for all arrays
NACOM	Number of array comments
ACOMNT	Comments
NDX = 3	mixing table data
LLNGTH	Length of b2 record
NMIX	Number of entries in the mixing table
NSCT	Number of scattering angles
MIX	Number of different mixtures to be mixed
MIXT	Largest mixture number to be mixed
NPL	Order of Legendre coefficients + 1
PBXS	Cross-section message cutoff value
MIXTUR	Array of Mixture numbers used in the mixing table
NUM	Array of the nuclide ID numbers used in the mixing table
DEN	Array of the number densities used in the mixing table
NDX = 4	extra data
LLNGTH	Length of b2 record
LENGTH	D(LENGTH) is the data contained in the b2 record
NDX = 5	weighting function
LLNGTH	Length of b2 record
NIMP	Number of biasing regions
WTAVG	Average weight by energy group and biasing region
NDX = 6	start data (initial source distribution)
LLNGTH	Length of b2 record
NTYPST	Start type to define the initial source distribution
TFX	X coordinate of neutron starting point
TFY	Y coordinate of neutron starting point
TFZ	Z coordinate of neutron starting point
NBXS	X index of unit's position in the global array
NBYS	Y index of unit's position in the global array
NBZS	Z index of unit's position in the global array
KFIS	Mixture whose fission spectrum is used for initial source
LFIN	Last neutron to be started at the specified point
NBOXST	Unit in which neutrons will be started
FRACT	Fraction of initial source to be started as a spike

Table F11.5.3 (continued)

FISVOL	Fraction of global system that contains fissile material
XSM	-X dimension of cuboid in which neutrons will be started
XSP	+X dimension of cuboid in which neutrons will be started
YSM	-Y dimension of cuboid in which neutrons will be started
YSP	+Y dimension of cuboid in which neutrons will be started
ZSM	-Z dimension of cuboid in which neutrons will be started
ZSP	+Z dimension of cuboid in which neutrons will be started
RFLKEY	Logical variable. Set TRUE if neutrons can be started in reflector
LPRT6	Logical variable. Set TRUE to print start type 6 data
LPSTP	Logical variable. Set TRUE to print initial source points
LFIL	Logical*1 blank variable used to pad to full word boundary
NPB	Number of histories per generation
X	X coordinate of neutron starting point for start type 6
Y	Y coordinate of neutron starting point for start type 6
Z	Z coordinate of neutron starting point for start type 6
NBX	X index of unit's position in the global array for start type 6
NBY	Y index of unit's position in the global array for start type 6
NBZ	Z index of unit's position in the global array for start type 6
NDX = 7	albedo data
LLNGTH	Length of b2 record
NALB	Number of different differential albedos to be used
NANG	Number of angles available in the albedo function
NG	Number of energy groups available in the albedo function
INTR	Type of boundary condition for each face
RNAMES	Name of boundary condition
IDALB	Index to the correct set of albedo data
NBXL	Specifies the face where the history will reenter
LNXX	Logical flag. Set TRUE for either specular or differential reflection
NABS	Pointers into the albedo data for each albedo used
LABS	Length of the albedo data for each albedo used
MAL	Pointer array for albedos
EALB	Energy boundaries for albedos
PLIM	Incident polar angle bins
CPOL	Cosine of returning polar angle
SPOL	Sine of returning polar angle
ALB	Probabilities for the returning energy group
ALB	Probabilities for the returning angle
LENG	Length of albedo data for a given angle
NDX = 8	mixed cross sections
LLNGTH	Length of b2 record
MATT	Number of mixtures used in the problem
NGP	Number of energy groups from the cross-section file
NSCT	Number of scattering angles
LXS	Length of each cross-section set
MANG	2*NSCT + 4
ID	Head record from mixed cross-section file
X1D	1-D cross sections

Table F11.5.3 (continued)

NN1D	Number of 1-D cross sections (ID(28))
MWA	Pointer array for cross sections
P0	Group-to-group transfer probabilities
ANG	Scattering angles
PRB	Probabilities for scattering at angle ANG
LNG	Length of the P0 arrays
NDX = 9	energies and inverse velocities
LLNGTH	Length of b2 record
NGP	Number of energy groups
E	Energy group bounds
VINV	Inverse velocities
NDX = 10	plot data
LLNGTH	Length of b2 record
NUMPLT	Number of plots to be generated
XL	X coordinate of the upper left corner of the plot
YL	Y coordinate of the upper left corner of the plot
ZL	Z coordinate of the upper left corner of the plot
XR	X coordinate of the lower right corner of the plot
YR	Y coordinate of the lower right corner of the plot
ZR	Z coordinate of the lower right corner of the plot
VX	X component of the direction cosine for the across axis of the plot
VY	Y component of the direction cosine for the across axis of the plot
VZ	Z component of the direction cosine for the across axis of the plot
UX	X component of the direction cosine for the down axis of the plot
UY	Y component of the direction cosine for the down axis of the plot
UZ	Z component of the direction cosine for the down axis of the plot
DELV	Horizontal spacing between points on the plot
DELU	Vertical spacing between points on the plot
NV	Number of characters across the plot
NU	Number of characters down the plot
LPIC	Plot type indicator. LPIC=1 mixture, LPIC=2 unit, LPIC=3 bias ID
PTITL	Plot title, 80 characters
TABLE	Characters used in plot, 59 characters
NDX = 11	biasing data
LLNGTH	Length of b2 record
NUMIDS	Number of bias ID's requested
NCS	Number of bias ID's entered from cards
NTSETS	Total number of group structures for all ID's
ID	ID number or set of weights to be read from WTS file
IBGN	Beginning importance region number assigned to ID
IEND	Ending importance region number assigned to ID
WTTITL	Title associated with the ID
NCID	ID to be read from cards

Table F11.5.3 (continued)

NCSETS	Number of group structures to be read from cards
CRDTTL	Title associated with the ID read from cards
NCTHK	Thickness per increment from cards
NUMINC	Number of increments from cards
NGPWTS	Number of energy groups for this set of weights
IPTWT	Pointer into the weights from cards
WTAVG	Weighting function by energy group and importance region
IGEN	Generation for which this set of restart data was written
RND	Last random number that was used
NPB	Number of generations
NGP	Number of histories per generation
KMAX	Number of geometry regions used
LBANK	Number of positions per neutron in the neutron bank
NBANK	Number of positions in the neutron bank
LIF	An array equivalenced to COMMON /LIFETM/
LOJIC	An array equivalenced to COMMON /LOGIC/
EFFK	Array containing k-effectives by generation
NUBANK	The neutron bank, NBANK by LBANK
FISDEN	Array containing the fission densities
TP	Fission production matrix by array position
SNP	Eigenvector of the fission production matrix by array position
SP	Source vector by array position
TU	Fission production matrix by unit
SNU	Eigenvector of the fission production matrix by unit
SU	Source vector by unit
TH	Fission production matrix by hole
SNH	Eigenvector of the fission production matrix by hole
SH	Source vector by hole
TA	Fission production matrix by array
SNA	Eigenvector of the fission production matrix by array
SA	Source vector by array
IGROUP	Current energy group
FLEAK	Leakage fraction
FMABS	Absorption fraction
FMFIS	Total fission production
FLUX	Flux
LOJIC(4)	Key indicating whether to calculate fission densities Specified by entering FDN= in the parameter data
LOJIC(10)	Key indicating whether to calculate matrix data by position Specified by entering MKP= in the parameter data
LOJIC(7)	Key indicating whether to calculate matrix data by unit Specified by entering MKU= in the parameter data
LOJIC(13)	Key indicating whether to calculate matrix data by hole Specified by entering MKH= in the parameter data
LOJIC(17)	Key indicating whether to calculate matrix data by array Specified by entering MKA= in the parameter data
LOJIC(1)	Key indicating whether to calculate fluxes Specified by entering FLX= in the parameter data

F11.5.12 MATRIX K-EFFECTIVE

Matrix k-effective calculations provide an alternate method of calculating the k-effective of the system. Cofactor k-effectives and source vectors are additional information that can be provided when the matrix k-effective is calculated. The necessary source and fission weight data are collected during the neutron tracking procedure in subroutine TRACK. This information is converted to a FISSION PRODUCTION MATRIX which is the number of next generation neutrons produced at J by a neutron born at I. The principal eigenvalue of the fission probability matrix is the matrix k-effective. KENO V offers four alternatives when calculating matrix k-effective as discussed below:

(1) If MKP=YES is specified in the parameter data, the fission production matrix is collected by array position or position index. The position index is used to reference a given location in a three-dimensional lattice. For a 2x2x2 array there are eight unique position indices as shown below.

POSITION INDEX	POSITION X	Y	Z
1	1	1	1
2	2	1	1
3	1	2	1
4	2	2	1
5	1	1	2
6	2	1	2
7	1	2	2
8	2	2	2

The fission production matrix is the number of next generation neutrons produced at index J by a neutron born at index I. This matrix is used to calculate the matrix k-effective, cofactor k-effectives and the source vector by position index. Because the size of the fission probability matrix is the square of the array size (for a 4x4x4 array there are 4096 entries), it can use vast amounts of computer memory.

(2) If MKU=YES is specified in the parameter data, the fission production matrix is collected by unit or box type. It is the number of next generation neutrons produced in unit or box type J by a neutron born in unit or box type I. This matrix is used to calculate the matrix k-effective, cofactor k-effectives and source vector by unit or box type.

(3) If MKH=YES is specified in the parameter data, the fission production matrix is collected by hole number. Matrix information can be collected at either the highest hole nesting level (first level of nesting) or the deepest hole nesting level. HHL=YES specifies that the matrix information will be collected at the first nesting level. By default, the matrix information is collected at the deepest nesting level. The fission production matrix is the number of next generation neutrons produced in hole J by a neutron born in hole I. This matrix is used to calculate the matrix k-effective, cofactor k-effectives and the source vector by hole.

(4) If MKA=YES is specified in the parameter data, the fission production matrix is collected by array number. It can be collected at the highest array level (first level of nesting) or at the deepest array level. HAL=YES specifies that the matrix information will be collected at the first nesting level. By default, the matrix information is collected at the deepest nesting level. The fission production matrix is the number of next generation neutrons produced in array J by a neutron born in array I. This matrix is used to calculate the matrix k-effective, cofactor k-effectives and the source vector by array.

The user can simultaneously utilize all methods of calculating the matrix k-effective. The results are labeled in the printout. Matrix k-effectives cannot be calculated for a single unit problem. If the user wishes to do so, the geometry description must have a cube or cuboid as its outer region and the problem description should include READ ARRAY END ARRAY. These two actions convert the single unit problem into a 1x1x1 array.

A cofactor k-effective is the eigenvalue of the fission production matrix, reduced by the row and column that references the specified unit or position index. The difference between the k-effective for the system and the cofactor k-effective for a unit or position index is an indication of the in situ k-effective of that unit or the contribution that unit makes to the k-effective of the system. The cofactor k-effective of a unit devoid of fissile material should approximate the k-effective of the system.

F11.5.13 DEVIATIONS

When a deviation is calculated by KENO V, it is the standard deviation of the mean. This assumes a large sample having a normal distribution. If results do not fall in this category, the deviations cannot be assumed to be correct. If results are correlated, the calculated deviations will be in error.

F11.5.14 GENERATION TIME AND LIFETIME

The generation time and lifetime calculations utilize the average velocity. The validity of these calculations is determined by how accurately the average velocity represents the spectrum over the range of the energy group.

F11.5.15 REFERENCE

1. S. K. Fraley, *Users Guide for ICE-II*, ORNL/CSD/TM-9/R1 (July 1977).

F11.6 DESCRIPTION OF OUTPUT

This section contains a brief description and explanation of the KENO V output. Portions of the printout will not be printed for every problem. Some printout is optional and is so noted in this section.

F11.6.1 HEADER PAGE

The first page of print from KENO V is the header page. A sample header page is shown in Fig. F11.6.1.

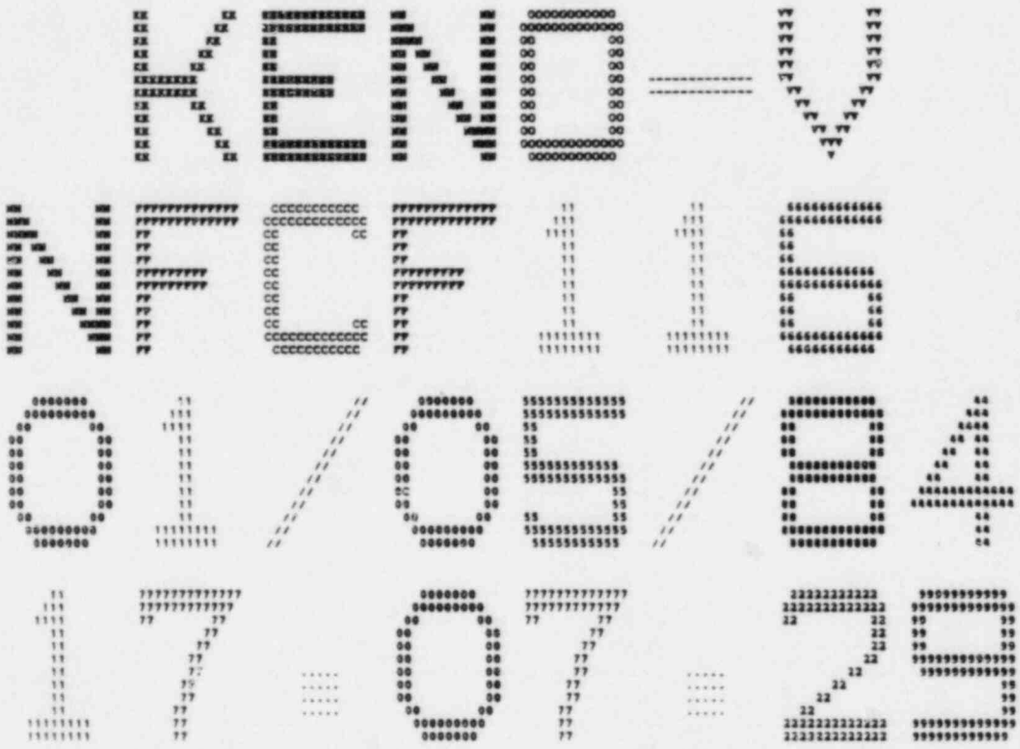


Fig. F11.6.1. Sample KENO V header page

The first line of block letters prints KENO V. The second line of block letters gives the job name from the job control language. The third line gives the date the job was run as month/day/year. The last line prints the time execution of the problem was begun (hour/minute/second) in terms of a 24-hour clock with midnight being 2400 hours. The header page is printed from subroutine MESAGE.

F11.6.2 PROGRAM VERIFICATION INFORMATION

Program verification information is printed after the header page. It lists the name of the program, the date the load module was created, and the library that contains the load module. The job name, date, and time of execution are also printed. This information may be used for quality assurance purposes.

```
PROGRAM VERIFICATION INFORMATION

        PROGRAM:  KENO5
  CREATION DATE:  84.095
        LIBRARY:  E.LMP19272.SCALE.PGMS

        JOBNAME:  NFCF116
DATE OF EXECUTION:  01/05/84
TIME OF EXECUTION:  17.07.30
```

Fig. F11.6.2. Sample program verification table

F11.6.3 TABLES OF PARAMETER DATA

The first two tables printed by KENO V list the numeric parameters and logical parameters that are used in the problem. They should always be examined by the user to verify that the parameter data block was input as desired. Examples of these tables are shown in Figs. F11.6.3 and F11.6.4.

```
RESTART OF SAMPLE PROBLEM 18

******  NUMERIC PARAMETERS  ******

TMX   MAXIMUM PROBLEM TIME (MIN)                    10.00
TBA   TIME PER GENERATION (MIN)                      0.50
GEN   NUMBER OF GENERATIONS                           103
NPG   NUMBER PER GENERATION                           350
NSK   NUMBER OF GENERATIONS TO BE SKIPPED               3
NBG   BEGINNING GENERATION NUMBER                      54
NRS   GENERATIONS BETWEEN CHECKPOINTS                 103
NUB   NUMBER OF EXTRA 1-D CROSS SECTIONS               1
MXN   NEUTRON BANK SIZE                              375
XNB   EXTRA POSITIONS IN NEUTRON BANK                  0
NFB   FISSION BANK SIZE                              350
XFB   EXTRA POSITIONS IN FISSION BANK                  0
WTA   DEFAULT VALUE OF WEIGHT AVERAGE             0.5000
WTH   WEIGHT HIGH FOR SPLITTING                   3.0000
WTL   WEIGHT LOW FOR RUSSIAN ROULETTE             0.3333
RND   STARTING RANDOM NUMBER                  00071AF9498D
NBA   NUMBER OF D.A. BLOCKS ON UNIT  8                200
NLB   LENGTH OF D.A. BLOCKS ON UNIT  8                512
ADJ   MODE OF CALCULATION                        FORWARD
      INPUT DATA WRITTEN ON RESTART UNIT             YES
      BINARY DATA INTERFACE                          YES
```

Fig. F11.6.3. Sample table of numeric parameter data

```
                                RESTART OF SAMPLE PROBLEM 10

                                *******      LOGICAL PARAMETERS      *******

 RUN  EXECUTE PROBLEM AFTER CHECKING DATA      YES        PLT  PLOT PICTURE MAP(S)                    YES
 FLX  COMPUTE FLUX                             YES        FDN  COMPUTE FISSION DENSITIES              YES
 SMU  COMPUTE AVG UNIT SELF-MULTIPLICATION      NO        MUB  COMPUTE MU-BAR & AVG FISSION GROUP      YES
 MKU  COMPUTE MATRIX K-EFF BY UNIT NUMBER      YES        MKP  COMPUTE MATRIX K-EFF BY UNIT LOCATION  YES
 CKU  COMPUTE COFACTOR K-EFF BY UNIT NUMBER    YES        CKP  COMPUTE COFACTOR K-EFF BY UNIT LOCATION YES
 FMU  PRINT FISS PROD MATRIX BY UNIT NUMBER    YES        FMP  PRINT FISS PROD MATRIX BY UNIT LOCATION YES
 MKH  COMPUTE MATRIX K-EFF BY HOLE NUMBER      YES        MKA  COMPUTE MATRIX K-EFF BY ARRAY NUMBER   YES
 CKH  COMPUTE COFACTOR K-EFF BY HOLE NUMBER    YES        CKA  COMPUTE COFACTOR K-EFF BY ARRAY NUMBER YES
 FMH  PRINT FISS PROD MATRIX BY HOLE NUMBER    YES        FMA  PRINT FISS PROD MATRIX BY ARRAY NUMBER YES
 HHL  COLLECT MATRIX BY HIGHEST HOLE LEVEL      NO        HAL  COLLECT MATRIX BY HIGHEST ARRAY LEVEL   NO
 AMX  PRINT ALL MIXED CROSS SECTIONS           YES        FAB  PRINT FIS. AND ABS. BY REGION          YES
 XS1  PRINT 1-D MIXTURE X-SECTIONS             YES        FAZ  PRINT ISEC-ALBEDO CORRELATION TABLES   YES
 XS2  PRINT 2-D MIXTURE X-SECTIONS             YES        FWT  PRINT WEIGHT AVERAGE ARRAY             YES
 ZAP  PRINT MIXTURE ANGLES & PROBABILITIES     YES        PGM  PRINT INPUT GEOMETRY                   YES
 PKI  PRINT FISSION SPECTRUM                   YES        DUG  PRINT DEBUG INFORMATION                 NO
 PXS  PRINT EXTRA 1-D CROSS SECTIONS           YES        TRK  PRINT TRACKING INFORMATION              NO

                    A RESTART PROBLEM IS TO BE RUN.  THE TITLE OF THE PARENT CASE IS
                    SAMPLE PROBLEM 10   1F27 DEMONSTRATION OF OPTIONS PROBLEM

                                PARAMETER INPUT COMPLETED

                ..... 205 IO'S WERE USED READING THE PARAMETER DATA .....

                ********** DATA READING COMPLETED **********
```

Fig. F11.6.4. Sample table of logical parameter data

```
                SAMPLE PROBLEM 10   1F27 DEMONSTRATION OF OPTIONS PROBLEM
                                GEOMETRY DESCRIPTION INPUT

UNIT     1
REGION
  1 CYLINDER     1  1  RADIUS =  9.5200     +Z =  8.7804     -Z = -8.7804
  2 CYLINDER     0  1  RADIUS =  9.5200     +Z =  8.9696     -Z = -8.7804
  3 CYLINDER     2  1  RADIUS =  10.160     +Z =  9.6296     -Z = -9.4204
  4 CUBOID       4  1     +X =  18.450     -X = -18.450     +Y =  18.450     -Y = -18.450     +Z =  17.895     -Z = -17.685

UNIT     2
REGION
  1 ARRAY        1  1     -X =  .0         -Y =  .0         -Z =  .0

UNIT     3
REGION
  1 ARRAY        2  1     -X =  .0         -Y =  .0         -Z =  .0

UNIT     4
REGION
  1 ARRAY        3  1     -X =  .0         -Y =  .0         -Z =  .0

UNIT     5
REGION
  1 ARRAY        4  1     -X =  .0         -Y =  .0         -Z =  .0

UNIT     6
REGION
  1 CUBOID       4  1     +X =  55.350     -X = -55.350     +Y =  55.350     -Y = -55.350     +Z =  53.370     -Z = -53.370
  6 REPLICATE    3  3  5 REGIONS, THICKNESS/FACE=  3.0000     3.0000     3.0000     3.0000     3.0000
  7 REPLICATE    3  7  1 REGIONS, THICKNESS/FACE=  .24000     .24000     .24000     .24000     .24000

                ********** DATA READING COMPLETED **********
```

Fig. F11.6.5. Example of unprocessed geometry input data

The title of the problem is printed at the top of each table. The first table, shown in Fig. F11.6.3, lists the numeric parameter data. It contains a triple column that lists the applicable parameter keyword used to input the data, a brief explanation of its meaning, and the associated data value. The last two entries in this table are slightly different than the others because a keyword is not listed with them. The INPUT DATA WRITTEN ON RESTART UNIT is set YES if a unit number is provided for writing restart data (WRS=). The BINARY DATA INTERFACE is set YES if a unit number is provided for reading restart data (RST=).

The second table, shown in Fig. F11.6.4, lists the logical parameter data. It contains two triple columns, each listing the applicable parameter keyword, a brief explanation of the parameter's function, and the value associated with the parameter. Messages concerning the parameter data may be printed at the bottom of the table. If the problem is one that is being restarted, the title of the parent case is printed at the bottom of the table. If the restart title or messages are not printed, the bottom section of the table is omitted. If the user desires to change some of the data in these tables, the appropriate parameter keyword must be entered in the parameter data followed by an equal sign and the desired value. Following this table is a statement affirming completion of the parameter input and a statement of the number of I/O's used in reading the parameter data. At this point the unprocessed input geometry may be printed as described in Sect. F11.6.4. These data are followed by a statement affirming the completion of the data reading.

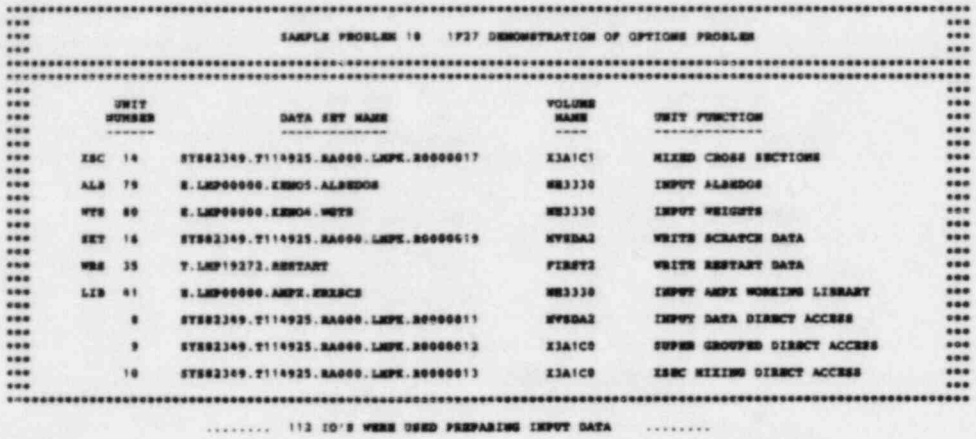
F11.6.4 UNPROCESSED GEOMETRY INPUT DATA

This printout is optional and is usually used to locate code difficulties, to show all the geometry input data when only part of it is used in the problem, or to show the order in which units were entered. It is considered debug information and is printed only if PGM=YES is specified in the parameter input data as described in Sect. F11.4.3. Standard KENO V use does not require printing this data because the processed geometry that is used in the problem is always printed. See Sects. F11.6.15 and F11.6.16 for examples of the standard printed KENO V geometry data. An example of the data printed in the unprocessed geometry input data is shown in Fig. F11.6.5. The unprocessed geometry is printed by subroutine KENOG.

When the unprocessed geometry input is printed, the problem title is located at the top of the page, followed by the heading "GEOMETRY DESCRIPTION INPUT." The region-dependent geometry information is then printed. If the problem contains a unit orientation array, the problem title is printed again, followed by the unit orientation. This is followed by a statement affirming the completion of the data input.

F11.6.5 TABLE OF DATA SETS USED IN THE PROBLEM

This is the third table of data printed by KENO V. It should be carefully scrutinized to verify the desired data set name is associated with the proper unit number and volume. An example of this table is shown in Fig. F11.6.6.



```
SAMPLE PROBLEM 10   1F27 DEMONSTRATION OF OPTIONS PROBLEM

 UNIT NUMBER        DATA SET NAME                                VOLUME NAME   UNIT FUNCTION
 -----------        -------------                                -----------   -------------
 XSC   14    ST882349.T114925.RA000.LMPX.R0000017                X3A1C1        MIXED CROSS SECTIONS
 ALB   75    H.LMP00000.KNO5.ALBEDOS                             HE3330        INPUT ALBEDOS
 WTS   80    H.LMP00000.KNO4.WGTS                                HE3330        INPUT WEIGHTS
 SET   16    ST882349.T114925.RA000.LMPX.R0000019                HVSDA2        WRITE SCRATCH DATA
 WRS   35    T.LMP19272.RESTART                                  FIRST2        WRITE RESTART DATA
 LIB   61    H.LMP00000.AMPX.XRXSC3                              HE3330        INPUT AMPX WORKING LIBRARY
       8     ST882349.T114925.RA000.LMPX.R0000011                HVSDA2        INPUT DATA DIRECT ACCESS
       9     ST882349.T114925.RA000.LMPX.R0000012                X3A1C0        SUPER GROUPED DIRECT ACCESS
      10     ST882349.T114925.RA000.LMPX.R0000013                X3A1C0        XSEC MIXING DIRECT ACCESS

 ..... 112 IO'S WERE USED PREPARING INPUT DATA .....
```

Fig. F11.6.6. Sample table of data sets used in the problem

This table, printed from subroutine DATAIN, lists unit numbers that are specified in the parameter data or are defaulted in the code, and information pertinent to them. This information is given in the following order, left to right: (1) the keyword used in the parameter data to define the unit number, (2) the unit number, (3) the data set name, (4) the name of the volume on which the data set resides, and (5) the type of data contained on the data set. This table can be useful for quality assurance purposes. Information for units whose default values have not been overridden are printed even though they may not be used in the problem. Information for every unit specified in the parameter data is also printed. Units 8, 9, and 10 are the direct access devices and their unit numbers are fixed within the code. This table is followed by a statement of the number of I/O's used preparing the input data and writing it on the direct access data sets.

F11.6.6 MIXING TABLE DATA

These data are printed by subroutine PRTMIX only if a mixing table data block is entered in the problem. It is not considered optional because it cannot be suppressed. A sample of the mixing table data is shown in Fig. F11.6.7.

The data printed in this table include the problem title and the number of scattering angles. This is followed by the entry number, the mixture number, the nuclide ID number and the number density for each entry, printed in columnar form. After all the entries have been printed, the number of I/O's used in mixing the cross sections is printed. If extra 1-D cross sections were specified in the problem (see X1D=, Sect. F11.4.3), the extra 1-D cross-section ID's will be printed. If the number of blocks on the direct access data set are insufficient to hold the cross-section data, a message is printed stating: THE NUMBER OF DIRECT ACCESS BLOCKS ON UNIT _____ HAS BEEN INCREASED TO _____ If the problem is to write a restart data set (RES=, Sect. F11.4.3) a message is printed stating that restart information was written and the restart I/O unit number is specified. This is followed by a statement of the number of I/O's used in preparing the cross sections.

```
 SAMPLE PROBLEM 18   1F17 DEMONSTRATION OF OPTIONS PROBLEM

                MIXING TABLE
     NUMBER OF SCATTERING ANGLES =  1
 CROSS SECTION MESSAGE THRESHOLD =1.0E-05
 ENTRY   MIXTURE    NUCLIDE       DENSITY
   1        1         92508     9.843350E-04
   2        1         92852     7.76600E-05
   3        1          7100     1.97460E-03
   4        1          8100     3.70860E-02
   5        1          1102     5.80760E-02
   6        2           602     1.00000E+00
   7        3          1102     8.25710E-03
   8        3          6100     3.96970E-02
   9        4           502     1.00000E-09


CROSS SECTIONS READ FROM THE ANSP WORKING LIBRARY ON UNIT=   41
  502    D2/H WATER  0.99820G/CC  H=.066742, O=.033371
  602    D2/H PLEXIGLAS  1.1820/CC  H=.056884, C=.035552, O=.01813
 1102    HYDROGEN D2/H                          HANSEN ROACH
 6100    CARBON                                 HANSEN ROACH
 7100    NITROGEN                               HANSEN ROACH
 8100    OXYGEN                                 HANSEN ROACH
92508    U-235-SR     SIG P = 1000              HANSEN ROACH
92852    U-238 SIG P = 28000                    HANSEN ROACH

........  243 IO'S WERE USED MIXING CROSS-SECTIONS        ........

 1-D CROSS SECTION ARRAY ID NUMBERS
    1  2002  1452    27    18  1018
*****  RESTART DATA HAS BEEN WRITTEN ON UNIT 35  *****

........  91 IO'S WERE USED PREPARING THE CROSS SECTIONS     ........
```

Fig. F11.6.7. Example of mixing table data

The user should examine the mixing table carefully to verify that the proper nuclides are specified for the proper mixtures and that all the number densities are correct. The mixing table is printed in subroutine PRTMIX. Following this table is a statement of the unit number from which the cross-section data were read, and a list of the cross section ID's and their titles. This information is printed by subroutine MIXMIX.

F11.6.7 ALBEDO-CROSS SECTION CORRESPONDENCE

Printing the albedo cross-section correspondence tables is optional. The headings for the tables are printed in subroutine CORRE, then subroutine RATIO prints the data. These tables are printed only if PAX=YES is specified in the parameter data as described in Sect. F11.4.3. Examples of these tables are shown in Figs. F11.6.8 and F11.6.9.

XSEC ENERGY GROUP	CROSS SECTION GROUP LETHARGY BOUNDARIES		CUMULATIVE PROBABILITIES FOR CHOOSING THE CORRESPONDING ALBEDO GROUP FOR EACH XSEC GROUP							
			ALBEDO GROUP	PROBABILITY	ALBEDO GROUP	PROBABILITY	ALBEDO GROUP	PROBABILITY	ALBEDO GROUP	PROBABILITY
1	0.0	0.120468E+01	1	1.00000						
2	0.120468E+01	0.196618E+01	2	1.00000						
3	0.196618E+01	0.240798E+01	3	1.00000						
4	0.240798E+01	0.321898E+01	4	1.00000						
5	0.321898E+01	0.460528E+01	5	1.00000						
6	0.460528E+01	0.637718E+01	6	1.00000						
7	0.637718E+01	0.811178E+01	7	1.00000						
8	0.811178E+01	0.980828E+01	8	1.00000						
9	0.980828E+01	0.115138E+02	9	1.00000						
10	0.115138E+02	0.127178E+02	10	1.00000						
11	0.127178E+02	0.138168E+02	11	1.00000						
12	0.138168E+02	0.150198E+02	12	1.00000						
13	0.150198E+02	0.161188E+02	13	1.00000						
14	0.161188E+02	0.170348E+02	14	1.00000						
15	0.170348E+02	0.180218E+02	15	1.00000						
16	0.180218E+02	0.190078E+02	16	1.00000						

Fig. F11.6.8. Cumulative probabilities for correlating the albedo energy group to the cross-section energy group

ALBEDO ENERGY GROUP	ALBEDO GROUP LETHARGY BOUNDARIES		CUMULATIVE PROBABILITIES FOR CHOOSING THE CORRESPONDING XSEC GROUP FOR EACH ALBEDO GROUP							
			XSEC GROUP	PROBABILITY	XSEC GROUP	PROBABILITY	XSEC GROUP	PROBABILITY	XSEC GROUP	PROBABILITY
1	-0.405478E+00	0.120480E+01	1	1.00000						
2	0.120480E+01	0.196618E+01	2	1.00000						
3	0.196618E+01	0.260798E+01	3	1.00000						
4	0.260798E+01	0.321899E+01	4	1.00000						
5	0.321899E+01	0.460528E+01	5	1.00000						
6	0.460528E+01	0.637718E+01	6	1.00000						
7	0.637718E+01	0.811178E+01	7	1.00000						
8	0.811178E+01	0.980828E+01	8	1.00000						
9	0.980828E+01	0.115138E+02	9	1.00000						
10	0.115138E+02	0.127178E+02	10	1.00000						
11	0.127178E+02	0.138168E+02	11	1.00000						
12	0.138168E+02	0.150198E+02	12	1.00000						
13	0.150198E+02	0.161188E+02	13	1.00000						
14	0.161188E+02	0.170348E+02	14	1.00000						
15	0.170348E+02	0.184218E+02	15	1.00000						
16	0.184218E+02	0.230368E+02	16	1.00000						

Fig. F11.6.9. Cumulative probabilities for correlating the cross-section energy group to the albedo energy group

The table shown in Fig. F11.6.8 contains, left to right, the cross-section energy group, the lower and upper lethargy bounds, the corresponding albedo energy groups and the cumulative probability associated with each albedo energy group for choosing the albedo energy group corresponding to the cross-section energy group. The table shown in Fig. F11.6.9 is the inverse of the table shown in Fig. F11.6.8. It provides the cumulative probabilities for choosing the cross-section energy group corresponding to the albedo energy group. The information in these tables is automatically generated by KENO V.

F11.6.8 1-D MACROSCOPIC CROSS SECTIONS

The decision to print the 1-D mixture cross sections is optional. They are printed in subroutine PRT1DS only if XS1=YES is specified in the parameter data as described in Sect. F11.4.3. When the 1-D cross sections are to be printed, they are printed a group at a time for each mixture. The 1-D mixture cross sections for a mixture are shown in Fig. F11.6.10.

SAMPLE PROBLEM 18 1F27 DEMONSTRATION OF OPTIONS PROBLEM

1 =1D MATERIAL= 1

GROUP	SGT	NAF	ABF	NFF	CHI	NWA1	NWA2	NWA3	SUM
1	1.399648E-01	9.756955E-01	2.330455E-02	2.597678E-02	2.040000E-01	1	6	1	1.000000E+00
2	1.777708E-01	9.913464E-01	8.653915E-03	1.022688E-02	5.480000E-01	7	12	2	1.000000E+00
3	3.197028E-01	9.956222E-01	4.377084E-03	9.530998E-03	7.160000E-01	13	18	3	9.999998E-01
4	3.839978E-01	9.963052E-01	3.694645E-03	7.659538E-03	8.960000E-01	19	24	4	9.999998E-01
5	5.445119E-01	9.969695E-01	3.030942E-03	6.359638E-03	9.860000E-01	25	30	5	1.000000E+00
6	8.292615E-01	9.969191E-01	3.088988E-03	7.314600E-03	1.000000E+00	31	36	6	1.000000E+00
7	9.897208E-01	9.994868E-01	5.533068E-03	1.023428E-02	1.000000E+00	37	42	7	1.000000E+00
8	1.044048E+00	9.893114E-01	1.060449E-02	1.825308E-02	1.000000E+00	43	48	8	9.999998E-01
9	1.061208E+00	9.737411E-01	2.626038E-02	4.269738E-02	1.000000E+00	49	54	9	1.000000E+00
10	1.094638E+00	9.451114E-01	5.488538E-02	9.261078E-02	1.000000E+00	55	60	10	9.999998E-01
11	1.095548E+00	9.444278E-01	5.557138E-02	7.556438E-02	1.000000E+00	61	66	11	9.999998E-01
12	1.103148E+00	9.373448E-01	6.265588E-02	6.094518E-02	1.000000E+00	67	71	12	1.000000E+00
13	1.074338E+00	9.614618E-01	3.859958E-02	6.734618E-02	1.000000E+00	72	75	13	1.000000E+00
14	1.289288E+00	9.351818E-01	6.482068E-02	1.309468E-01	1.000000E+00	76	78	14	1.000000E+00
15	1.838078E+00	8.752828E-01	1.247978E-01	2.427318E-01	1.000000E+00	79	80	15	1.000000E+00
16	3.382128E+00	8.161918E-01	1.838088E-01	3.678748E-01	1.000000E+00	81	81	16	1.000000E+00

XSEC ID GRP.	18
1	8.851908E-03
2	6.966528E-03
3	3.767488E-03
4	3.076128E-03
5	2.585218E-03
6	2.985558E-03
7	4.177218E-03
8	7.450228E-03
9	1.734588E-02
10	3.372198E-02
11	3.884268E-02
12	2.487568E-02
13	2.748798E-02
14	5.344758E-02
15	9.907308E-02
16	1.501538E-01

Fig. F11.6.10. Example of macroscopic 1D cross sections

When the 1-D mixture cross sections are printed, the problem title is printed at the top of the page. The mixture ID and MATERIAL numbers are then printed. ID is the mixture number from the mixing table and MATERIAL is the index used to reference it. This is followed by a heading to identify the different 1-D cross sections. GROUP is the energy group, SGT is the total cross section for the mixture, NAP is the nonabsorption probability, NFP is the production probability, CHI is the fission spectrum, MWA1 is the pointer for the first position of the cross sections for the energy group, MWA2 is the pointer for the last position of the cross sections for the energy group, and MWA3 contains the group for the transfer corresponding to the first position. SUM is the sum of the absorption probability and the nonabsorption probability. The absorption probability is defined as the absorption cross section divided by the total cross section. The nonabsorption probability is the sum of the group-to-group transfers for this group, divided by the total cross section. The production probability is defined as the fission production cross section divided by the total cross section. The nonabsorption probability and the production probability are not true probabilities in that they may be greater than one. This is because the nonabsorption probability has the (n,2n) transfer array summed into the total transfer array twice, and the (n,3n) is summed three times, etc. The production probability is $\nu\Sigma_f$ and can also be greater than one.

F11.6.9 EXTRA 1-D CROSS SECTIONS

Printing the extra 1-D cross sections is optional. They are printed in subroutine PRT1DS if P1D=YES is specified in the parameter data (Sect. F11.4.3). Extra 1-D cross sections are not used in KENO V unless NUB=YES is specified in the parameter data or the user has altered the code to access and utilize other 1-D cross sections. If NUB=YES is specified, the extra 1-D cross section is the fission cross section, and is used to calculate the average number of neutrons per fission. This is printed only for fissile mixtures as shown in Fig. F11.6.10. The fission cross-section heading is XSEC ID 18 and it follows the table of 1-D cross sections.

F11.6.10 2-D MACROSCOPIC CROSS SECTIONS

The decision to print the 2-D mixture cross sections is optional. They are printed in subroutine PRT2DS only if XS2=YES is specified in the parameter data. They are printed after the 1-D cross sections for the mixture. A heading is printed, followed by the transfer data. An example of the 2-D mixture cross sections is given in Fig. F11.6.11.

```
SCATTERING TRANSFER ARRAY FOR MATERIAL     1

     FROM   GRP  1     GRP  2     GRP  3     GRP  4     GRP  5     GRP  6     GRP  7     GRP  8     GRP  9     GRP 10
TO GRP
   =  0  2.542099E-01 3.105339E-01 2.901828E-01 3.710848E-01 4.553199E-01 4.726111E-01 4.568233E-01 4.472678E-01 4.489633E-01 3.305588E-01
   =  1  4.486899E-01 2.731078E-01 4.376708E-01 4.858499E-01 4.561028E-01 4.376482E-01 4.458778E-01 4.561148E-01 3.887258E-01 4.512428E-01
   =  2  1.044768E-01 2.315508E-01 1.976938E-01 1.187798E-01 7.317428E-02 7.391048E-02 7.959868E-03 6.859678E-02 1.079098E-01 1.526358E-01
   =  3  1.066808E-01 1.395568E-01 5.497748E-02 2.083788E-02 1.251168E-02 1.332618E-02 1.233228E-02 1.024178E-02 3.821788E-02 4.378638E-02
   =  4  6.965498E-02 3.865768E-02 9.487678E-03 3.339648E-03 2.353568E-03 2.121838E-03 3.599358E-03 6.859678E-03 1.079098E-02 1.302368E-02
   =  5  2.128198E-02 6.590868E-03 2.189468E-03 9.108108E-04 5.348598E-04 8.487318E-04 1.770178E-03 2.923808E-03 5.395468E-03 8.757278E-03

     FROM   GRP 11     GRP 12     GRP 13     GRP 14     GRP 15     GRP 16
TO GRP
   =  0  3.029468E-01 3.192038E-01 2.891368E-01 3.463478E-01 8.186198E-01 1.000008E+00
   =  1  4.940318E-01 4.599008E-01 4.338848E-01 6.966918E-01 1.893818E-01
   =  2  1.357338E-01 1.327758E-01 2.071458E-01 1.589628E-01
   =  3  9.035798E-02 6.638738E-03 6.983558E-03
   =  4  2.020798E-02 2.173598E-03
   =  5  6.735678E-03
```

Fig. F11.6.11. Example of 2-D macroscopic cross sections

F11.6.11 PROBABILITIES AND ANGLES

Printing the probabilities and angles is optional. They are printed if the number of scattering angles is greater than zero and XAP=YES is specified in the parameter data as described in Sect. F11.4.3. The probabilities and angles are printed for each mixture by subroutine PRT2DS. An example of the probabilities is shown in Fig. F11.6.12 and an example of the angles is shown in Fig. F11.6.13. If the group-to-group transfer for a mixture is isotropic, the first angle for that transfer will be set to -2.0 as a flag to the code.

```
PROBABILITY   1  ARRAY FOR MATERIAL     1

     FROM   GRP  1     GRP  2     GRP  3     GRP  4     GRP  5     GRP  6     GRP  7     GRP  8     GRP  9     GRP 10
TO GRP
  +  0     1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00
        TRANSFERS FROM GRP+  1 TO GRP+  4 ARE THE SAME AS ABOVE
  +  5     1.000000E+00 1.000000E+00 0.0          0.0          0.0          0.0          1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00

     FROM   GRP 11     GRP 12     GRP 13     GRP 14     GRP 15     GRP 16
TO GRP
  +  0     1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00
  +  1     1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00
  +  2     1.000000E+00 1.000000E+00 1.000000E+00 1.000000E+00
  +  3     1.000000E+00 1.000000E+00 1.000000E+00
  +  4     1.000000E+00 1.000000E+00
  +  5     1.000000E+00
```

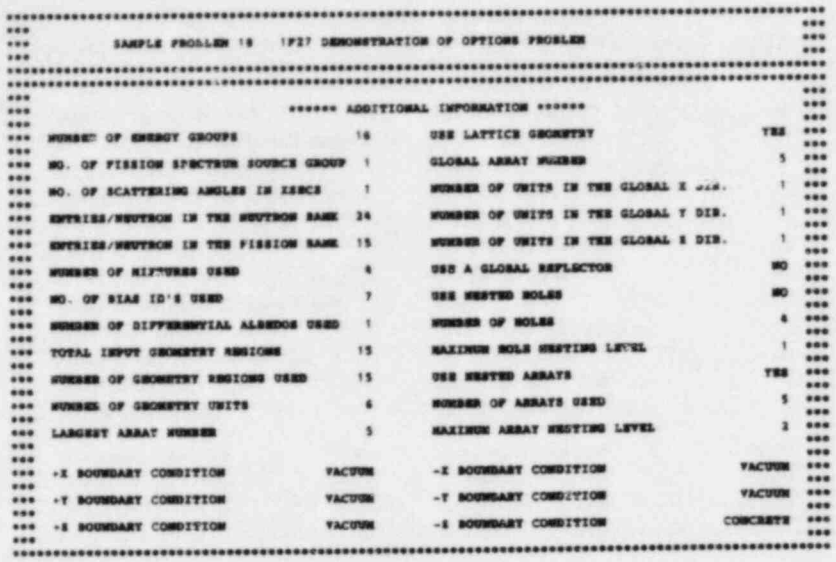
Fig. F11.6.12. Example of macroscopic probabilities

```
ANGLE   1  ARRAY FOR MATERIAL     1

     FROM   GRP  1     GRP  2     GRP  3     GRP  4     GRP  5     GRP  6     GRP  7     GRP  8     GRP  9     GRP 10
TO GRP
  +  0     2.567117E-02 1.379553E-01 1.346164E-02 7.281156E-02 2.880556E-01 6.281699E-01 6.404545E-01 6.425068E-01 6.431508E-01 3.702149E-01
  +  1     5.308288E-01 6.165968E-01 5.692998E-01 5.777113E-01 5.526058E-01 5.451508E-01 5.491968E-01 5.502618E-01 5.780918E-01 6.175428E-01
  +  2     5.205568E-01 5.650128E-01 6.679218E-01 3.169888E-01 2.266008E-01 2.373218E-01 2.409198E-01 2.540908E-01 3.317718E-01 3.630018E-01
  +  3     3.823588E-01 3.494838E-01 2.256738E-01 1.212128E-01 1.025648E-01 1.016048E-01 1.100488E-01 1.416678E-01 1.882358E-01 2.025648E-01
  +  4     2.365838E-01 1.704938E-01 9.615388E-02 6.565658E-02 6.565658E-02 3.333338E-02 6.557378E-02 8.196728E-02 1.041678E-01 1.266908E-01
  +  5     1.197358E-01 5.000008E-02-2.000008E+00-2.000008E+00-2.000008E+00-2.000008E+00 3.333338E-02 3.846158E-02 6.250008E-02 7.051288E-02

     FROM   GRP 11     GRP 12     GRP 13     GRP 14     GRP 15     GRP 16
TO GRP
  +  0     3.372148E-01 3.360738E-01 3.092008E-01 5.563268E-01 6.453588E-01 2.524158E-01
  +  1     6.186488E-01 6.199058E-01 6.404468E-01 6.231998E-01 3.812598E-01
  +  2     3.569068E-01 3.790198E-01 6.081098E-01 1.818188E-01
  +  3     2.114058E-01 2.267348E-01 1.288258E-01
  +  4     1.250008E-01 7.235148E-02
  +  5     4.166668E-02
```

Fig. F11.6.13. Example of macroscopic angles

F11.6.12 TABLE OF ADDITIONAL INFORMATION

This is the fourth table of data printed by KENO V. It contains information determined by the input data and is printed by subroutine NSUPG. An example of this table is shown in Fig. F11.6.14.



SAMPLE PROBLEM 18 !F27 DEMONSTRATION OF OPTIONS PROBLEM

****** ADDITIONAL INFORMATION ******

NUMBER OF ENERGY GROUPS	16	USE LATTICE GEOMETRY	YES
NO. OF FISSION SPECTRUM SOURCE GROUP	1	GLOBAL ARRAY NUMBER	5
NO. OF SCATTERING ANGLES IN XSECS	1	NUMBER OF UNITS IN THE GLOBAL X DIR.	1
ENTRIES/NEUTRON IN THE NEUTRON BANK	24	NUMBER OF UNITS IN THE GLOBAL Y DIR.	1
ENTRIES/NEUTRON IN THE FISSION BANK	15	NUMBER OF UNITS IN THE GLOBAL Z DIR.	1
NUMBER OF MIXTURES USED	4	USE A GLOBAL REFLECTOR	NO
NO. OF BIAS ID'S USED	7	USE NESTED HOLES	NO
NUMBER OF DIFFERENTIAL ALBEDOS USED	1	NUMBER OF HOLES	4
TOTAL INPUT GEOMETRY REGIONS	15	MAXIMUM HOLE NESTING LEVEL	1
NUMBER OF GEOMETRY REGIONS USED	15	USE NESTED ARRAYS	YES
NUMBER OF GEOMETRY UNITS	6	NUMBER OF ARRAYS USED	5
LARGEST ARRAY NUMBER	5	MAXIMUM ARRAY NESTING LEVEL	2
+X BOUNDARY CONDITION	VACUUM	-X BOUNDARY CONDITION	VACUUM
+Y BOUNDARY CONDITION	VACUUM	-Y BOUNDARY CONDITION	VACUUM
+Z BOUNDARY CONDITION	VACUUM	-Z BOUNDARY CONDITION	CONCRETE

Fig. F11.6.14. Sample table of additional information

This table should be examined by the user to verify the problem data. The NUMBER OF ENERGY GROUPS is read either from the Monte Carlo formatted library, identified by the keyword XSC and the unit function name MIXED CROSS SECTIONS from the Table of Data Sets in Sect. F11.6.5, or from the restart unit, identified by the keyword RST and the unit function name, READ RESTART DATA. The NO. OF FISSION SPECTRUM SOURCE GROUP is the number of different energy groups for which a fission spectrum is defined. In the present version, this number should always be one. The NO. OF SCATTERING ANGLES IN XSECS is the number of scattering angles to be used in processing the cross sections. The default value is one, and may be overridden by specifying the parameter SCT= in the mixing table input as described in Sect. F11.4.10. One scattering angle yields P_1 cross sections, two scattering angles yield P_3 cross sections, three scattering angles yields P_5 cross sections, etc. ENTRIES/NEUTRON IN THE NEUTRON BANK specifies the number of pieces of data that are banked for each history during tracking. ENTRIES/NEUTRON IN THE FISSION BANK is the amount of data stored for each source neutron for each generation.

The NUMBER OF MIXTURES USED is the number of different mixtures (media) used in the geometry data utilized by the problem. This may be less than the total number of different mixtures specified in the geometry data if portions of the geometry data are not used in the problem.

The NUMBER OF BIAS ID'S USED is the number of different biasing regions used in the problem. This will always be one unless a biasing data block is entered in the problem as described in Sect. F11.4.7.

The NUMBER OF DIFFERENTIAL ALBEDOS USED is the number of different differential albedo reflectors used in the problem. This will always be zero unless the boundary condition data specify the use of differential albedo reflection on one or more faces of the system as described in Sect. F11.4.6. Also check the BOUNDARY CONDITION data printed in this table. The number of different differential albedos specified on the faces should be consistent with the NUMBER OF DIFFERENTIAL ALBEDOS USED. Specular, mirror, vacuum, and periodic are not differential albedos. Be aware that several different keywords may be used to specify the same differential albedo. See Table F11.4.4 for a list of differential albedo keywords.

The TOTAL INPUT GEOMETRY REGIONS is the number of geometry regions specified in the problem input. This excludes UNIT and BOX TYPE, but includes the core boundary description. It excludes the automatic reflector description but includes the geometry regions generated by it. The NUMBER OF GEOMETRY REGIONS USED is the number of geometry regions used in the problem. It may be less than or equal to the TOTAL INPUT GEOMETRY REGIONS. The NUMBER OF GEOMETRY UNITS is the number of different units and/or box types described in the geometry data. The LARGEST ARRAY NUMBER is the largest array number specified in the array data. (See Sect. F11.4.5.)

USE LATTICE GEOMETRY is determined by the logical flag that indicates whether or not the problem is a single unit problem. This should be YES for any problem that is not a single unit problem and NO for a single unit problem. By definition, a single unit problem is a problem that does not utilize array data in any form. Section F11.4.5 describes array data. The GLOBAL ARRAY NUMBER is the number of the array designated as the global, overall or universal array. The global array can be thought of as the array that defines the overall system.

The NUMBER OF UNITS IN THE GLOBAL X/Y/Z DIR. defines the size of the global array in terms of the number of units and/or box types that are located along the edge of the array boundaries in the x/y/z directions. For a single unit, all three of these should be zero. For a simple 1x1x1 array consisting of one unit type, all three of these numbers should be one.

USE GLOBAL REFLECTOR indicates if the global array is reflected.

USE NESTED HOLES is set YES if holes are nested deeper than one level.

NUMBER OF HOLES is the number of HOLES that are entered in the geometry region data (Sect. F11.4.4).

The MAXIMUM HOLE NESTING LEVEL is the deepest level of hole nesting.

USE NESTED ARRAYS is set YES if arrays are nested deeper than one level.

The NUMBER OF ARRAYS USED is the number of array descriptions (Sect. F11.4.5) actually used in the problem description.

MAXIMUM ARRAY NESTING LEVEL is the deepest level of array nesting.

There are six BOUNDARY CONDITIONs printed near the bottom of the table. They show the type of boundary condition that is applied to each face of the system. These should all be VACUUM unless albedo boundary conditions are applied to one or more faces of the system as described in Sect. F11.4.6. Also refer to the NUMBER OF DIFFERENTIAL ALBEDOS USED, discussed previously in the description of this table of information.

F11.6.13 TABLE OF SPACE AND SUPERGROUP INFORMATION

This is the fifth table printed by KENO V. It summarizes the space requirements for the problem and prints information related to the supergroups. This table is printed by subroutines NSUPG and LIMLN.

```
SAMPLE PROBLEM 16   IP27 DEMONSTRATION OF OPTIONS PROBLEM

                ****** SPACE AND SUPERGROUP INFORMATION ******

      20000 WORDS IS THE TOTAL SPACE AVAILABLE.

      15833 WORDS WERE USED FOR NON-SUPERGROUP STORAGE.

       4167 WORDS OF STORAGE ARE AVAILABLE FOR SUPERGROUPED DATA.

      19599 WORDS OF STORAGE ARE AVAILABLE FOR CONSTRUCTING THE SUPERGROUPS.

       4114 WORDS OF STORAGE ARE AVAILABLE TO EACH SUPERGROUP.

        727 WORDS ARE NEEDED FOR THE LARGEST GROUP.

      16704 WORDS OF STORAGE IS SUFFICIENT TO RUN THIS PROBLEM.

      24024 WORDS OF STORAGE WILL ALLOW THE PROBLEM TO RUN WITH ONE SUPERGROUP.

 SUPERGROUP   STARTING GROUP   ENDING GROUP   XSEC LENGTH   ALBEDO LENGTH   TOTAL LENGTH

          1               1              1            28              68            727
          2               2              7           148             308           3622
          3               8             16           156             180           3931

 ..... 206 IO'S WERE USED IN SUPERGROUPING .....
```

Fig. F11.6.15. Sample table of space and supergroup information

This table contains information about the space requirements of the problem and the number of supergroups used in the problem. The table is basically self-explanatory.

The TOTAL SPACE AVAILABLE is the amount of memory, in words, available to contain the data for the problem. The NONSUPERGROUP STORAGE is the number of words of memory containing nonsupergrouped data.

The bottom portion of this table contains information about the supergroups. Warning and error messages may appear in this portion of the table. The printed supergroup data include the supergroup number, the first energy group in the supergroup, the last energy group in the supergroup, the length (in words) of the cross-section data for the supergroup, the length (in words) of the albedo data for the supergroup, and the total length (in words) of all the data for the supergroup. This information is printed for each supergroup. Following the supergroup data are statements concerning the amount of storage available for supergrouping, the amount of storage available for each supergroup, the number of words of storage needed for the largest supergroup, the amount of storage needed to run the problem, and the amount of storage that would be necessary to run the problem with one supergroup.

F11.6.14 ARRAY SUMMARY

The arrays that are used in the problem are summarized in the table shown in Fig. F11.6.16. This table is printed by subroutine PRTARA whenever more than one array is used in the problem.

```
 ARRAY NUMBER   UNITS IN X DIR.   UNITS IN Y DIR.   UNITS IN Z DIR.   NESTING LEVEL

            1               3                 2                 2               3
            2               2                 2                 1               2
            3               1                 2                 3               2
            4               3                 1                 3               2
            5 GLOBAL        1                 1                 1               1

 ..... 11 IO'S WERE USED LOADING THE DATA .....
```

Fig. F11.6.16. Example of array summary

The ARRAY NUMBER is the number by which the array is designated in the input data. The number of units in the x, y, and z directions is listed for each array. The NESTING LEVEL indicates the level of nesting for each array. The global, overall, or universe array is flagged by the word GLOBAL. The global array should always appear at the first nesting level. Arrays that have been placed in the global reflector by using holes should also appear at the first nesting level. A nesting level of one is the highest or first nesting level. The larger the number in the nesting level column, the deeper the nesting level.

F11.6.15 GEOMETRY DATA

The geometry region data utilized by the problem are printed by subroutine PRTJOM and cannot be suppressed. They should be carefully examined by the user to verify the mixture number, bias ID, and geometry specifications used in the problem. If geometry region data are input but are not referenced in the unit orientation array data, they will not be printed here. An example would be to input geometry region data describing Units 1, 2, 3, and 4 and to utilize only Units 1, 3, and 4 in the unit orientation array. Then the geometry region data for Unit 2 will not be printed. An example of the geometry region printout for a problem is given in Fig. F11.6.17.

SAMPLE PROBLEM 18 1F27 DEMONSTRATION OF OPTIONS PROBLEM

REGION	MEDIA NUM	BIAS ID	GEOMETRY DESCRIPTION FOR THOSE UNITS UTILIZED IN THIS PROBLEM					
			----- UNIT 1 -----					
1 CYLINDER	1	1	RADIUS = 9.5200	+X = 9.7804	-X = -9.7804	CENTERLINE IS AT	X = .0	Y = .0
2 CYLINDER	8	1	RADIUS = 9.5200	+X = 9.9896	-X = -9.7804	CENTERLINE IS AT	X = .0	Y = .0
3 CYLINDER	2	1	RADIUS = 10.160	+X = 9.6296	-X = -9.4304	CENTERLINE IS AT	X = .0	Y = .0
4 CUBOID	6	1	+X = 18.450	-X = -18.450	+Y = 18.450	-Y = -18.450	+Z = 17.895	-Z = -17.685
			----- UNIT 2 EXTERNAL TO LATTICE 1 -----					
1 ARRAY NUMBER	1		+X = 73.800	-X = .0	+Y = 73.800	-Y = .0	+Z = 71.160	-Z = .0
			----- UNIT 3 EXTERNAL TO LATTICE 2 -----					
1 ARRAY NUMBER	2		+X = 73.800	-X = .0	+Y = 73.800	-Y = .0	+Z = 35.580	-Z = .0
			----- UNIT 4 EXTERNAL TO LATTICE 3 -----					
1 ARRAY NUMBER	3		+X = 36.900	-X = .0	+Y = 73.800	-Y = .0	+Z = 106.74	-Z = .0
			----- UNIT 5 EXTERNAL TO LATTICE 4 -----					
1 ARRAY NUMBER	4		+X = 110.70	-X = .0	+Y = 36.900	-Y = .0	+Z = 106.74	-Z = .0
			********** GLOBAL UNIT 6 **********					
			-----					
1 CUBOID	6	1	+X = 55.350	-X = -55.350	+Y = 55.350	-Y = -55.350	+Z = 53.370	-Z = -53.370
HOLE NUMBER	1		AT X = -55.350	Y = -18.450	Z = -17.790	IS UNIT NUMBER	2	
HOLE NUMBER	2		AT X = -55.350	Y = -18.450	Z = -53.370	IS UNIT NUMBER	3	
HOLE NUMBER	3		AT X = 18.450	Y = -18.450	Z = -53.370	IS UNIT NUMBER	4	
HOLE NUMBER	4		AT X = -55.350	Y = -55.350	Z = -53.370	IS UNIT NUMBER	5	
2 CUBOID	3	2	+X = 58.350	-X = -58.350	+Y = 58.350	-Y = -58.350	+Z = 56.370	-Z = -56.370
3 CUBOID	3	3	+X = 61.350	-X = -61.350	+Y = 61.350	-Y = -61.350	+Z = 59.370	-Z = -59.370
4 CUBOID	3	4	+X = 64.350	-X = -64.350	+Y = 64.350	-Y = -64.350	+Z = 62.370	-Z = -62.370
5 CUBOID	3	5	+X = 67.350	-X = -67.350	+Y = 67.350	-Y = -67.350	+Z = 65.370	-Z = -65.370
6 CUBOID	3	6	+X = 70.350	-X = -70.350	+Y = 70.350	-Y = -70.350	+Z = 68.370	-Z = -68.370
7 CUBOID	3	7	+X = 70.590	-X = -70.590	+Y = 70.590	-Y = -70.590	+Z = 68.610	-Z = -68.610

Fig. F11.6.17. Example of geometry region data

The problem title and a heading are printed at the top of each page. REGION is the region number within a unit. Each unit has its regions numbered sequentially, beginning with one. MEDIA NUM is the mixture number or mixture ID that occupies the volume defined by the region. BIAS ID is the bias ID that corresponds to the desired set of weight average for biasing the region. The unit number is printed at the top of each unit's geometry region description, near the center of the page. The data printed for each geometry region include (1) the region number relative to the unit (numbered sequentially within the unit), (2) the shape of the geometry region, (3) the mixture ID of the material within the volume defined by the region, (4) the bias ID to define the average weight of a neutron in the region, and (5) the dimensions defining the outer boundaries of the geometry region. If additional geometry surrounds an array, a heading is printed stating: UNIT _____ EXTERNAL TO LATTICE _____ The lattice number is the number of the array that is surrounded by the specified geometry. The unit number is the unit or box type that contains the specified geometry. In the case of an external reflector for the global array, the unit number is assigned by the code.

F11.6.16 UNIT ORIENTATION DESCRIPTION

Each unit orientation description defines the location of units in the three-dimensional lattice that represents the specified array. The array that is described is identified in the heading: UNIT ORIENTATION DESCRIPTION FOR ARRAY _____ The arrays used in the problem are stacked together to represent the physical problem being analyzed. The unit orientation description is not printed if only Unit 1 is described in the problem. The user should carefully examine the unit orientation descriptions to assure proper placement of the units in each lattice. A sample unit orientation description is shown in Fig. F11.6.18.

```
                    SAMPLE PROBLEM 2  2C8 BASE WITH 8 UNIT TYPES MATRIX CALCULATION

                 --------  UNIT ORIENTATION DESCRIPTION FOR ARRAY    1  --------

 1 LAYER    1, X COLUMN    1 TO    2  LEFT TO RIGHT    Y ROW    1 TO    2  BOTTOM TO TOP
  1 4
  1 2

 2 LAYER    2 X COLUMN    1 TO    2  LEFT TO RIGHT    Y ROW    1 TO    2  BOTTOM TO TOP
  7 8
  3 6
```

Fig. F11.6.18. Example of unit orientation description

If a very large array is utilized by the problem, its unit orientation description may be spread over several pages. When checking the printout, the user should pay careful attention to the headings which indicate the portion of each lattice being printed. The unit orientation descriptions are printed in subroutine PRTLBA.

F11.6.17 VOLUME INFORMATION

The calculated volumes are printed by subroutine VOLUME and cannot be suppressed. The data printed include the volume of each individual region, the cumulative volumes for each region in the unit and the total volume of each region throughout the entire lattice. An example of the volume printout is given in Fig. F11.6.19.

The unit number is printed under the heading UNIT. The REGION refers to the number of the geometry region within the unit. The geometry regions within a unit are numbered sequentially, starting with one. GEOMETRY CARD refers to the entry number for the individual geometry region. These are numbered sequentially, starting with one, through the TOTAL INPUT GEOMETRY REGIONS defined in Sect. F11.6.12. The VOLUME is the net volume of the individual region. The CUMULATIVE VOLUME is calculated from the dimensions of the region. For example, assume region 1 is a cube 3 cm on a side, region 2 is a cube 4 cm on a side, and region 3 is a cube 5 cm on a side. The CUMULATIVE VOLUME of region 1 is 9 cc, of region 2 is 16 cc, and region 3 is 25 cc.

```
                SAMPLE PROBLEM 10   1F27 DEMONSTRATION OF OPTIONS PROBLEM
                VOLUMES FOR THOSE UNITS UTILIZED IN  THIS PROBLEM

                          GEOMETRY                                  CUMULATIVE
UNIT      REGION           REGION              VOLUME                VOLUME

  1            1                1        6.999978E+03 CM**3        6.999978E+03 CM**3
               2                2        5.956648E+01 CM**3        5.059548E+03 CM**3
               3                3        1.117823E+03 CM**3        6.177778E+03 CM**3
               4                4        6.226823E+04 CM**3        6.844608E+04 CM**3

SURROUNDING GEOMETRY VOLUMES -  GEOMETRY REGION      5 IS AN ARRAY PLACEMENT BOUNDARY REGION

  2            1                5        3.875688E+05 CM**3        3.875688E+05 CM**3

SURROUNDING GEOMETRY VOLUMES -  GEOMETRY REGION      6 IS AN ARRAY PLACEMENT BOUNDARY REGION

  3            1                6        1.937844E+05 CM**3        1.937844E+05 CM**3

SURROUNDING GEOMETRY VOLUMES -  GEOMETRY REGION      7 IS AN ARRAY PLACEMENT BOUNDARY REGION

  4            1                7        2.906768E+05 CM**3        2.906768E+05 CM**3

SURROUNDING GEOMETRY VOLUMES -  GEOMETRY REGION      8 IS AN ARRAY PLACEMENT BOUNDARY REGION

  5            1                8        4.368148E+05 CM**3        4.368148E+05 CM**3

  6            1                9        6.875008E+00 CM**3        1.388058E+06 CM**3
               2               10        2.273508E+05 CM**3        1.535408E+06 CM**3
               3               11        2.522738E+05 CM**3        1.787678E+06 CM**3
               4               12        2.788988E+05 CM**3        2.066168E+06 CM**3
               5               13        3.060058E+05 CM**3        2.372178E+06 CM**3
               6               14        3.348148E+05 CM**3        2.706988E+06 CM**3
               7               15        2.806888E+04 CM**3        2.735058E+06 CM**3

                    UNIT      USES      REGION          TOTAL VOLUME

                      1         27           1      1.349998E+05 CM**3
                                             2      1.600298E+03 CM**3
                                             3      3.019238E+04 CM**3
                                             4      1.141248E+06 CM**3

                      2          1           1      3.875688E+05 CM**3

                      3          1           1      1.937844E+05 CM**3

                      4          1           1      2.906768E+05 CM**3

                      5          1           1      4.368148E+05 CM**3

                      6          1           1      6.875008E+00 CM**3
                                             2      2.273508E+05 CM**3
                                             3      2.522738E+05 CM**3
                                             4      2.788988E+05 CM**3
                                             5      3.060058E+05 CM**3
                                             6      3.348148E+05 CM**3
                                             7      2.806888E+04 CM**3
```

Fig. F11.6.19. Sample volume information

The VOLUME of region 1 is 9 cc, region 2 is 5 cc (16-9), and region 3 is 9 cc (25-16). The TOTAL VOLUME of each region is determined by multiplying the VOLUME for a region by the number of times the unit containing that region is used in the problem. The number of times a unit is used in the problem is listed under the heading USES.

F11.6.18 BIASING INFORMATION

This table specifies the weighting or biasing data to be used in the problem. An example of biasing information is given in Fig. F11.6.20.

```
*****************************************************************************************
***                                                                                   ***
***                                  BIASING INFORMATION                              ***
***                                                                                   ***
***   WEIGHTING INTERVALS   1 TO   6 FOR PARAFFIN    ,MAT ID=  400 WILL BE USED FOR BIAS ID'S   2 TO   7   ***
***                                                                                   ***
***   A DEFAULT WEIGHT OF   0.500 WILL BE USED FOR ALL OTHER BIAS ID'S.               ***
***                                                                                   ***
*****************************************************************************************
```

Fig. F11.6.20. Biasing information

The user is responsible for determining from the input data whether the group-dependent weights (*wtavg*) for the specified material(s) were obtained from the weighting library or were entered by the user. The group-dependent weights can be printed for verification purposes as shown in Sect. F11.6.19.

F11.6.19 GROUP-DEPENDENT WEIGHTS

Printing the group-dependent weights is optional. They are printed by subroutine PRTWTS if PWT=YES (Sect. F11.4.3) is entered in the parameter data. An example of the printed group-dependent weights is shown in Fig. F11.6.21.

SAMPLE PROBLEM 16 1F27 DEMONSTRATION OF OPTIONS PROBLEM

GROUP DEPENDENT WEIGHTS

ENERGY GROUP	BIAS ID 1	BIAS ID 2	BIAS ID 3	BIAS ID 4	BIAS ID 5	BIAS ID 6	BIAS ID 7
1	5.980000E-01	5.960000E-01	8.460000E-01	1.280000E+00	2.080000E+00	3.510000E+00	6.080000E+00
2	5.890000E-01	5.890000E-01	8.730000E-01	1.500000E+00	2.870000E+00	5.880000E+00	1.260000E+01
3	5.830000E-01	5.830000E-01	9.150000E-01	1.800000E+00	4.170000E+00	1.070000E+01	2.940000E+01
4	5.770000E-01	5.770000E-01	9.720000E-01	2.170000E+00	5.800000E+00	1.730000E+01	5.530000E+01
5	5.960000E-01	5.960000E-01	1.140000E+00	2.990000E+00	9.420000E+00	3.280000E+01	1.200000E+02
6	5.980000E-01	6.480000E-01	1.440000E+00	4.440000E+00	1.610000E+01	6.190000E+01	2.420000E+02
7	6.710000E-01	6.710000E-01	1.630000E+00	5.360000E+00	2.020000E+01	7.850000E+01	3.090000E+02
8	6.840000E-01	6.840000E-01	1.750000E+00	6.020000E+00	2.300000E+01	9.010000E+01	3.560000E+02
9	6.970000E-01	6.970000E-01	1.840000E+00	6.570000E+00	2.540000E+01	9.970000E+01	3.930000E+02
10	6.730000E-01	6.730000E-01	1.880000E+00	6.950000E+00	2.710000E+01	1.070000E+02	4.200000E+02
11	6.480000E-01	6.480000E-01	1.890000E+00	7.120000E+00	2.790000E+01	1.100000E+02	4.320000E+02
12	6.310000E-01	6.310000E-01	1.950000E+00	7.480000E+00	2.940000E+01	1.160000E+02	4.550000E+02
13	7.680000E-01	7.680000E-01	2.650000E+00	1.030000E+01	4.050000E+01	1.590000E+02	6.260000E+02
14	8.180000E-01	8.180000E-01	2.970000E+00	1.160000E+01	4.560000E+01	1.800000E+02	7.070000E+02
15	8.240000E-01	8.240000E-01	3.050000E+00	1.190000E+01	4.700000E+01	1.850000E+02	7.290000E+02
16	8.910000E-01	8.910000E-01	3.520000E+00	1.390000E+01	5.460000E+01	2.150000E+02	8.470000E+02

Fig. F11.6.21. Example of biasing data

The title is printed at the top of the table. The average weight (*wtavg*) is printed for each energy group and each BIAS ID. The BIAS ID number printed at the top of the column corresponds to the BIAS ID used in the geometry region description and printed in the biasing information.

F11.6.20 PICTURE OR PLOT REPRESENTATION

Printer plots representing two-dimensional slices through the geometrical description of the problem are optional. They are created if picture or plot data are entered as specified in Sect. F11.4.11 unless PLT=NO is specified either in the plot data or the parameter data (Sect. F11.4.3). A sample picture and its associated data are shown in Figs. F11.6.22 and F11.6.23.

```
!F27 XY PLOT AT X=0.0

                                MIXTURE MAP

MIXTURE    0   1   2   3   4
 SYMBOL        +   -   }
      OVERALL SYSTEM COORDINATES:
      XMIN= 0.0          XMAX= 1.61180E+02   YMIN= 0.0          YMAX= 1.61180E+02   XMIN= 0.0          XMAX= 1.37220E+02

      UPPER  LEFT        LOWER RIGHT
      COORDINATES        COORDINATES

X      0.0                3.1412E+03
Y      3.1412E+03         0.0
Z      0.6861E+02         0.6861E+02

         U AXIS      V AXIS
         (DOWN)      (ACROSS)

X        0.0          1.00000
Y       -1.00000      0.0
Z        0.0          0.0

NU=  104   NV=  130     DELU= 1.3575E+00     DELV= 1.0840E+00
```

Fig. F11.6.22. Summary of picture symbols and coordinates

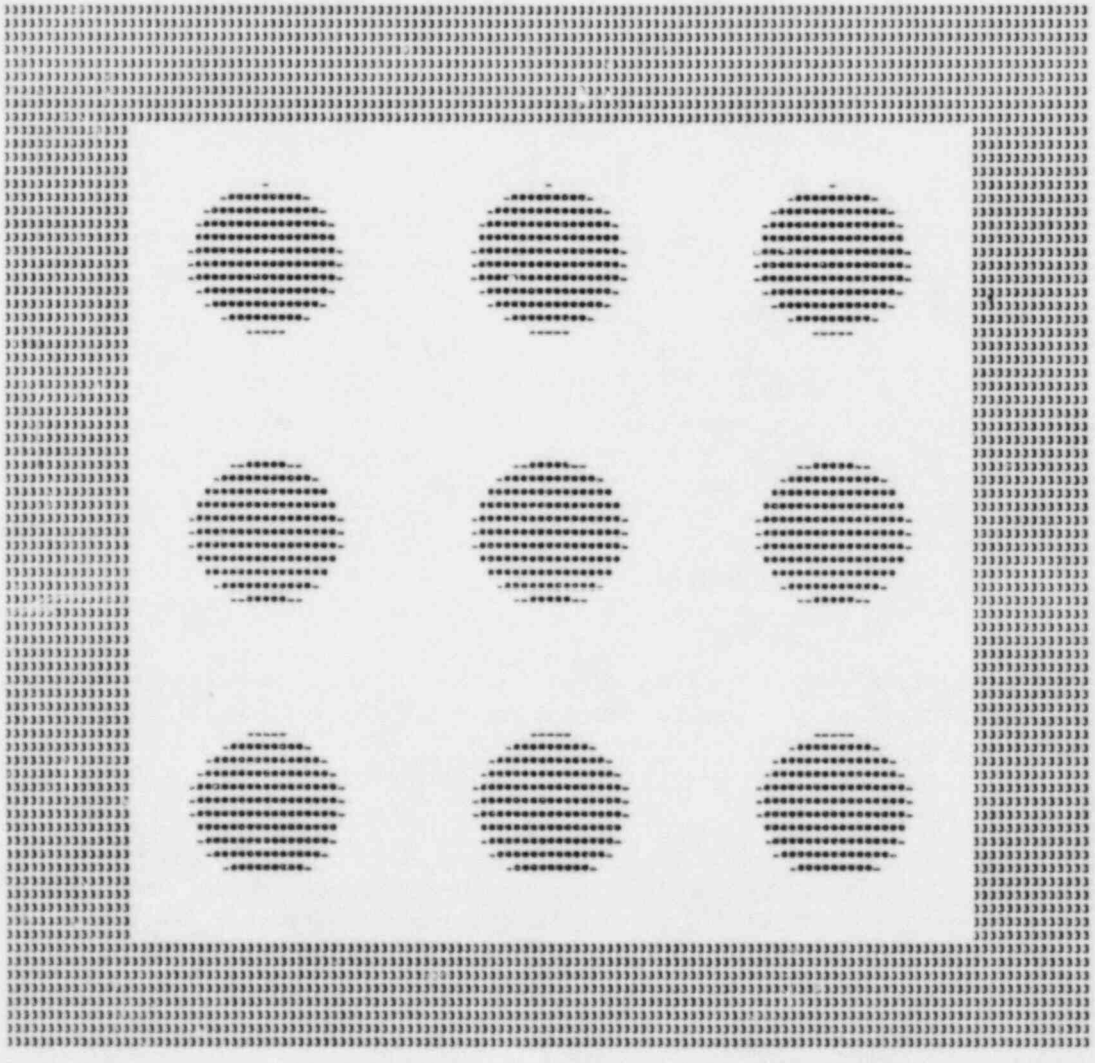


Fig. F11.6.23. Sample plot representation

Figure F11.6.22 summarizes the data used to generate the picture or plot. Figure F11.6.23 is an example of a printer plot of the two-dimensional slice specified through the geometrical description of the problem.

In Fig. F11.6.22, the plot title is printed at the top of the page. If a plot title was not entered in the plot data, the plot title is defaulted to the problem title. The title is followed by a heading specifying the type of picture (MIXTURE MAP, BIAS ID MAP or UNIT MAP). This is followed by a table that correlates the symbols to be used in the printer plot with the mixture numbers, bias ID numbers or unit numbers that were used in the problem. The overall system coordinates are then printed, followed by the coordinates of the upper left corner and lower right corner of the picture. This is followed by the direction cosines down and across the page. NU is the number of characters printed in the "U" direction, NV is the number of characters printed in the "V" direction, DELU is the incremental distance, in cm, represented by each character in the "U" direction and DELV is the incremental distance, in cm, represented by each character the the "V" direction.

Figure F11.6.23 shows a printer plot of a two-dimensional slice specified through the geometrical description of the problem. These plots aid the user in verifying that the problem is described correctly. Any number of plots can be made.

F11.6.21 CONCLUDE PRE-TRACKING EDITS

Prior to calculating the k-effectives for each generation, KENO V prints a final edit as shown in Fig. F11.6.24.

```
.........  877 IO'S WERE USED IN KENO-V BEFORE TRACKING        .........
.........  0.18117MINUTES WERE USED PROCESSING DATA.           .........

VOLUME FRACTION OF FISSILE MATERIAL IN THE CORE= 0.493598-01

START TYPE 6 WAS USED.

NEUTRONS WERE STARTED AT THE FOLLOWING POSITIONS.
NEUTRONS    1 TO  350 WERE STARTED AT X=    0.0          Y=    0.0          Z=    0.0
        RELATIVE TO THE UNIT LOCATED AT ARRAY POSITION (    1,    1,    1).

0.00017 MINUTES WERE REQUIRED FOR STARTING.   TOTAL ELAPSED TIME IS 0.18133 MINUTES.

STARTING POINTS WILL BE PRINTED USING THE DENOV TRACKING FORMAT BECAUSE START PARAMETER PSP= WAS ENTERED AS YES.
X, Y AND Z ARE THE COORDINATES OF THE STARTING POINT RELATIVE TO UNIT OR BOX TYPE LL LOCATED AT POSITION NBX, NBY, NBZ
IN THE ARRAY.  FOR A SINGLE UNIT PROBLEM X, Y AND Z ARE GIVEN RELATIVE TO THE ORIGIN OF THE SYSTEM.
THE INFORMATION OF INTEREST IS X, Y, Z, NBX, NBY, NBZ, LL AND IG(ENERGY GROUP).
```

Fig. F11.6.24. Pre-tracking edits

The information printed in this edit includes the number of I/O's used prior to tracking, the time in minutes used for preparing and processing the data prior to preparing the initial source distribution, and the volume fraction of fissile material in the core. If RFL=YES (Sect. F11.4.8) was specified in the start data, the volume fraction message is changed to: VOLUME FRACTION OF FISSILE MATERIAL IN THE SYSTEM =.

Most of the remaining data printed in this edit are related to the choice of the initial neutron source distribution. Once the initial source distribution has been prepared, the time required for preparing them (starting) is printed, as well as the total cumulative time used by the code.

F11.6.22 PRINT STARTING POINTS

This printout is optional and is used to verify the initial source starting points. An example of this information is given in Fig. F11.6.25.

WHERE	NEUTRON	X U	Y V	Z W	K IG	NCW KCOR	NBX K1	NBY K2	NBZ IGEO	LL KCOL	KR NRIG	KI NG	WT RANDOM NUMBER
START	1	0.0 -0.399648E+00	0.0 0.160978E+00	0.104608E+00 -0.902438E+00	1 4	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	2	0.0 0.982699E+00	0.0 0.582249E-01	0.104608E+00 -0.175899E+00	1 2	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	3	0.0 0.777467E+00	0.0 0.623536E-01	0.104608E+00 -0.625822E+00	1 2	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	4	0.0 0.136768E+00	0.0 -0.287928E-01	0.104608E+00 0.990195E+00	1 4	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	5	0.0 -0.924968E+00	0.0 0.646898E-01	0.104608E+00 0.374647E+00	1 1	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	6	0.0 0.583068E+00	0.0 -0.211318E+00	0.104608E+00 -0.784678E+00	1 2	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	7	0.0 0.759468E+00	0.0 0.459138E+00	0.104608E+00 0.468968E+00	1 4	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	8	0.0 -0.350138E-01	0.0 -0.132598E+00	0.104608E+00 0.990398E+00	1 1	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	9	0.0 0.942938E+00	0.0 0.269678E+00	0.104608E+00 0.195358E+00	1 4	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	10	0.0 0.277488E+00	0.0 0.643548E+00	0.104608E+00 -0.713358E+00	1 1	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	11	0.0 -0.979538E+00	0.0 0.194628E+00	0.104608E+00 -0.507158E-01	1 3	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	12	0.0 -0.556448E+00	0.0 0.118948E-01	0.104608E+00 -0.830808E+00	1 4	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5
START	13	0.0 -0.672658E+00	0.0 -0.580188E+00	0.104608E+00 0.459288E+00	1 3	1 5	2 0	1 0	1 0	1 0	1 1	0 1	0.100008E+01 EP9CC32B80F5

Fig. F11.6.25. Example of initial source points

The information pertinent to the initial source distribution is printed two lines at a time and appears under the designated headings. For example, the coordinates X, Y, and Z are printed on one line and the direction cosines U, V, and W are printed directly under them. The data printed for each source neutron include the following:

NEUTRON is the ID number of the neutron.

X, Y, and Z are the coordinates of the starting point relative to the coordinate system of the unit.

K is the region number that contains the point X, Y, Z.

NBX, NBY, NBZ are the coordinates of the unit within the array.

LL is the unit or box type number.

KR is the mixture number present at the starting point.

KI is the bias ID number at the point.

WT is the current weight (WT is always 1.0 for a neutron when it is started).

U, V, and W are the direction cosines defining the direction the history is traveling.

IG is the energy group.

KCOR is the core boundary region of the array the neutron is in.

K1 is the region number of the first region in the unit.

K2 is the region number of the last region in the unit.

IGEO is an integer that defines the geometry shape of the region.

KCOL is the region number in which the last collision occurred.

NSIG is the position of the group IG in the supergroup.

SG is the supergroup number.

RANDOM NUMBER is the current random number.

When starting points are printed, many of the above named variables have not been initialized. For starting, the variables of interest are X, Y, Z, U, V, W, NBX, NBY, NBZ, and LL.

F11.6.23 K-EFFECTIVES BY GENERATION

At the completion of each generation, KENO V prints the k-effective for that generation and associated information. An example of this printout is given in Fig. F11.6.26.

The k-effectives for each generation are printed in subroutine FISFLX. The headings are printed by subroutine GUIDE. The problem title is printed at the top of the page. A descriptive heading is printed at the top of each column of data. The data that are printed include (1) the generation number, (2) the k-effective calculated for the generation, (3) the elapsed CPU time in minutes, (4) the average value of k-effective through the current generation (excluding the first two generations), (5) the deviation associated with the average k-effective, and (6) the matrix k-effective for the generation and the deviation associated with the matrix k-effective. The last two columns are filled with zeros if the user did not specify matrix k-effective calculations. The matrix k-effective is the largest eigenvalue of the fission production matrix. Matrix information can be calculated based on (1) position index, (2) unit number, (3) hole number, and (4) array number. The matrix k-effective printed in the sixth column is based on this order. If the matrix k-effective is calculated by position index, it is the one printed in the sixth column. The matrix k-effective by unit number is given second preference, followed by hole number and then array number.

SAMPLE PROBLEM 16 1F27 DEMONSTRATION OF OPTIONS PROBLEM

GENERATION	GENERATION K-EFFECTIVE	ELAPSED TIME MINUTES	AVERAGE K-EFFECTIVE	AVG K-EFF DEVIATION	MATRIX K-EFFECTIVE	MATRIX K-EFF DEVIATION
1	1.408948E+00	1.120008E-01	1.000000E+00	0.0	1.407738E+00	0.0
2	1.161448E+00	1.611678E-01	1.000000E+00	0.0	1.284159E+00	1.235418E-01
3	1.127398E+00	2.171678E-01	1.127398E+00	0.0	1.231698E+00	8.856378E-02
4	1.102528E+00	2.728338E-01	1.114958E+00	1.246798E-02	1.199228E+00	7.053498E-02
5	1.080488E+00	3.286678E-01	1.090138E+00	2.583768E-02	1.167368E+00	6.334948E-02
6	1.012988E+00	3.806678E-01	1.070848E+00	2.656548E-02	1.111548E+00	5.773868E-02
7	1.086408E+00	4.371678E-01	1.073558E+00	2.076778E-02	1.133258E+00	4.949178E-02
8	9.979178E-01	4.866678E-01	1.060958E+00	2.111908E-02	1.116308E+00	4.810358E-02
9	9.650358E-01	5.346678E-01	1.044398E+00	2.435218E-02	1.087338E+00	4.491108E-02
10	1.063168E+00	5.883338E-01	1.046738E+00	2.131648E-02	1.093768E+00	4.031858E-02
11	9.780468E-01	6.375008E-01	1.038568E+00	2.038568E-02	1.082848E+00	3.807198E-02
12	1.104958E+00	7.123338E-01	1.045248E+00	1.937078E-02	1.088588E+00	3.479898E-02
13	1.076088E+00	7.645008E-01	1.047868E+00	1.772678E-02	1.083728E+00	3.282268E-02
14	1.035268E+00	8.120008E-01	1.046818E+00	1.621858E-02	1.080228E+00	2.995418E-02
15	9.646638E-01	8.565008E-01	1.038958E+00	1.668578E-02	1.071168E+00	2.923328E-02
16	1.102488E+00	9.080008E-01	1.043498E+00	1.625558E-02	1.073078E+00	2.741228E-02
17	9.773028E-01	9.640008E-01	1.039088E+00	1.575968E-02	1.067428E+00	2.636388E-02
18	1.012058E+00	1.010838E+00	1.037398E+00	1.483828E-02	1.064318E+00	2.508948E-02
19	1.091948E+00	1.067508E+00	1.040608E+00	1.430468E-02	1.065738E+00	2.373688E-02
20	1.070918E+00	1.123508E+00	1.042288E+00	1.359378E-02	1.065968E+00	2.252088E-02
21	9.887948E-01	1.170508E+00	1.039468E+00	1.316458E-02	1.062258E+00	2.173918E-02
22	1.079858E+00	1.225178E+00	1.041488E+00	1.284788E-02	1.063938E+00	2.076188E-02
23	9.943708E-01	1.283678E+00	1.038678E+00	1.392768E-02	1.055678E+00	2.113978E-02
24	1.018568E+00	1.333838E+00	1.033758E+00	1.329648E-02	1.054108E+00	2.030848E-02
25	9.742518E-01	1.383678E+00	1.031168E+00	1.296738E-02	1.050898E+00	1.973498E-02
26	9.405978E-01	1.440508E+00	1.027398E+00	1.297958E-02	1.046638E+00	1.943238E-02
27	1.063468E+00	1.494838E+00	1.028038E+00	1.253158E-02	1.047238E+00	1.879928E-02
28	9.461158E-01	1.547008E+00	1.025658E+00	1.245728E-02	1.043608E+00	1.839028E-02
29	1.007258E+00	1.592338E+00	1.024978E+00	1.208338E-02	1.042338E+00	1.779068E-02
30	1.031538E+00	1.641678E+00	1.025208E+00	1.157168E-02	1.041958E+00	1.719108E-02
31	1.034968E+00	1.693178E+00	1.025548E+00	1.117278E-02	1.041718E+00	1.662908E-02
32	1.038728E+00	1.743838E+00	1.025988E+00	1.080028E-02	1.041608E+00	1.610178E-02
33	1.043118E+00	1.793178E+00	1.026538E+00	1.046348E-02	1.041638E+00	1.560648E-02
34	9.632148E-01	1.849508E+00	1.024558E+00	1.031868E-02	1.039318E+00	1.531728E-02
35	1.027288E+00	1.897678E+00	1.024638E+00	1.000388E-02	1.038958E+00	1.487418E-02
36	1.013708E+00	1.946508E+00	1.023178E+00	9.713218E-03	1.038238E+00	1.447668E-02
37	1.111608E+00	1.996678E+00	1.026888E+00	9.756138E-03	1.040198E+00	1.421628E-02
38	9.599628E-01	2.055508E+00	1.024958E+00	9.660438E-03	1.038078E+00	1.399898E-02
39	9.747108E-01	2.108338E+00	1.023598E+00	9.494678E-03	1.036438E+00	1.373358E-02
40	1.044188E+00	2.156838E+00	1.024138E+00	9.257538E-03	1.036618E+00	1.338708E-02
41	1.043148E+00	2.216338E+00	1.024628E+00	9.023998E-03	1.036768E+00	1.305758E-02
42	9.953518E-01	2.271508E+00	1.023898E+00	8.830708E-03	1.035768E+00	1.278218E-02
43	9.663778E-01	2.332178E+00	1.022488E+00	8.726458E-03	1.034148E+00	1.258678E-02
44	1.057798E+00	2.391838E+00	1.023328E+00	8.558448E-03	1.034668E+00	1.230888E-02
45	1.026778E+00	2.455338E+00	1.023408E+00	8.357358E-03	1.034478E+00	1.203388E-02
46	1.019738E+00	2.510678E+00	1.023328E+00	8.166448E-03	1.034148E+00	1.177368E-02
47	1.094558E+00	2.562508E+00	1.024208E+00	8.137288E-03	1.035478E+00	1.159048E-02
48	9.636468E-01	2.614008E+00	1.022818E+00	8.012058E-03	1.034338E+00	1.139908E-02
49	1.004648E+00	2.671178E+00	1.023598E+00	7.889568E-03	1.033718E+00	1.118168E-02
50	9.709988E-01	2.720008E+00	1.022588E+00	7.762998E-03	1.032488E+00	1.102888E-02
51	9.115638E-01	2.772178E+00	1.020238E+00	7.931128E-03	1.030078E+00	1.106858E-02
52	9.845558E-01	2.819068E+00	1.019528E+00	7.803708E-03	1.029188E+00	1.088948E-02

KENO MESSAGE NUMBER KS-121 FOLLOWS:
EXECUTION TERMINATED DUE TO INSUFFICIENT TIME IN THE JOB STEP.
4.76 SECONDS ARE NEEDED PER GENERATION, BUT ONLY 2.81 REMAIN IN THE JOB STEP.

THE MATRIX K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY UNIT NUMBER MATRIX.

Fig. F11.6.26. Example of k-effectives by generation

After the last generation, a message is printed to indicate why execution was terminated. If matrix k-effectives were calculated, this is followed by a message stating the method used to determine the matrix k-effective.

The user should examine this portion of the printed results to assure that the two methods of calculating k-effective are in acceptable agreement and to verify that the average value of k-effective has become relatively stable. If the k-effectives appear to be oscillating or drifting significantly, the user should consider rerunning the problem with a larger number of histories per generation.

If a problem is restarted, the generation numbers and k-effectives are printed and the words FROM RESTART UNIT are printed in the elapsed time column. All other columns are blank. When the generation at which the problem is to be restarted is reached, the print reverts to the normal format as shown in Fig. F11.6.26.

F11.6.24 FINAL K-EFFECTIVE EDIT

The final edit of the k-effectives is printed by subroutine KEDIT following the k-effectives by generation. The title is printed at the top of the page, followed by the lifetime and the generation time and their associated deviations. The lifetime is the average lifespan of a neutron (in seconds) from the time it is born until it is absorbed or leaks from the system. The generation time is the average time (in seconds) between successive neutron generations. If NUB=YES is specified in the parameter data, (Sect. F11.4.3) the average number of neutrons per fission, NU BAR and its associated deviation are printed and the AVERAGE FISSION GROUP (the average energy group at which fission occurs) and its associated deviation are printed. If SMU=YES is specified in the parameter data, the average self-multiplication of a unit and its associated deviation is printed. This self-multiplication results from fissions caused by neutrons born in the unit. Fissions caused by neutrons that exit the unit and return are not included. Then the final k-effective edit is printed as shown in Fig. F11.6.27.

SAMPLE PROBLEM 18 !F27 DEMONSTRATION OF OPTIONS PROBLEM

LIFETIME = 2.646238-04 +OR- 1.677928-05 GENERATION TIME = 1.245928-04 +OR- 3.198438-06
NU BAR = 2.453848+00 +OR- 4.373188-05 AVERAGE FISSION GROUP = 1.440578+01 +OR- 8.245128-03

NO. OF INITIAL GENERATIONS SKIPPED	AVERAGE K-EFFECTIVE	DEVIATION	67 PER CENT CONFIDENCE INTERVAL	95 PER CENT CONFIDENCE INTERVAL	99 PER CENT CONFIDENCE INTERVAL	NUMBER OF HISTORIES
3	1.01732	+OR- 0.00763	1.00968 TO 1.02495	1.00205 TO 1.03259	0.99442 TO 1.04022	17150
4	1.01554	+OR- 0.00758	1.00796 TO 1.02312	1.00038 TO 1.03071	0.99280 TO 1.03829	16800
5	1.01501	+OR- 0.00772	1.00729 TO 1.02274	0.99956 TO 1.03046	0.99184 TO 1.03819	16450
6	1.01506	+OR- 0.00789	1.00716 TO 1.02295	0.99927 TO 1.03085	0.99137 TO 1.03874	16100
7	1.01352	+OR- 0.00792	1.00560 TO 1.02143	0.99768 TO 1.02935	0.98977 TO 1.03727	15750
8	1.01387	+OR- 0.00809	1.00578 TO 1.02196	0.99769 TO 1.03005	0.98960 TO 1.03814	15400
9	1.01547	+OR- 0.00813	1.00734 TO 1.02359	0.99924 TO 1.03171	0.99112 TO 1.03982	15059
10	1.01439	+OR- 0.00823	1.00611 TO 1.02257	0.99788 TO 1.03080	0.98964 TO 1.03903	14708
11	1.01532	+OR- 0.00837	1.00695 TO 1.02369	0.99857 TO 1.03207	0.99020 TO 1.04044	14358
12	1.01389	+OR- 0.00828	1.00482 TO 1.02137	0.99654 TO 1.02965	0.98826 TO 1.03792	14008
17	1.01114	+OR- 0.00863	1.00251 TO 1.01977	0.99388 TO 1.02839	0.98526 TO 1.03702	12258
22	1.00488	+OR- 0.00910	0.99578 TO 1.01398	0.98668 TO 1.02309	0.97758 TO 1.03219	10508
27	1.01021	+OR- 0.00917	1.00105 TO 1.01938	0.99188 TO 1.02855	0.98271 TO 1.03772	8758
32	1.00984	+OR- 0.01041	0.99943 TO 1.02066	0.98821 TO 1.03147	0.97740 TO 1.04228	7008
37	1.00253	+OR- 0.01189	0.99064 TO 1.01442	0.97875 TO 1.02631	0.96686 TO 1.03820	5258
42	1.00206	+OR- 0.01621	0.98585 TO 1.01828	0.96963 TO 1.03449	0.95342 TO 1.05071	3508
47	0.97108	+OR- 0.01582	0.95526 TO 0.98691	0.93943 TO 1.00273	0.92361 TO 1.01856	1758

Fig. F11.6.27. Example of the final k-effective edit

The final k-effective edit prints the average k-effective and its associated deviation and the limits of k-effective for the 67, 95, and 99% confidence intervals. The number of histories used in calculating the average k-effective is also printed. This is done skipping various numbers of generations. The user should carefully examine the final k-effective edit to determine if the average k-effective is relatively stable. If a noticeable drift is apparent as the number of initial generations skipped increases, it may indicate a problem in converging the source. If this appears to be the case, the problem should be rerun with a better initial source distribution and should be run for a sufficient number of generations that the average k-effective becomes stable.

F11.6.25 PLOT OF AVERAGE K-EFFECTIVE BY GENERATION RUN

This plot consists of average k-effectives plotted versus the number of generations run. The limits of one standard deviation are plotted on either side of each average k-effective. These average k-effectives are not necessarily the same as the average k-effectives described in Sect. F11.6.23. The code omits the k-effectives of the first *nskip*-1 generations when the average k-effectives for this plot are calculated. Although the k-effective of the *nskip* generation is summed into the average k-effective, it is not plotted because standard deviations cannot be calculated for a single point. Thus, if *nskip* is 3 (i.e., the first three generations are skipped), the first value plotted is the average k-effective corresponding to the fourth generation. The dotted line represents the value of the average k-effective corresponding to the smallest deviation when the average k-effective and its deviation are computed for each generation over the range of nskip through the total number of generations. Figure F11.6.28 is an example of this type of plot. The primary use for this plot is to determine if the problem has source convergence difficulties.

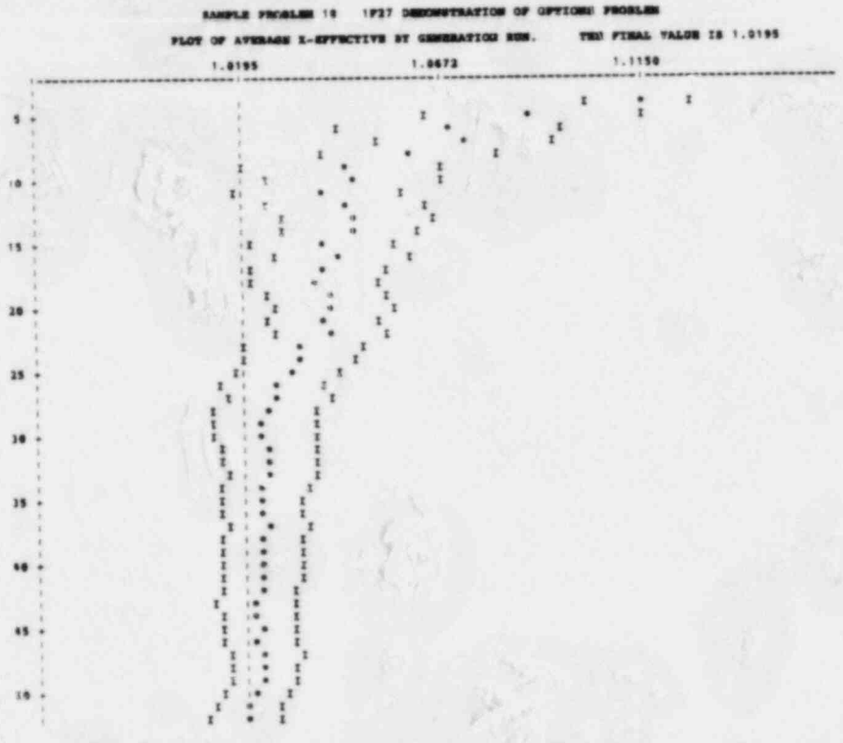


Fig. F11.6.28. Sample plot of average k-effective by generation run

F11.6.26 PLOT OF AVERAGE K-EFFECTIVE BY GENERATION SKIPPED

This plot illustrates the average k-effective versus the number of generations skipped as shown in Fig. F11.6.29. The limits of one standard deviation are plotted on either side of the average k-effective. The dotted line represents the value of the average k-effective corresponding to the smallest deviation when the average k-effective and its deviation are computed for the number of generations skipped over the range of nskip-1 through two less than the total number of generations calculated. The plot is essentially a plot of the data described in Sect. F11.6.24. It is useful for determining if source convergence has been achieved.

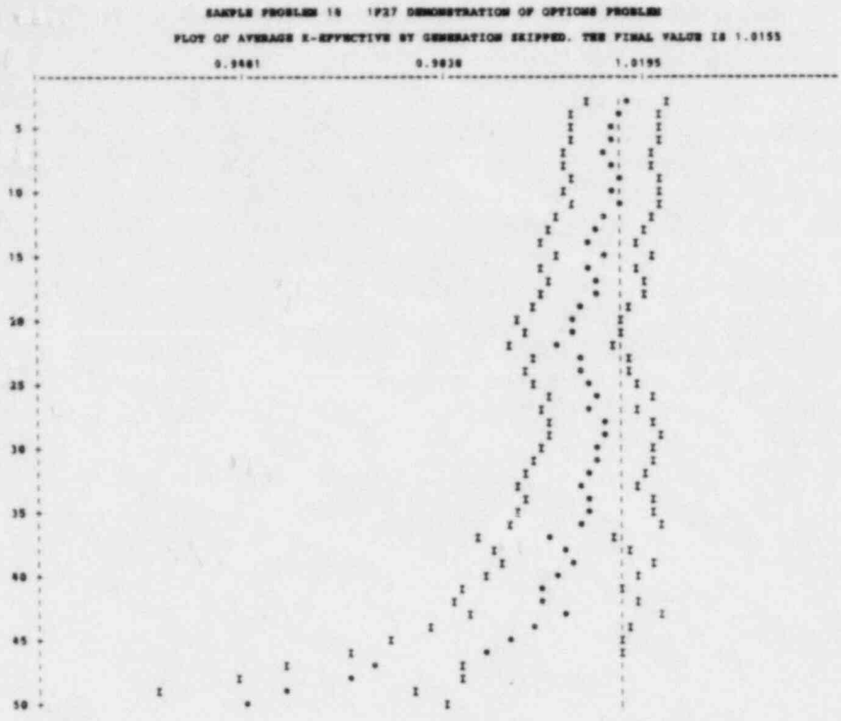


Fig. F11.6.29. Sample plot of average k-effective by generation skipped

F11.6.27 FINAL EDIT OF FISSIONS, ABSORPTIONS, AND LEAKAGE

This table is printed by subroutine KEDIT after the final k-effective edit. It prints the fission fraction for each group and the fission production, absorptions, and leakage with their associated percent deviation. If FAR=YES is specified in the parameter data, the fission production and absorptions and their percent deviation are also printed for each geometry region utilized in the problem. The FISSIONS, ABSORPTIONS, and LEAKAGE are given in units of neutrons per source neutron. The samples of the final edits of fissions, absorptions, and leakage are shown in Figs. F11.6.30 and F11.6.31.

If FAR=YES is specified in the parameter data, the fissions, and absorptions for each geometry region are printed for each energy group as shown in Fig. F11.6.31. Leakage is not collected by geometry region but rather represents the leakage from the system. In both figures, GROUP is the energy group number, FISSION FRACTION is the fraction of the fissions that occur in that energy group. The percent deviation for the fission fraction is the same as that of the fissions in the same group. The heading, UNIT, refers to the unit or box type, and REGION is the region number within the specified unit or box type. The geometry regions are numbered sequentially within each unit, starting with one. The sum of the fissions for every region for a given energy group is the total printed for that energy group. The same is true of absorptions. The fissions, absorptions, and leakages are given in units of "per source neutron." The SYSTEM TOTAL is the sum, over all the energy groups, of the fissions, absorptions and leakage. The associated percent deviation is printed for each.

The sum of the leakage and absorptions printed for the system total should be close to one. The fissions printed for the system total should be the same as the first k-effective printed in the final k-effective edit described in Sect. F11.6.24. If differential albedos are used, the leakage does not include the weight lost in the albedo reflection. A message stating the weight lost in the albedo is printed. This is the weight lost due to absorptions in the albedo reflector and leakage from the albedo reflector. There is no leakage associated with faces having specular, mirror image, or periodic reflection. Thus there is no leakage associated with an infinite problem. The total elapsed time and final random number are printed at the end of this edit.

SAMPLE PROBLEM 2 2C0 BARE WITH 6 UNIT TYPES MATRIX CALCULATION SKIPPING 3 GENERATIONS

GROUP	FISSION FRACTION	UNIT	REGION	FISSIONS	PERCENT DEVIATION	ABSORPTIONS	PERCENT DEVIATION	LEAKAGE	PERCENT DEVIATION
1	0.1446			1.43712D-01	1.4642	5.07491D-02	1.4642	7.98268D-02	1.9625
2	0.2389			2.37763D-01	1.1243	9.68543D-02	1.1243	1.45503D-01	1.4647
3	0.1446			1.43962D-01	1.5946	6.22898D-02	1.5946	9.26695D-02	1.3520
4	0.2273			2.26223D-01	1.1465	1.02884D-01	1.1465	1.46301D-01	0.9835
5	0.1964			1.95491D-01	1.4027	9.28766D-02	1.4027	9.41746D-02	1.1309
6	0.0477			4.74512D-02	2.6086	2.46553D-02	2.6686	1.13305D-02	3.7472
7	0.0008			8.16769D-04	24.0380	4.46576D-04	24.0380	2.06232D-04	26.5563
8	0.0			0.0	0.0	0.0	0.0	0.0	0.0
9	0.0			0.0	0.0	0.0	0.0	0.0	0.0
10	0.0			0.0	0.0	0.0	0.0	0.0	0.0
11	0.0			0.0	0.0	0.0	0.0	0.0	0.0
12	0.0			0.0	0.0	0.0	0.0	0.0	0.0
13	0.0			0.0	0.0	0.0	0.0	0.0	0.0
14	0.0			0.0	0.0	0.0	0.0	0.0	0.0
15	0.0			0.0	0.0	0.0	0.0	0.0	0.0
16	0.0			0.0	0.0	0.0	0.0	0.0	0.0
SYSTEM TOTAL =				9.95810D-01	0.4852	4.38769D-01	0.4869	5.79810D-01	0.3568

ELAPSED TIME = 0.37667 MINUTES

RANDOM NUMBER= 9F3639DCD175

Fig. F11.6.30. Sample of the final edit of fissions, absorptions, and leakage with region-dependent information suppressed

SAMPLE PROBLEM 16 (F27 DEMONSTRATION OF OPTIONS PROBLEM

SKIPPING 3 GENERATIONS

GROUP	FISSION FRACTION	UNIT	REGION	FISSIONS	PERCENT DEVIATION	ABSORPTIONS	PERCENT DEVIATION	LEAKAGE	PERCENT DEVIATION
15	0.1837			1.86896D-01	1.4588	1.86479D-01	1.3722	0.0	0.0
		1	1	1.86896D-01	1.4588	9.60901D-02	1.4588		
			2	0.0	0.0	0.0	0.0		
			3	0.0	0.0	6.77136D-04	3.4852		
			4	0.0	0.0	0.0	0.0		
		2	1	0.0	0.0	0.0	0.0		
		3	1	0.0	0.0	0.0	0.0		
		4	1	0.0	0.0	0.0	0.0		
		5	1	0.0	0.0	0.0	0.0		
		6	1	0.0	0.0	0.0	0.0		
			2	0.0	0.0	5.03679D-03	1.9810		
			3	0.0	0.0	2.38350D-03	4.5143		
			4	0.0	0.0	1.17314D-03	12.4949		
			5	0.0	0.0	8.80764D-06	42.7242		
			6	0.0	0.0	2.37616D-04	57.7853		
			7	0.0	0.0	0.0	0.0		
16	0.8082			8.18767D-01	0.8608	7.20053D-01	3.9278	0.0	0.0
		1	1	8.18767D-01	0.8608	3.09168D-01	0.8608		
			2	0.0	0.0	0.0	0.0		
			3	0.0	0.0	8.58896D-03	1.9287		
			4	0.0	0.0	0.0	0.0		
		2	1	0.0	0.0	0.0	0.0		
		3	1	0.0	0.0	0.0	0.0		
		4	1	0.0	0.0	0.0	0.0		
		5	1	0.0	0.0	0.0	0.0		
		6	1	0.0	0.0	0.0	0.0		
			2	0.0	0.0	1.56770D-01	2.1633		
			3	0.0	0.0	1.21758D-01	4.5042		
			4	0.0	0.0	6.86269D-02	14.1206		
			5	0.0	0.0	3.98139D-02	28.6564		
			6	0.0	0.0	1.53267D-02	59.0868		
			7	0.0	0.0	1.00258D-04	71.8429		
SYSTEM TOTAL =				1.01732D+00	0.7584	9.66993D-01	2.9329	1.31513D-02	28.2595

THE WEIGHT LOST IN THE ALBEDO PORTION OF THE PROBLEM = 8.79938-04 + OR - 0.0002

ELAPSED TIME 2.02100 MINUTES

RANDOM NUMBER= 87AB2E30CD1D

Fig. F11.6.31. Sample of the final edit of fissions, absorptions, and leakage by region

F11.6.28 MATRIX K-EFFECTIVE BY UNIT LOCATION

The matrix k-effective by unit location (also referred to as array position or position index) is calculated if MKP=YES is specified in the parameter data (Sect. F11.4.3). It is the largest eigenvalue of the fission production matrix, collected by position index. The position index is a number referencing a position in a three-dimensional lattice. An example of the matrix k-effective by unit location is given in Fig. F11.6.32. It is contained within two rows of asterisks to draw attention to it.

```
SAMPLE PROBLEM 2  2C8 BARE WITH 8 UNIT TYPES MATRIX CALCULATION
**************************************************************************************************************************

 POSITION K-EFFECTIVE=  9.933858-01  + OR -  2.379318-03
 THE POSITION K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY POSITION INDEX MATRIX.
**************************************************************************************************************************
```

Fig. F11.6.32. Example of matrix k-effective by unit location

F11.6.29 FISSION PRODUCTION BY POSITION INDEX MATRIX

To obtain this information, the user must specify MKP=YES and FMP=YES in the parameter data. It is then printed by subroutine MATRIX. The number of entries in the fission production matrix by position index is the square of the array size. Thus for a 2x2x2 array there are 64 entries, and for a 4x4x4 array there are 4096 entries in the fission production matrix by position index. An example of the fission production matrix by position index for a 2x2x2 array is shown in Fig. F11.6.33.

```
FISSION PRODUCTION BY POSITION INDEX MATRIX
( I, J) P IS THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED AT POSITION INDEX J BY A NEUTRON BORN AT POSITION INDEX I.
( 1, 1) 7.37E-01  ( 1, 2) 4.67E-02  ( 1, 3) 5.24E-02  ( 1, 4) 2.83E-02  ( 1, 5) 7.12E-03  ( 1, 6) 2.39E-03
( 1, 7) 2.26E-03  ( 1, 8) 1.76E-03
( 2, 1) 5.42E-01  ( 2, 2) 6.91E-01  ( 2, 3) 2.56E-02  ( 2, 4) 4.97E-02  ( 2, 5) 2.50E-02  ( 2, 6) 7.70E-03
( 2, 7) 1.70E-02  ( 2, 8) 2.49E-03
( 3, 1) 4.56E-02  ( 3, 2) 2.53E-02  ( 3, 3) 7.44E-01  ( 3, 4) 4.82E-02  ( 3, 5) 2.83E-02  ( 3, 6) 1.91E-02
( 3, 7) 6.67E-03  ( 3, 8) 3.26E-03
( 4, 1) 3.70E-02  ( 4, 2) 5.16E-02  ( 4, 3) 4.81E-02  ( 4, 4) 6.97E-01  ( 4, 5) 2.36E-02  ( 4, 6) 2.71E-02
( 4, 7) 2.94E-02  ( 4, 8) 7.21E-03
( 5, 1) 5.65E-02  ( 5, 2) 2.26E-02  ( 5, 3) 2.61E-02  ( 5, 4) 1.88E-02  ( 5, 5) 7.49E-01  ( 5, 6) 5.46E-02
( 5, 7) 5.12E-02  ( 5, 8) 2.40E-02
( 6, 1) 2.72E-02  ( 6, 2) 6.26E-02  ( 6, 3) 1.91E-02  ( 6, 4) 2.30E-02  ( 6, 5) 4.82E-02  ( 6, 6) 7.36E-01
( 6, 7) 2.73E-02  ( 6, 8) 5.19E-02
( 7, 1) 2.21E-02  ( 7, 2) 2.13E-02  ( 7, 3) 6.41E-02  ( 7, 4) 2.73E-02  ( 7, 5) 5.22E-02  ( 7, 6) 2.61E-02
( 7, 7) 7.25E-01  ( 7, 8) 5.37E-02
( 8, 1) 1.95E-02  ( 8, 2) 2.41E-02  ( 8, 3) 1.95E-02  ( 8, 4) 6.56E-02  ( 8, 5) 2.80E-02  ( 8, 6) 5.58E-02
( 8, 7) 5.62E-02  ( 8, 8) 7.32E-01
```

Fig. F11.6.33. Sample fission production matrix by position index

The position index definition is given in Sect. F11.6.31. For each position index in the array, the number of next generation neutrons produced at position index J per neutron born at position index I is determined. The fission production matrix by position index is used to determine the matrix k-effective, cofactor k-effective and source vector by position index.

F11.6.30 SOURCE VECTOR BY POSITION INDEX

This information is printed by subroutine MATRIX only if MKP=YES is specified in the parameter data. The source vector by position index is the eigenvector of the fission production matrix by position index and should sum to 1.0. It represents the fission source for the specified locations in the three-dimensional lattice representing the physical problem being analyzed. The position index is defined in Sect. F11.6.31. An example of the source vector by position index is shown in Fig. F11.6.34. The average self-multiplication by array position is the overall average of the self-multiplication of all units used in the problem.

```
SAMPLE PROBLEM 2  2C9 BARE WITH 8 UNIT TYPES MATRIX CALCULATION

 SOURCE VECTOR BY POSITION INDEX

 INDEX      VECTOR
   1     1.281698E-01
   2     1.457703E-01
   3     1.256278E-01
   4     1.091918E-01
   5     1.416532E-01
   6     1.351658E-01
   7     1.278438E-01
   8     1.257339E-01

                AVERAGE SELF MULTIPLICATION BY ARRAY POSITION

THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN A UNIT LOCATED AT A GIVEN POSITION IN THE ARRAY BY
A NEUTRON BORN IN THAT BARE UNIT IS   7.278928E-01 + OR -  2.937165E-03
```

Fig. F11.6.34. Example of source vector by position index

F11.6.31 COFACTOR K-EFFECTIVE BY POSITION INDEX

These data are printed by subroutine MATRIX only if MKP=YES is specified in the parameter data. This means that the fission production matrix is collected by position index. Calculating and printing cofactor k-effectives by position index can be avoided by specifying CKP=NO in the parameter data. An example of the cofactor k-effective by position index is shown in Fig. F11.6.35. See Sect. F11.D for a description of the problem used for the example.

SAMPLE PROBLEM 2 2C8 BARE WITH 8 UNIT TYPES MATRIX CALCULATION

POSITION INDEX	POSITION X	Y	Z	UNIT	COFACTOR K-EFFECTIVE	DEVIATION
1	1	1	1	1	9.322868E-01	8.896298E-03
2	2	1	1	2	9.617268E-01	8.805308E-03
3	1	2	1	3	9.576928E-01	9.093498E-03
4	2	2	1	4	9.602048E-01	9.027808E-03
5	1	1	2	5	9.009368E-01	8.583608E-03
6	2	1	2	6	9.542588E-01	8.690708E-03
7	1	2	2	7	9.547828E-01	9.003638E-03
8	2	2	2	8	9.560598E-01	9.107368E-03

Fig. F11.6.35. Example of cofactor k-effective by position index

The cofactor k-effective for a given position index is the largest eigenvalue of the fission production matrix collected by position index, reduced by the row and column associated with that position index. Thus the cofactor k-effective is the value of k-effective for the system calculated without the fission source of the unit located at the specified position index.

The POSITION INDEX is a number referencing a position in a three-dimensional lattice. POSITION is the x, y, and z location within the lattice. UNIT is the unit or box type located at the specified location in the lattice. Thus in Fig. F11.6.35, Unit 1 is located at the lower left-hand front corner of the array or three-dimensional lattice representing the problem (x=1, y=1, z=1) and the corresponding POSITION INDEX is 1. POSITION INDEX 8 is the top right-hand back corner of the lattice, POSITION x=2, y=2, z=2 and the unit located at that position is UNIT 8.

F11.6.32 MATRIX K-EFFECTIVE BY UNIT NUMBER

The matrix k-effective by unit number (unit k-effective) is the largest eigenvalue of the fission production by unit matrix. It is calculated only if MKU=YES is specified in the parameter data (Sect. F11.4.3). An example of the matrix k-effective by unit is given in Fig. F11.6.36.

SAMPLE PROBLEM 2 2C8 BARE WITH 8 UNIT TYPES MATRIX CALCULATION

****

UNIT K-EFFECTIVE= 9.932168E-01 +- OR - 8.346818E-03
THE UNIT K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY UNIT NUMBER MATRIX.

****

Fig. F11.6.36. Example of matrix k-effective by unit number

F11.6.33 FISSION PRODUCTION BY UNIT NUMBER MATRIX

These data are printed by subroutine MATRIX only if MKU=YES is specified in the parameter data, which results in the code calculating the fission production matrix by unit. Thus for each unit or box type in the array, the number of next generation neutrons produced in Unit J per neutron born in Unit I is determined. This is the fission production matrix by unit, and it is used to determine the matrix k-effective by unit, the cofactor k-effective by unit, and the source vector by unit. An example of the fission production matrix by unit is shown in Fig. F11.6.37.

```
FISSION PRODUCTION BY UNIT NUMBER MATRIX
( I, J) F IS THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN UNIT J BY A NEUTRON BORN IN UNIT I.
( 1, 1) 7.37E-01  ( 1, 2) 6.67E-02  ( 1, 3) 5.24E-02  ( 1, 4) 2.83E-02  ( 1, 5) 7.12E-03  ( 1, 6) 2.39E-02
( 1, 7) 2.26E-02  ( 1, 8) 1.76E-02
( 2, 1) 5.42E-02  ( 2, 2) 6.91E-01  ( 2, 3) 2.56E-02  ( 2, 4) 6.97E-02  ( 2, 5) 2.50E-02  ( 2, 6) 7.70E-02
( 2, 7) 1.70E-02  ( 2, 8) 2.49E-02
( 3, 1) 4.54E-02  ( 3, 2) 2.53E-02  ( 3, 3) 7.44E-01  ( 3, 4) 4.82E-02  ( 3, 5) 2.83E-02  ( 3, 6) 1.91E-02
( 3, 7) 6.67E-02  ( 3, 8) 2.26E-02
( 4, 1) 2.70E-02  ( 4, 2) 5.16E-02  ( 4, 3) 4.81E-02  ( 4, 4) 6.97E-01  ( 4, 5) 2.26E-02  ( 4, 6) 2.71E-02
( 4, 7) 2.94E-02  ( 4, 8) 7.21E-02
( 5, 1) 6.65E-02  ( 5, 2) 2.26E-02  ( 5, 3) 2.41E-02  ( 5, 4) 1.88E-02  ( 5, 5) 7.49E-01  ( 5, 6) 5.46E-02
( 5, 7) 5.12E-02  ( 5, 8) 2.48E-02
( 6, 1) 2.72E-02  ( 6, 2) 6.26E-02  ( 6, 3) 1.91E-02  ( 6, 4) 2.30E-02  ( 6, 5) 4.82E-02  ( 6, 6) 7.36E-01
( 6, 7) 2.73E-02  ( 6, 8) 5.18E-02
( 7, 1) 2.21E-02  ( 7, 2) 2.13E-02  ( 7, 3) 6.41E-02  ( 7, 4) 2.73E-02  ( 7, 5) 5.22E-02  ( 7, 6) 2.61E-02
( 7, 7) 7.25E-01  ( 7, 8) 5.37E-02
( 8, 1) 1.95E-02  ( 8, 2) 2.41E-02  ( 8, 3) 1.95E-02  ( 8, 4) 6.56E-02  ( 8, 5) 2.80E-02  ( 8, 6) 5.58E-02
( 8, 7) 5.62E-02  ( 8, 8) 7.32E-01
```

Fig. F11.6.37. An example of the fission probability matrix by unit

F11.6.34 SOURCE VECTOR BY UNIT NUMBER

These data are printed by subroutine MATRIX only if MKU=YES is specified in the parameter data. The source vector by unit is the eigenvector of the fission production matrix by unit. It represents the fission source for the units used in the problem. The components of the source vector should sum to 1.0. An example of the source vector by unit is given in Fig. F11.6.38. The average self-multiplication by unit is printed following the source vector. This value of self-multiplication includes those histories born in the unit which cause fissions in the same unit regardless of whether or not it exited and then returned. Therefore, this value will not agree with the value printed for the self-multiplication of the unit in Sect. F11.6.24 if the problem utilizes multiple units, the system is reflected, or a differential albedo is used in the problem.

```
SAMPLE PROBLEM 2  ICSBAB WITH 8 UNIT TYPES MATRIX CALCULATION

 SOURCE VECTOR BY UNIT

  UNIT      VECTOR
    1    1.27974E-01
    2    1.06535E-01
    3    1.27177E-01
    4    1.10871E-01
    5    1.39015E-01
    6    1.22649E-01
    7    1.28707E-01
    8    1.27072E-01

                AVERAGE SELF MULTIPLICATION BY UNIT

THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN A UNIT BY
A NEUTRON BORN IN THAT SAME UNIT IS   7.27785E-01 + OR -  2.91844E-03
```

Fig. F11.6.38. Example of the source vector by unit

F11.6.35 COFACTOR K-EFFECTIVE BY UNIT NUMBER

Cofactor k-effectives are printed by subroutine MATRIX only if MKU=YES is specified in the parameter data. Calculating and printing cofactor k-effectives by unit can be avoided by specifying CKU=NO in the parameter data. The cofactor k-effective for a given unit is the k-effective of the system calculated without the fission source of that unit. This is accompliished by determining the eigenvalue of the fission production matrix by unit after it has been reduced by the row and column associated with that unit. An example of the cofactor k-effective by unit is given in Fig. F11.6.39.

```
SAMPLE PROBLEM 2  2C8 BARE WITH 8 UNIT TYPES MATRIX CALCULATION

                                  COFACTOR
                      UNIT        K-EFFECTIVE        DEVIATION
                        1         8.325118E-01       8.867118E-03
                        2         9.617328E-01       8.801108E-03
                        3         9.576489E-01       9.092998E-03
                        4         9.602048E-01       9.027638-03
                        5         9.009708E-01       8.583458-03
                        6         9.542508E-01       8.690678-03
                        7         9.587828E-01       9.003628-03
                        8         9.560598E-01       9.107368-03

ELAPSED TIME    0.37983 MINUTES
```

Fig. F11.6.39. Example of cofactor k-effective by unit number

F11.6.36 MATRIX K-EFFECTIVE BY HOLE NUMBER

The matrix k-effective by hole number is calculated if MKH=YES was specified in the parameter data, Sect. F11.4.3. It is the largest eigenvalue of the fission production matrix collected by hole number. An example of the matrix k-effective by hole number is given in Fig. F11.6.40.

```
SAMPLE PROBLEM 10   1P27 DEMONSTRATION OF OPTIONS PROBLEM
*****************************************************************************************************************************

 HOLE K-EFFECTIVE=  1.027148+00  + OR -  2.795948-02
 THE HOLE K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY HOLE NUMBER MATRIX.
*****************************************************************************************************************************
```

Fig. F11.6.40. Example of matrix k-effective by hole number

F11.6.37 FISSION PRODUCTION BY HOLE NUMBER MATRIX

This is the fission production matrix collected by hole number. It is printed only if MKH=YES and FMH=YES were specified in the parameter data, Sect. F11.4.3. An example of this fission production matrix is given in Fig. F11.6.41. This matrix indicates the number of next generation neutrons produced in hole number J by a neutron born in hole number I.

```
FISSION PRODUCTION BY HOLE NUMBER MATRIX
  (  I,  J) P IS THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN HOLE J BY A NEUTRON BORN IN HOLE I.
  (  1,  1) 7.578E-01  (  1,  2) 6.555E-02  (  1,  3) 9.598E-02  (  1,  4) 1.258E-01
  (  2,  1) 1.438E-01  (  2,  2) 6.908E-01  (  2,  3) 9.178E-02  (  2,  4) 1.298E-01
  (  3,  1) 1.278E-01  (  3,  2) 5.268E-02  (  3,  3) 6.838E-01  (  3,  4) 1.348E-01
  (  4,  1) 1.118E-01  (  4,  2) 5.578E-02  (  4,  3) 7.498E-02  (  4,  4) 7.828E-01
```

Fig. F11.6.41. Example of fission production matrix by hole

F11.6.38 SOURCE VECTOR BY HOLE NUMBER

This information is printed by subroutine MATRIX only if MKH=YES is specified in the parameter data, Sect. F11.4.3. The source vector by hole is the eigenvalue of the fission production matrix by hole number. The source vector should sum to 1.0. An example of the source vector by hole is shown in Fig. F11.6.42. The average self-multiplication by hole is the overall average of the self-multiplication of all the holes in the problem.

```
SAMPLE PROBLEM 18   1F27 DEMONSTRATION OF OPTIONS PROBLEM

 SOURCE VECTOR BY HOLE

  HOLE      VECTOR
    1    3.121108E-01
    2    1.461108E-01
    3    2.001998E-01
    4    3.435748E-01

                AVERAGE SELF MULTIPLICATION BY HOLE

THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN A HOLE BY
A NEUTRON BORN IN THAT SAME HOLE IS   7.395668E-01  + OR -  1.571095E-02
```

Fig. F11.6.42. Example of source vector by hole number

F11.6.39 COFACTOR K-EFFECTIVE BY HOLE NUMBER

The cofactor k-effective by hole number is calculated if CKH=YES is entered in the parameter data, Sect. F11.4.3. The cofactor k-effective for a given hole is the k-effective of the system calculated without the fission source of that hole. This is done by determining the eigenvalue of the fission production matrix by hole after it has been reduced by the row and column associated with that hole. An example of the cofactor k-effective by hole number is given in Fig. F11.6.43.

```
SAMPLE PROBLEM 18   1F27 DEMONSTRATION OF OPTIONS PROBLEM

                                  COFACTOR
  HOLE          UNIT          K-EFFECTIVE          DEVIATION
    1             2          6.175638E-01         2.051048E-02
    2             3          9.657108E-01         2.923718E-02
    3             4          9.487698E-01         3.184528E-02
    4             5          7.200408E-01         4.323128E-02
```

Fig. F11.6.43. Example of cofactor k-effective by hole number

F11.6.40 MATRIX K-EFFECTIVE BY ARRAY NUMBER

The matrix k-effective by array number is calculated if MKA=YES is entered in the parameter data, Sect. F11.4.3. It is the largest eigenvalue of the fission production matrix collected by array number. An example is given in Fig. F11.6.44. The number of next generation neutrons produced in array number J by a neutron born in array number I is given in this fission production matrix.

```
SAMPLE PROBLEM 18   1F27 DEMONSTRATION OF OPTIONS PROBLEM
**********************************************************************************************************************

 ARRAY K-EFFECTIVE=  1.027168E+00  + OR -  2.795594E-02
 THE ARRAY K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY ARRAY NUMBER MATRIX.
**********************************************************************************************************************
```

Fig. F11.6.44. Example of matrix k-effective by array number

F11.6.41 FISSION PRODUCTION BY ARRAY NUMBER MATRIX

The fission production matrix collected by array number is shown in Fig. F11.6.45. It is printed only if MKA=YES and FMA=YES are specified in the parameter data, Sect. F11.4.3.

```
FISSION PRODUCTION BY ARRAY NUMBER MATRIX
( I, J) P IS THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN ARRAY J BY A NEUTRON BORN IN ARRAY I.
( 1, 1) 7.57E-01  ( 1, 2) 6.55E-02  ( 1, 3) 9.59E-02  ( 1, 4) 1.25E-01  ( 1, 5) 0.0
( 2, 1) 1.43E-01  ( 2, 2) 6.80E-01  ( 2, 3) 9.17E-02  ( 2, 4) 1.29E-01  ( 2, 5) 0.0
( 3, 1) 1.27E-01  ( 3, 2) 5.26E-02  ( 3, 3) 6.83E-01  ( 3, 4) 1.34E-01  ( 3, 5) 0.0
( 4, 1) 1.11E-01  ( 4, 2) 5.57E-02  ( 4, 3) 7.49E-02  ( 4, 4) 7.82E-01  ( 4, 5) 0.0
( 5, 1) 0.0       ( 5, 2) 0.0       ( 5, 3) 0.0       ( 5, 4) 0.0       ( 5, 5) 0.0
```

Fig. F11.6.45. An example of the fission production matrix by array number

F11.6.42 SOURCE VECTOR BY ARRAY NUMBER

This information is printed by subroutine MATRIX only if MKA=YES is specified in the parameter data, Sect. F11.4.3. The source vector by array number is the eigenvector of the fission production matrix by array number. The source vector should sum to 1.0. An example of the source vector by array number is shown in Fig. F11.6.46. The average self-multiplication by array number is the overall self-multiplication of all the arrays in the problem.

```
SAMPLE PROBLEM 18   1F27 DEMONSTRATION OF OPTIONS PROBLEM

 SOURCE VECTOR BY ARRAY

 ARRAY      VECTOR
   1     3.121110E-01
   2     1.661116E-01
   3     2.081998E-01
   4     3.435574E-01
   5     0.0

                    AVERAGE SELF MULTIPLICATION BY ARRAY

THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN  AN ARRAY BY
A NEUTRON BORN IN THAT SAME ARRAY IS   7.395668E-01  + OR -  1.360608E-02
```

Fig. F11.6.46. Example of source vector by array number

F11.6.43 COFACTOR K-EFFECTIVE BY ARRAY NUMBER

The cofactor k-effective by array number is calculated if CKA=YES is entered in the parameter data, Sect. F11.4.3. The cofactor k-effective for a given array is the k-effective of the system calculated without the fission source of that array. This is achieved by determining the eigenvector of the fission production matrix by array after reducing it by the row and column associated with the specified array. Figure F11.6.47 is an example of the cofactor k-effective by array number.

```
SAMPLE PROBLEM 18   1F27 DEMONSTRATION OF OPTIONS PROBLEM

   ARRAY        ARRAY       COFACTOR
   INDEX        NUMBER      K-EFFECTIVE      DEVIATION
     1            1        6.175638E-01    2.051094E-02
     2            2        9.657710E-01    2.923771E-02
     3            3        9.487698E-01    3.184532E-02
     4            4        7.200480E-01    4.323132E-02
     5            5        1.027158E-01    2.795748E-02
```

Fig. F11.6.47. Example of cofactor k-effective by array number

F11.6.44 FISSION DENSITY EDIT

The fission density edit is optional. Subroutine KEDIT prints the fission density for each geometry region if FDN=YES is specified in the parameter data. An example of the fission density edit is shown in Fig. F11.6.48.

SAMPLE PROBLEM 2 2C0 BARE WITH 8 UNIT TYPES MATRIX CALCULATION

UNIT	REGION	**** FISSION DENSITIES **** FISSION DENSITY	PERCENT DEVIATION	TOTAL FISSIONS
1	1	1.071E-04	2.04	1.196E-01
	2	0.0	0.0	0.0
2	1	9.404E-05	2.41	1.051E-01
	2	0.0	0.0	0.0
3	1	1.110E-04	2.23	1.240E-01
	2	0.0	0.0	0.0
4	1	9.288E-05	1.80	1.038E-01
	2	0.0	0.0	0.0
5	1	1.258E-04	2.76	1.406E-01
	2	0.0	0.0	0.0
6	1	1.258E-04	3.65	1.406E-01
	2	0.0	0.0	0.0
7	1	1.172E-04	2.15	1.309E-01
	2	0.0	0.0	0.0
8	1	1.170E-04	2.11	1.307E-01
	2	0.0	0.0	0.0

Fig. F11.6.48. Example of the fission density edit

The UNIT is the unit or box type number from the geometry data, the REGION is the region number relative to the unit, the FISSION DENSITY is the fissions/cc/source neutron for that geometry region, the PERCENT DEVIATION is the uncertainty associated with the fission density, and the TOTAL FISSIONS is the total number of fissions per source neutron in the geometry region.

F11.6.45 FLUX EDIT

Printing the fluxes is optional. They are printed by subroutine PRTFLX only if FLX=YES is specified in the parameter data. The fluxes are printed for each unit and each geometry region in the unit for every energy group. A sample of a flux edit is given in Fig. F11.6.49.

SAMPLE PROBLEM 18 1F27 DEMONSTRATION OF OPTIONS PROBLEM

FLUXES FOR UNIT 1

	REGION 1		REGION 2		REGION 3		REGION 4	
GROUP	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION
1	8.042E-06	2.02	6.307E-06	7.03	5.324E-06	2.66	3.208E-06	3.03
2	1.522E-05	1.64	1.091E-05	4.69	9.155E-06	1.78	5.868E-06	2.05
3	7.040E-06	1.99	5.279E-06	6.15	4.503E-06	2.47	2.736E-06	3.01
4	8.839E-06	1.51	7.196E-06	5.98	5.924E-06	2.29	3.681E-06	2.57
5	9.649E-06	1.14	6.714E-06	6.46	6.054E-06	2.57	3.964E-06	2.58
6	6.790E-06	1.47	4.788E-06	7.11	4.481E-06	2.62	3.021E-06	2.97
7	5.114E-06	1.47	3.945E-06	11.47	3.311E-06	2.84	2.295E-06	3.33
8	4.567E-06	1.28	3.379E-06	12.21	2.888E-06	2.33	2.175E-06	3.53
9	4.226E-06	1.47	2.836E-06	10.54	2.858E-06	3.44	2.117E-06	3.27
10	2.781E-06	1.56	2.297E-06	12.06	1.797E-06	3.56	1.442E-06	4.30
11	2.328E-06	1.91	1.743E-06	14.44	1.594E-06	4.33	1.331E-06	4.49
12	2.337E-06	1.52	1.399E-06	13.59	1.616E-06	3.15	1.281E-06	4.88
13	2.036E-06	1.88	1.269E-06	14.75	1.444E-06	3.85	1.240E-06	5.16
14	1.540E-06	1.82	1.445E-06	16.71	1.132E-06	3.84	9.511E-07	4.89
15	3.086E-06	1.30	2.492E-06	9.38	1.909E-06	2.63	2.707E-06	3.71
16	3.692E-06	1.16	1.234E-05	6.36	1.726E-05	1.98	2.378E-05	2.06

Fig. F11.6.49. An example of a flux edit

The title of the problem is printed at the top of the page. The heading FLUXES FOR UNIT _____ indicates the geometry unit for which fluxes are being printed. The region numbers relative to the unit are identified by the heading REGION _____. The geometry regions within each unit are numbered sequentially, beginning with one. GROUP is the heading for the energy groups. The headings FLUX and PERCENT DEVIATION are printed for each geometry region in the unit. The flux and its associated percent deviation are printed for every energy group and every geometry region. The flux is in units of neutrons/cm^2/source neutron.

F11.6.46 FREQUENCY DISTRIBUTIONS

Subroutine FREAK is responsible for printing the frequency distributions which are the last data printed for a problem. A frequency distribution consists of a bar graph indicating the number of generations having k-effective in a specified interval. The intervals are determined by the code, based on the upper and lower limits of the k-effectives calculated for the generations. One asterisk is printed for each generation k-effective. Four frequency distributions are printed as shown in Fig. F11.6.50.

```
SAMPLE PROBLEM 18   1F27 DEMONSTRATION OF OPTIONS PROBLEM

                              FREQUENCY FOR GENERATIONS    4 TO   52
0.8732 TO 0.8946     *
0.8946 TO 0.9160     *
0.9160 TO 0.9374
0.9374 TO 0.9587     ****
0.9587 TO 0.9801     ********
0.9801 TO 1.0015     *****
1.0015 TO 1.0229     *******
1.0229 TO 1.0443     **********
1.0443 TO 1.0657     ***
1.0657 TO 1.0870     ****
1.0870 TO 1.1084     *****
1.1084 TO 1.1298     *

                              FREQUENCY FOR GENERATIONS   16 TO   52
0.8732 TO 0.8946     *
0.8946 TO 0.9160     *
0.9160 TO 0.9374
0.9374 TO 0.9587     **
0.9587 TO 0.9801     *******
0.9801 TO 1.0015     ****
1.0015 TO 1.0229     ******
1.0229 TO 1.0443     ********
1.0443 TO 1.0657     **
1.0657 TO 1.0870     **
1.0870 TO 1.1084     ***
1.1084 TO 1.1298     *

                              FREQUENCY FOR GENERATIONS   28 TO   52
0.8732 TO 0.8946
0.8946 TO 0.9160     *
0.9160 TO 0.9374
0.9374 TO 0.9587     *
0.9587 TO 0.9801     *****
0.9801 TO 1.0015     ***
1.0015 TO 1.0229     ****
1.0229 TO 1.0443     ********
1.0443 TO 1.0657     *
1.0657 TO 1.0870
1.0870 TO 1.1084     *
1.1084 TO 1.1298     *

                              FREQUENCY FOR GENERATIONS   40 TO   52
0.8732 TO 0.8946
0.8946 TO 0.9160     *
0.9160 TO 0.9374
0.9374 TO 0.9587
0.9587 TO 0.9801     **
0.9801 TO 1.0015     ***
1.0015 TO 1.0229     **
1.0229 TO 1.0443     ***
1.0443 TO 1.0657     *
1.0657 TO 1.0870
1.0870 TO 1.1084     *
1.1084 TO 1.1298

CONGRATULATIONS!  YOU HAVE  SUCCESSFULLY TRAVERSED THE PERILOUS PATH THROUGH KENO V IN   0.38667 MINUTES
```

Fig. F11.6.50. Example of the frequency distributions

F11.7 WARNING MESSAGES AND ERROR MESSAGES

KENO V prints warning and error messages that are identified by K5- followed by a unique number, i.e., K5-1 is the identifier of the first message. For additional information concerning the message, simply look up the identifier number in this section.

Warning messages appear when a possible error is encountered. If the code alters data, that fact is stated in the message. It is the responsibility of the user to verify correct usage whenever a warning message is printed.

When an error is encountered, the error flag MFLAG is set true and an error message is printed. The code stops if the error is too severe to continue. The warning and error messages in this section may show an underscore _____ or a numbered underscore (1) where data will be printed by the code. The explanation of the message will show an underscore or a numbered underscore to indicate the corresponding data.

F11.7.1 MESSAGES

K5-1 ***** WARNING ***** READ FLAG NOT FOUND. ASSUME PARAMETER DATA FOLLOWS.

This message occurs in subroutine INITIAL. It indicates that the word READ was not the first word of data encountered after the title card. If a parameter data block is to be entered, the code expects the words READ PARAMETERS to precede the parameter input data. If the word READ is not the first word, the code expects parameter input data immediately.

K5-2 *** ERROR *** THE NUMBER OF ENERGY GROUPS IS OUT OF RANGE FOR THE CROSS SECTION LIBRARY ON UNIT _____ THE JOB CONTROL LANGUAGE MAY NOT SPECIFY A VALID DATA SET ON THIS UNIT OR THE MODULE THAT WAS TO CREATE THE CROSS SECTION LIBRARY ON THIS UNIT MAY HAVE FAILED.

This message occurs in subroutine INITIAL after subroutine PARAM has been executed. Check unit number _____ to see that it was properly specified in the job control language. Verify that the data set name associated with this unit number is the correct one. This information is given in the printout in the third table. Make sure the module that generated the cross sections executed properly and that the data were saved or passed correctly. When this message is printed for an AMPX working format library, a STOP 108 is executed. When this message is printed for a mixed cross-section format library, a STOP 109 is executed.

K5-3 *** WARNING *** THE CROSS SECTION LIBRARY ON UNIT _____ HAS _____ ENERGY GROUPS.

This message is printed in subroutine INITIAL after the call to PARAM. It is activated if the cross-section library has more than 227 energy groups. The largest standard cross-section library in the SCALE system contains 227 energy groups.

K5-4 INVALID INPUT PARAMETER NAME _____

This message comes from subroutine PARAM. The keyword for entering parameter data was misspelled. A list of allowed keywords is given in Table F11.4.1 in the KENO V input outline.

K5-5 ***** AN ERROR WAS ENCOUNTERED IN THE ALPHANUMERIC PARAMETER DATA. THE DATA WERE _____ _____ *****

This message comes from subroutine PARAM. The keyword for the alphanumeric parameter data was entered correctly, but the data associated with it were not YES or NO as is required. The _____ _____ in the error message could be something like FLX=YEX instead of FLX=YES.

K5-6 ***** WARNING ***** READ FLAG FOUND WHEN LOOKING FOR END FLAG. PARAMETER INPUT ASSUMED COMPLETE

This message occurs in subroutine PARAM. It indicates that the keywords END PARAMETERS were not found. The keywords READ _____ were found instead. The code assumes the parameter data are complete and proceed normally.

K5-7 ATTEMPT TO FIND END PARAMETER FLAG WAS UNSUCCESSFUL

This message from subroutine PARAM occurs during the reading of the parameter data if the word END is found and it is not followed by the word PARAMETERS. A STOP 118 may be executed when this message is printed.

K5-8 ***** AN END OF FILE WAS ENCOUNTERED WHILE ATTEMPTING TO READ PARAMETER DATA *****

This self-explanatory message is from subroutine PARAM. A STOP 118 may be executed when this message is printed.

K5-9 *** DUE TO INCONSISTENCIES BETWEEN INPUT AND RESTART DATA, FISSIONS AND ABSORPTIONS BY REGION WILL BE CALCULATED BUT NOT PRINTED. INPUT DATA SET FAR=NO, BUT DATA FROM THE RESTART UNIT SPECIFIED YES.

This message occurs in subroutine PARAM. It is mostly self-explanatory. The original problem (parent case) that wrote the restart data specified data inconsistent with the parameter data input to the restarted problem. The title of the parent case is given at the end of the parameter tables. The specification of the restart unit RST is given in the third table of the KENO V output.

K5-10 *** DUE TO INCONSISTENCIES BETWEEN INPUT AND RESTART DATA, FLUXES WILL BE CALCULATED BUT NOT PRINTED. INPUT DATA SET FLX=NO, BUT DATA FROM THE RESTART UNIT SPECIFIED YES.

This message occurs in subroutine PARAM. It is mostly self-explanatory. The original problem (parent case) that wrote the restart data specified data that did not agree with the parameter data input to the restarted problem. The title of the parent case is given at the end of the parameter tables. The specification of the restart unit RST is given in the third table of the KENO V output.

K5-12 INPUT PARAMETER NBK WAS ENTERED AS _____ IT WAS CHANGED TO _____ AT LEAST _____ POSITIONS ARE NECESSARY TO ACCOMMODATE THE NEUTRON BANK DATA.

This self-explanatory message is from subroutine PARAM. NBK should not be entered as input data unless it is known that the default value is inadequate.

K5-14 ***** ERROR - KEYWORD _____ _____ IS NOT A VALID MIXING TABLE KEYWORD. *****

This message is from subroutine MIXIT. It can only be encountered if a mixing table is expected (i.e., READ MIX or READ MIXT has been entered as data). At this point the only valid keywords are MIX=, EPS= or SCT=. The keyword that was entered is printed in the message. See Sect. F11.4.10 for assistance in setting up the mixing table data. A common error made by KENO IV users is to enter a negative nuclide ID number in the mixing table. The code will interpret this to be a keyword. A whole list of K5-14 error messages will be generated as the code reads through the mixing table in search of a valid keyword. The code considers any character string that does not begin with a number to be a keyword.

K5-15 MIXING TABLE TOO BIG

This message is from subroutine MIXIT. It indicates that additional core space is necessary to allow entry of the existing mixing table. A STOP 114 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-16 *** ERROR *** ERROR *** A VALUE MUST BE ENTERED FOR LIB IN THE PARAMETER INPUT SO CROSS SECTIONS CAN BE MIXED.

This message is from subroutine DATAIN. It occurs when a mixing table has been read but the unit number for the AMPXS working library is undefined. This is corrected by entering LIB=_____ in the parameter input data and making sure the desired AMPXS working library is properly defined as being on that unit in the job control language.

K5-17 UNRECOGNIZABLE GEOMETRY WORD _____

This message is from subroutine KENOG. In the process of reading the geometry data, the word _____ was encountered when a geometry word was expected. Several of these messages may be generated. A message is generated for each word of data that is read, until a valid geometry word is found. The data are out of phase or the geometry word is misspelled. Check the previous geometry card for a mixture ID, a bias ID, and the proper number of dimensions. See Sect. F11.4.4 for a list of accepted geometry words.

K5-18 ********** ERROR.....NHCYL=_____ **********

This message from subroutine KENOG indicates that the hemicylinder geometry word was incorrectly specified. See Sect. F11.4.4 for the correct hemicylinder specification.

K5-19 AN ERROR WAS FOUND IN THE HEMISPHERE DESIGNATION

This message from subroutine KENOG indicates that the direction in which the hemisphere exists was incorrectly specified. See Sect. F11.4.4 to determine the correct specification.

K5-20 ********** IGEO=_____ INVALID IN READGM **********

This message from subroutine READGM means that IGEO is negative or larger than 23 when the geometry data are read from the scratch unit, SKRT. This usually means that a code error was introduced when changes were made to the program. True geometry errors should be detected when the scratch unit, SKRT, is written. A STOP 125 is executed when this message is printed.

K5-21 _____ IS AN INVALID PARAMETER NAME FOR BIASING DATA. ID= OR WT= OR WTS= SHOULD HAVE BEEN ENTERED.

This self-explanatory message is from subroutine RDBIAS. See Sect. F11.4.7 for assistance in determining the proper procedure for entering biasing data.

K5-22 FIRST TWO NUMBERS ARE WORDS OF STORAGE NEEDED AND ALLOCATED. THIRD IS REQUIRED ADDITIONAL REGION SIZE.
PERTINENT CONSTANTS
_____ _____ _____

This message from subroutine WATES indicates that the allocated computer storage will not hold the weighting or biasing array. The first number printed is the amount of storage, in words, needed to hold the data. The second number is the allocated computer storage in words. The third number is the minimum additional region size, in units of K bytes necessary to hold the biasing or weighting data to this point. Increase the region size for the "go step" in the job control language by the additional required region size (the third number) and resubmit the problem. A STOP 150 is executed when this message is printed.

K5-23 *** ERROR *** ERROR *** NO _____ ENERGY GROUP WEIGHTS WERE FOUND FOR ID _____ *** ERROR *** ERROR ***

This message from subroutine WATES occurs if the weights requested in the biasing information were not on the standard weights data set and were not entered from cards. See Table F11.4.5 for the weights that are available on the standard data set. The procedure for entering weights from cards is explained in Sect. F11.4.7.

K5-24 INCORRECT FLAG RETURNED FROM AREAD. IRET=_____

This message from subroutine RDBIAS indicates that an error was encountered while reading the biasing data. The biasing data were not entered properly. See Sect. F11.4.7 for assistance.

K5-25 THE FIRST NUMBER IS THE AMOUNT OF STORAGE NEEDED; THE SECOND IS THE AMOUNT ALLOCATED. PERTINENT CONSTANTS _____ _____

This message is from subroutine ARAYIN or subroutine GEOMIN. It indicates that additional core space is required to allow the use of the array definition data or the geometry region data. At least _____ words of storage are needed to run the problem but only _____ words of storage are available. Increase the amount of computer storage requested in the job control language to correspond to the amount needed. A STOP 100 or 155 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-26 SET NUMBER _____ OF THE UNIT ORIENTATION DATA CONTAINS _____ ERROR(S)

This message from subroutine RDBOX is triggered when input errors are recognized in the unit orientation data. A set of unit orientation data consists of 10 numbers as shown in the companion message K5-27. The number of errors printed in this message is a lower bound. More errors may actually exist. This message often means that a number was omitted or a blank was omitted when entering the unit orientation data.

K5-27 LTYPE=_____ IX1=_____ IX2=_____ INCX=_____ IY1=_____
IY2=_____ INCY=_____ IZ1=_____ IZ2=_____ INCZ=_____

This message is a companion message for K5-26. It indicates how the unit orientation data description for the set named in K5-26 was entered. See Sect. F11.4.5 for information pertaining to unit orientation data.

K5-28 THE ABOVE UNIT ORIENTATION CARD(S) CONTAIN(S) AT LEAST ONE OF THE FOLLOWING ERRORS

1. IX1,IY1,IZ1,INCX,INCY, OR INCZ IS LESS THAN OR EQUAL TO ZERO

2. IX2 IS LESS THAN IX1, IY2 IS LESS THAN IY1, OR IZ2 IS LESS THAN IZ1

3. IX2 IS GREATER THAN NBXMAX, IY2 IS LARGER THAN NBYMAX OR IZ2 IS LARGER THAN NBZMAX

4. LTYPE IS LESS THAN 1 OR GREATER THAN NBOX

This self-explanatory message is from subroutine RDBOX. It pertains to the input orientation data for LOOP. See Sect. F11.4.5 for input instructions.

K5-29 *** ERROR *** THE ARRAY SIZE HAS BEEN SPECIFIED INCORRECTLY FOR ARRAY_____
NBXMAX=_____ NBYMAX=_____ NBZMAX=_____
UNIT ORIENTATION DATA CANNOT BE READ UNLESS NBXMAX, NBYMAX, AND NBZMAX ARE GREATER THAN ZERO.

This message from subroutine ARAYIN indicates that the array definition data were incorrectly specified. It occurs only if one or more of NBXMAX, NBYMAX, or NBZMAX is less than 1. In the array information data these are entered in the form NUX=_____ NUY=_____ NUZ=_____ See Sect. F11.4.5. If a unit orientation data description is to be entered, NBXMAX, NBYMAX, and NBZMAX must all be greater than zero.

K5-30 END _____ FLAG WAS NOT FOUND. _____ _____ WAS READ INSTEAD.

This message from subroutine ARAYIN occurs if the unit orientation data description is terminated with the incorrect END flag.

K5-31 _____ IS AN INVALID PARAMETER NAME IN THE ARRAY DATA.

This message is written from subroutine ARAYIN if the array data block contains an incorrect keyword. The allowed keywords include NUX= NUY= NUZ= FILL and LOOP. See Sect. F11.4.5 for additional assistance. A STOP 101 is executed when this message is printed.

K5-32 *** AN ERROR EXISTS IN UNIT ORIENTATION ARRAY NUMBER _____ ***

This message from subroutine SORTA is printed when an error is recognized in the array description. The type of error that will trigger the message is for a position in the unit orientation array to be undefined, zero, negative or greater than NBOX, the number of input units. K5-33 is a companion message.

K5-33 UNIT _____ IS INVALID AT X INDEX= _____ Y INDEX= _____ Z INDEX= _____

This message comes from subroutine SORTA. It is printed for each position in the unit orientation array that is in error. The message is printed a maximum of 10 times. Refer to Sect. F11.4.5 for assistance in correcting the error(s).

K5-34 ***** ERROR ***** THE NUMBER OF MIXTURES REQUESTED IN THE GEOMETRY IS _____ THE NUMBER OF MIXTURE CROSS SECTIONS IS _____

This message from subroutine FLDATA occurs if the number of mixture cross sections from the restart unit, RSTRT, does not equal the number of mixtures requested in the geometry for a restarted problem.

K5-35 ***** ERROR ***** IN THE ALBEDO INPUT DATA _____ IS AN INVALID FACE CODE NAME.

This message is from subroutine RDREF. It occurs if an invalid face code name was entered in the albedo data. See Table F11.4.3 in Sect. F11.4.6 for a list of acceptable face code names.

K5-36 A PERIODIC BOUNDARY CONDITION WAS SPECIFIED WITH A NON-COMPATIBLE BOUNDARY CONDITION ON THE OPPOSING FACE. THE PROBLEM WILL NOT BE RUN.

This self-explanatory message is from subroutine RDREF. If a periodic boundary condition is specified on one x face, it must also be specified on the other x face, etc.

K5-37 ***** ERROR AVERAGE NU-BAR AND AVG. FISSION GROUP WAS SPECIFIED, BUT THE FISSION XSEC ID (18) WAS NOT FOUND IN THE EXTRA 1-D ARRAY (MT).

This message is from subroutine IDX1D. It indicates that the parameter data contained NUB=YES but the corresponding necessary type of data was absent from the extra 1-D array. This can be due to a code error or an error concerning the extra 1-D data (X1D= in the parameter data).

K5-38 INPUT DATA INDICATED NO EXTRA 1-D XSEC IDS TO BE READ, BUT A READ FLAG WAS ENCOUNTERED.

This message from subroutine DATAIN is printed when the parameter data did not specify X1D= and the words READ X1DS were encountered later in the data. If extra 1-D data are to be used, X1D= must be entered in the parameter data and appropriate code modifications must be made to properly utilize the extra 1-D data.

K5-39 INVALID START PARAMETER NAME _____

This message is from subroutine RDSTRT. It indicates that an invalid start parameter name was encountered when the start data block was being read. A list of allowed start parameter names is contained in Sect. F11.4.8.

K5-40 LNU FOR START TYPE 6 WAS ENTERED AS _____ IT WAS CHANGED TO _____ WHICH IS THE VALUE ENTERED FOR NFB.
THE LARGEST VALUE NEEDED FOR LNU IS NPG. THE LARGEST VALUE ALLOWED FOR LNU IS NFB. BOTH NPG AND NFB ARE PARAMETER DATA.

This self-explanatory message is from subroutine RDSTRT. See Sect. F11.4.8 for assistance in determining a valid value for LNU. NFB, the fission bank size, is the largest value allowed for LNU. NPG, the number of histories per generation, is the smallest value allowed for LNU.

K5-41 *** ERROR *** ALPHANUMERIC START DATA MUST BE ENTERED AS YES OR NO. THE DATA READ WERE _____ _____

This self-explanatory message is from subroutine RDSTRT. See Sect. F11.4.8 for assistance concerning start data.

K5-42 END _____ FLAG WAS NOT FOUND. _____ _____ WAS READ INSTEAD.

This message is from subroutine DATAIN. It occurs when the READ _____ and END _____ do not match. When entering data blocks, each block must start with READ _____ and end with END _____

K5-43 ***** AN END OF FILE WAS ENCOUNTERED BEFORE AN END DATA WAS FOUND. THE PROBLEM WILL NOT RUN. *****

This message is from subroutine DATAIN, ARAYIN, KENOG, or RDPLOT. It occurs when an end of file is encountered while reading data.

K5-45 ***** ILLEGAL DATA BLOCK IDENTIFIER _____ _____ *****

This message from subroutine DATAIN is printed whenever an invalid data block identifier is encountered. This can be caused by having the data out of order, by omitting data or by misspelling data. A block identifier consists of the words READ XXXX where XXXX is a keyword identifying the type of data to be read. Acceptable keywords are listed in Table F11.4.1, in Sect. F11.4.1.

Consider the following examples:

ERROR MESSAGE EXAMPLE 1

```
READ PARAM TME=2.9 FLX=YES XSC=38 END PARAM
READ GOEM   CYLINDER   1 1 5.0 5.0 -5.0 END GEOM
END DATA
END
```

The following message would occur:

***** ILLEGAL DATA BLOCK IDENTIFIER READ GOEM *****

The keyword GEOM was misspelled as GOEM. The correct data block identifier is READ GEOM.

ERROR MESSAGE EXAMPLE 2

```
READ PARAM   TME=2.9 FLX=YES XSC=38 END PARAM
END GEOM
CYLINDER   1 1 5.0 5.0 -5.0   END GEOM
END DATA
END
```

The following message would occur:

***** ILLEGAL DATA BLOCK CYLINDER *****

The words END GEOM are read and recognized as the end of a data block. The words CYLI and NDER are then read as the data block identifier. Since CYLI is not END, the code assumes it is at the beginning of a data block that is identified by the keyword NDER, which is not one of the acceptable keywords.

ERROR MESSAGE EXAMPLE 3

```
READ PARAM   TME=2.9 FLX=YES XSC=38 END PARAM
END GEOM
READ GEOM
CYLINDER   1 1 5.0 5.0 -5.0
END GEOM
END DATA
END
```

In this example, no errors will be found and the problem will run correctly. The END PARAM signals the end of the parameter data block. The first END GEOM signals the end of a geometry block, the READ GEOM signals the beginning of a geometry block, the second END GEOM signals the end of a geometry block, and the END DATA signals the end of the problem.

K5-46 ***** IPT= _____ IS OUTSIDE THE ALLOWABLE LIMIT OF _____

This self-explanatory message is from subroutine DATAIN. It is indicative of a code error. IPT is the index into the LPOINT array, which contains the direct access pointers for the various types of data.

LPOINT(1) is the pointer for the geometry region data.
LPOINT(2) is the pointer for the array description (unit orientation) data.
LPOINT(3) is the pointer for the mixing table data.
LPOINT(4) is the pointer for extra data.
LPOINT(5) is the pointer for the biasing or weighting data.
LPOINT(6) is the pointer for the start data.
LPOINT(7) is the pointer for the albedo data.
LPOINT(8) is the pointer for the mixed cross-section data.
LPOINT(9) is the pointer for the energy and inverse velocity data.
LPOINT(10) is the pointer for the plot data.
LPOINT(11) is the pointer for the biasing input data.
LPOINT(12) is the pointer for albedo-xsec energy correspondence.

K5-47 *** ERROR *** AN ATTEMPT WAS MADE TO READ A NONEXISTENT TYPE OF DATA FROM THE RESTART UNIT. NDX=_____ NREC=_____

This message is from subroutine RDRST. It indicates that the restart file has been destroyed or is undefined.

K5-48 *** ERROR *** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO READ RESTART DATA FROM UNIT _____ NDX=_____ NREC=_____

This message is from subroutine RDRST. It indicates that the restart data associated with the index NDX had the wrong number of records. This message can also be caused by a code error introduced as the result of making changes in the code.

NDX=1 is the geometry data. NREC should be 3.
NDX=2 is the array description data. NREC should be 3 or more.
NDX=3 is the mixing table data. NREC should be 3.
NDX=4 is the extra data. NREC must be set by the user.
NDX=5 is the biasing or weight data. NREC should be 3.
NDX=6 is the start data. NREC should be 3.
NDX=7 is the albedo data. NREC should be at least 2.
NDX=8 is the mixed cross-section data.
NDX=9 is the energy and inverse velocity data.
NDX=10 is the plot data.
NDX=11 is the biasing input data.
NDX=12 is the albedo-xsec energy correspondence.
A STOP 123 is executed when this message is printed.

K5-49 _____ WORDS ARE NEEDED TO HOLD THE ALBEDO DATA. BUT ONLY _____ WORDS ARE AVAILABLE.

This message from subroutine RDALB is self-explanatory. More space is needed to contain the albedo data. A STOP 120 is executed when this message is printed.

K5-50 THE SPACE AVAILABLE IN SUBROUTINE RDICE IS _____ MORE IS NEEDED.

This self-explanatory message from subroutine RDICE indicates that more space is needed to store the macroscopic cross sections in ICE format. A STOP 122 is executed when this message is printed.

K5-51 ***** ERROR ***** THE _____ FACE REQUESTED A DIFFERENTIAL ALBEDO THAT IS NOT ON THE ALBEDO DATA SET. THE REQUESTED ALBEDO NAME IS _____ _____

This self-explanatory message is from subroutine ALBRD. A list of the albedos that are on the standard albedo data set is given in Table F11.4.4 in Sect. F11.4.6.

K5-52 MIXTURE NUMBER TOO LARGE

This self-explanatory message is from subroutine MIXCRS. A STOP 111 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-53 NOT ENOUGH STORAGE TO MIX

This message from subroutine MIXER indicates that more storage is necessary in order to do the cross-section mixing operations. A STOP 112 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-54 NOT ENOUGH STORAGE TO MAKE ANGLES

This message from subroutine MIXER indicates that more storage is necessary to perform the calculations required to make the angles during the cross-section mixing operations. A STOP 113 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-55 **** ERROR FOUND IN MIXING CROSS SECTIONS ****
THE FOLLOWING NUCLIDE(S) SPECIFIED IN THE MIXING TABLE WERE NOT FOUND ON THE CROSS SECTION LIBRARY.

ENTRY	NUCLIDE ID
.	.
.	.
.	.

This self-explanatory message is from subroutine MIXMIX. Either a nuclide ID was misspelled or was not in the cross-section library.

K5-56 ERROR - MIXTURE _____ LACKS EITHER NU*FISSION OR CHI DATA.

This message from subroutine NORM1D indicates that a mixture that contains fissile material is missing the nu-fission cross section or the fission spectrum. A STOP 115 is executed when this message is printed.

K5-57 NOT ENOUGH STORAGE TO MAKE ANGLES AND PROBS

This self-explanatory message from subroutine MAKANG indicates that more storage is needed to complete the cross-section mixing operations. Resubmit the problem and request more storage space in the job control language. A STOP 110 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-58 This message appears in two forms, listed below:

K5-58 ** ERROR **** ERROR ** INVALID BIAS ID _____ IN REGION _____ OF UNIT _____

This form of message K5-58 is from subroutine READGM. It is printed if a negative or zero bias ID is encountered anywhere in the geometry data. The error flag (MFLAG) is not activated so the problem will execute if the unit containing the error is not utilized in the unit orientation array. If that unit is used in the unit orientation array, the following form of message K5-58 will also be printed.

K5-58 ** ERROR **** ERROR ** INVALID BIAS ID _____ IN REGION _____ ABOVE.

This form of message K5-58 is from subroutine PRTJOM. It is printed if a negative or zero bias ID is entered for the specified geometry region. The problem will not execute if this form of message K5-58 is printed. Review Sect. F11.4.4 for correct geometry data specification information.

K5-59 THE CALCULATION WAS TERMINATED BECAUSE OF EXCESSIVE SPLITTING

This message from subroutine GUIDE is printed only if message K5-128 is printed 50 or more times for a given generation. This indicates that the problem and/or the code is incapable of achieving a reasonable solution. If changes have been made in the code, they should be carefully scrutinized. If a biasing data block has been entered (Sect. F11.4.7), it should be checked carefully.

K5-60 THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE _____ HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP _____ TO GROUP _____

THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
(P_0) (P_1) (P_2) ... (P_n)
THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
(M_1) (M_2) ... (M_n)
THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
(M_1) (M_2) ... (M_n)
THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
(P_0) (P_1) (P_2) ... (P_n)

_____ MOMENTS WERE ACCEPTED

This message from subroutine BADMOM indicates that the moments from the cross-section data are incorrect for the group transfer shown. The code replaces these moments with acceptable moments and proceeds normally. The user can suppress these messages by entering an appropriate value for the "cross-section message cutoff parameter," EPS= in the mixing table data, Sect. F11.4.10. See Sect. F11.5.4.4 for assistance in determining an appropriate value.

K5-61 MKP= MKU= MKH= AND/OR MKA= WAS SPECIFIED AS YES IN THE PARAMETER DATA, BUT IS NOT ALLOWED FOR A SINGLE UNIT PROBLEM. NO MATRIX CALCULATIONS WILL BE MADE FOR THIS PROBLEM.

This message from subroutine DATAIN is self-explanatory. If matrix information is desired, the problem should be converted to a 1x1x1 array by entering the appropriate ARRAY DEFINITION DATA and assuring that the last geometry region is a cube or cuboid.

K5-62 POSIT ERROR IN TRACK --- ILLEGAL GEOMETRY TYPE X=_____ Y=_____ Z=_____ K1=_____ K2=_____ K=_____ IGEO=_____

This message from subroutine TRACK is usually the result of a code error that was introduced when changes were made to the code. A STOP 134 is printed in conjunction with this message.

K5-63 ***** WARNING ***** PARAMETER INPUT DATA FOR NBK=AND/OR XNB= WAS ENTERED FOR A RESTART PROBLEM HAVING BEG= GREATER THAN 1.
NBK= AND XNB= HAVE BEEN RESET TO THE VALUES FROM THE RESTART UNIT.

This self-explanatory message is from subroutine PARAM. The configuration of the neutron bank cannot be changed for a restarted problem unless it restarts at the first generation.

K5-65 AN AMPX WORKING LIBRARY WAS SPECIFIED ON UNIT _____ BUT NO MIXING DATA WAS READ.

This message from subroutine ICEMIX occurs if the parameter data specified LIB=_____ but no cross-section mixing data block was entered. The cross-section mixing data block begins with READ MIXT. See Sect. F11.4.3 for parameter data and Sect. F11.4.10 for mixing table information.

K5-66 _____ MIXTURES WERE REQUESTED IN THE GEOMETRY, BUT ONLY _____ MIXTURES ARE ON THE MIXED CROSS SECTION LIBRARY.

This message from subroutine ICEMIX indicates that more mixtures were requested in the geometry region data than were available on the mixed cross-section library. See Sect. F11.4.3 for the specification of the unit number of the mixed cross-section library (XSC=), Sect. F11.4.10 for the specification of the mixing table, and Sect. F11.4.4 to determine the mixtures used in the geometry region data.

K5-67 THE ADJOINT INPUT PARAMETER WAS _____ BUT THE ADJOINT INDICATOR FROM THE MIXTURE CROSS SECTION LIBRARY WAS _____ KENO WILL NOT EXECUTE.

This message from subroutine ICEMIX occurs if the adjoint input parameter ADJ= specified a forward calculation and the cross sections were adjointed or the adjoint input parameter specified an adjoint calculation and the cross sections were not adjointed. T indicates true, F indicates false.

K5-68 THE MIXED CROSS SECTION LIBRARY ON UNIT _____ IS COMPLETELY COUPLED. KENO WILL NOT BE EXECUTED.

This message is from subroutine ICEMIX. It indicates that the mixed cross-section library is a completely coupled neutron-gamma library and therefore cannot be used for a KENO calculation. A STOP 106 is executed when this message is printed.

K5-69 THE AMOUNT OF REMAINING SPACE IS INSUFFICIENT TO CONTAIN THE NECESSARY DATA. LIMIT=_____

This message from subroutine RDTAPE indicates that more storage is needed to allow processing of the premixed ice format cross-section data. A STOP 124 is executed when this message is printed.

K5-70 ERROR IN SUBROUTINE WRTRST. NDX=_____

This message from subroutine WRTRST occurs only if the type of data to be written on the restart (WSTRT) unit is undefined (i.e., NDX is less than 1 or greater than 12). NDX is the index in the LPOINT array as described in messages K5-46 and K5-48. This error is usually caused by code errors that were introduced when changes were made to the code. A STOP 133 is executed when this message is printed.

K5-71 INSUFFICIENT SPACE ALLOWED IN SUBROUTINE ALBWRT. _____ WORDS ARE ALLOWED, BUT _____ WORDS WERE REQUESTED.

This message from subroutine WRTALB indicates that more core storage is needed to allow loading the albedo data from the direct access device into core. A STOP 105 is executed when this message is printed. Resubmit the problem, requesting more storage space in the job control language.

K5-72 ARRAY IS TOO LARGE TO FIT IN SPACE ALLOTTED IN SUBROUTINE WRTICE.

This message is from subroutine WRTICE. It indicates that more storage is needed to allow loading the mixed cross-section data from the direct access device into core. A STOP 132 is executed when this message is printed. Resubmit the problem, requesting more storage space in the job control language.

K5-73 _____ MIXTURES WERE REQUESTED IN THE GEOMETRY DATA, BUT ONLY _____ OF THESE WERE FOUND IN THE MIXED CROSS SECTIONS.

This self-explanatory message is from subroutine MASTER. Either the wrong mixed cross-section data set (XSC= from the parameter data, Sect. F11.4.3) is being used, or one or more mixture numbers are in error in the geometry region data (see Sect. F11.4.4).

K5-74 ***** _____ WORDS OF STORAGE WERE ALLOCATED TO RUN THIS PROBLEM, BUT _____ WORDS ARE REQUESTED FOR THE INPUT DATA. *****

This self-explanatory message is from subroutine MASTER. Additional storage is required to run this problem. Alter the job control language to request sufficient storage.

K5-75 *** NOT ENOUGH STORAGE TO PRINT ONE DIMENSIONAL CROSS SECTIONS. _____ LOCATIONS ARE NEEDED, BUT ONLY _____ ARE AVAILABLE

This self-explanatory message is from subroutine PRTXS. Additional storage must be requested in the job control language in order to print the 1-D mixture cross sections.

K5-76 *** NOT ENOUGH STORAGE TO PRINT TWO DIMENSIONAL CROSS SECTIONS. _____ ARE NEEDED, BUT ONLY _____ ARE AVAILABLE.

This self-explanatory message is from subroutine PRTXS. Additional storage is needed to print the 2-D mixture cross sections. More storage must be requested in the job control language.

K5-77 *** NOT ENOUGH STORAGE TO PRINT ANGLES AND PROBABILITIES. _____ LOCATIONS ARE NEEDED, BUT ONLY _____ ARE AVAILABLE.

This self-explanatory message is from subroutine PRTXS. Additional storage must be requested in the job control language to print the angles and probabilities for the mixture cross sections.

K5-78 ***** WARNING ***** MORE SPACE MAY BE NEEDED THAN IS INDICATED.

This message is from subroutine NSUPG. In the process of supergrouping the energy-dependent information, the code ran out of available storage space. An attempt has been made to estimate the amount of storage needed, but the estimate may be low. Resubmit the job and request sufficient storage in the job control language.

K5-79 THIS PROBLEM IS TOO LARGE FOR THE AMOUNT OF CORE THAT WAS ALLOCATED.

This message is from subroutine NSUPG. It occurs during the process of determining the number of supergroups, if the amount of storage is found to be insufficient to contain the energy-dependent information associated with one of the energy groups. Change the job control language to allow more storage. A STOP 116 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-80 _____ WORDS OF STORAGE WERE ALLOCATED, BUT AT LEAST _____ ADDITIONAL WORDS ARE NEEDED TO HOLD THE INPUT DATA. EVEN MORE SPACE WILL BE NECESSARY TO RUN THE PROBLEM.

This message from subroutine NSUPG is printed if the available storage is too small to hold the input data. Change the job control language to allow more storage.

K5-81 THE AVAILABLE SPACE IS TOO SMALL TO CONTAIN THE DATA OF THE LARGEST ENERGY GROUP. MAXL=_____ LFTLNG=_____ LTOT=_____ LTOTAL=_____ NSG=_____

This message is from subroutine NSUPG.

MAXL is the size of the energy-dependent data associated with the largest energy group.

LFTLNG is the amount of space available to the super group.

LTOT is the total length of the cross-section data.

LTOTAL is the total length of the albedo data.

NSG is the supergroup being processed.

A STOP 117 is executed when this message is printed. Increase the allocated storage in the "go step" of the job control language and resubmit the problem.

K5-82 TOO MANY STORAGE LOCATIONS REQUIRED. _____ WORDS OF STORAGE ARE NEEDED, BUT ONLY _____ ARE AVAILABLE.

This self-explanatory message is from subroutine POINT. To run the problem, the job control language must be changed to increase the amount of storage to be consistent with that specified in the message.

K5-83 ***** FOR ARRAY _____ THE _____ DIMENSIONS OF UNIT _____ AT (_____, _____,_____) DO NOT MATCH THOSE OF UNIT _____ AT (_____,_____,_____) *****
FOR UNIT _____+_____=_____ AND -_____=_____ WHILE FOR UNIT _____+_____=_____ AND -_____=_____

This message is from subroutine ARASIZ. The common faces of adjacent units or box types must be the same size and shape. This message occurs whenever this requirement is not met. One or more of the dimensions of the units or box types specified in the message may be incorrect, or the array definition data may be incorrect. Carefully check the input data relating to the geometry region data and the array definition data as described in Sects. F11.4.4 and F11.4.5.

K5-84 ********** UNIT _____ IN UNIT ORIENTATION ARRAY NUMBER _____ IS UNDEFINED IN THE INPUT DATA **********

This message from subroutine ARASIZ occurs when the array description data block specifies a unit or box type that was not defined in the geometry region data. Verify the array definition data and the geometry region data as described in Sects. F11.4.4 and F11.4.5.

K5-85 UNIT _____ IS INVALID AT POSITION X=_____ Y=_____ Z=_____ IN UNIT ORIENTATION ARRAY NUMBER _____

This message from subroutine ARASIZ occurs if the unit number named in the message is less than or equal to zero or greater than NBOX (the number of different box types). The position of the offending unit is also given. This error usually results from leaving some positions undefined in the unit orientation array or from erroneous data in the unit orientation data. (This includes extra data, mispunched data and omitted data.) See Sect. F11.4.5 for additional information.

K5-86 UNRECOGNIZED GEOMETRY TYPE. IGEOM=_____ IN SUBROUTINE PRTJOM.

This message from subroutine PRTJOM is self-explanatory. The printed value of IGEOM must be greater than zero and less than 24 to be valid. If the geometry words (see *fgeom* in Sect. F11.4.4) are correct, this message is due to a code error that has been introduced when changes were made to the code.

K5-87 ERROR IN HEMISPHERE DESIGNATION. ISET=_____ IN SUBROUTINE PRTJOM.

This message from subroutine PRTJOM is self-explanatory. ISET should be greater than zero and less than 7. It defines the orientation of a hemisphere. A code error is the likely cause of this message.

K5-88 ERROR IN HEMICYLINDER DESIGNATION. NHCYL=_____ IN SUBROUTINE PRTJOM

This message from subroutine PRTJOM can occur if the hemicylinder designation is incorrect or if a code error has been introduced. See Sect. F11.4.4 for correct hemicylinder specification information.

K5-89 NBOX EXCEEDS 10,000. EXECUTION IS TERMINATED.

The message is from subroutine PRTLBA. The limit of 10000 for NBOX (the number of different units or box types) is arbitrary. Such a large number of units or box types would use up a lot of storage and be an exceedingly large problem. The number of lines of print for the unit orientation array would be very large. A STOP 119 is executed when this message is printed. Therefore execution is terminated.

This message from subroutine PRTLBA occurs if NBXMAX, NBYMAX or NBZMAX (NUX=, NUY=, NUZ=) from the array data, Sect. F11.4.5 was incorrectly specified.

K5-91 ********** UNIT _____ CONTAINS THE FOLLOWING GEOMETRY INCONSISTENCIES. **********

This message from subroutines JOMCHK, HOLEXT or HOLHOL indicates that one or more intersecting geometry regions were encountered in unit or box type _____. K5-92 is the companion message from subroutine JOMCHK and specifies the regions that intersect. K5-166 is the companion message from subroutine HOLEXT. K5-169 and K5-179 are the companion messages from subroutine HOLHOL. Correct the geometry region data and resubmit the problem. Section F11.4.4 may provide assistance in correctly specifying the data.

K5-92 REGION NUMBER _____ INTERSECTS REGION NUMBER _____

This message from subroutine JOMCHK is a companion message to K5-91 and K5-94. It specifies the intersecting regions. The user must determine which region is incorrectly specified or if the data are out of order. KENO REQUIRES THAT EACH SUCCESSIVE GEOMETRY REGION MUST COMPLETELY ENCLOSE THE PREVIOUS REGION. THIS DOES ALLOW COMMON FACES AND TANGENCY.

K5-93 INVALID GEOMETRY TYPE. IGEO=_____

This message is from subroutine JOMCHK. IGEO must be greater than zero and less than 20. If it does not fall in this range, a code error is the probable cause. KENOG or READGM are the subroutines likely to have the code error. Verify that all the geometry words (*fgeom*, Sect. F11.4.4) are correct.

K5-94 ***** THE REFLECTOR DIMENSIONS ARE INCONSISTENT *****

This message is from subroutine JOMCHK. It is a companion to message K5-95 and is printed whenever one or more intersecting geometry regions are encountered in the external reflector.

K5-95 REGION NUMBER _____ IN UNIT NUMBER _____ CONTAINS AN ERROR IN THE DIMENSIONS.

This message from subroutine VOLUME indicates an error in the geometry input data such that the negative dimension specification for a cube or cuboid is larger than the positive dimension specification (i.e., the -x dimension is greater than the +x dimension, or the -y dimension is greater than the +y dimension or the -z dimension is greater than the +z dimension). This message is also printed if the magnitude of the chord for a hemicylinder or hemisphere is larger than the radius. See Sect. F11.4.4 for assistance in specifying the geometry correctly.

K5-96 THE VOLUME DEFINED BY GEOMETRY CARD _____ IS NEGATIVE.

This message from subroutine VOLUME is printed whenever a negative volume is calculated. This can be caused by the positive dimension being smaller than the negative dimension on a face of a geometry region. It can also be caused by having intersecting regions, or be the result of roundoff when the volumes are calculated. Either the geometry regions are incorrectly specified, or the data are out of order, or the dimensions are so tight fitting that roundoff causes the net volume of the region to be negative. If the error is caused by roundoff, adjust the appropriate dimensions slightly. See Sect. F11.4.4.

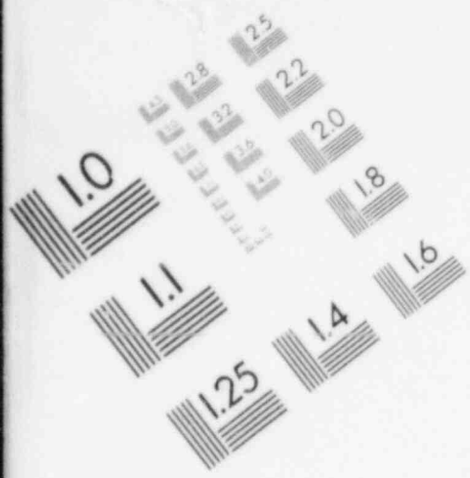
K5-97 ERROR ERROR. THE VOLUME FOR UNIT _____ IS NEGATIVE.

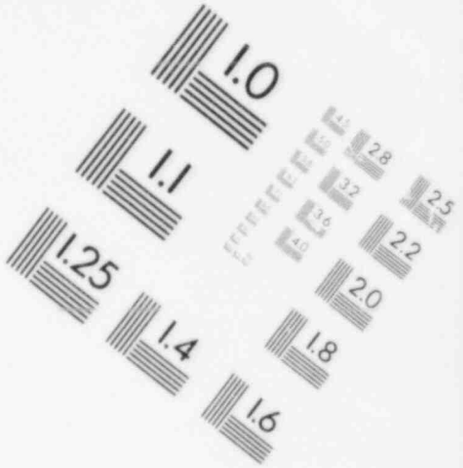
This message is from subroutine VOLUME. A negative volume for a unit or box type can be caused by having intersecting regions within the unit, or by having a unit or box type consisting of one region and having a positive dimension smaller than the negative dimension on one or more faces. Message K5-95 or K5-96 may accompany this message. See Sect. F11.4.4 for assistance in specifying the geometry data correctly.

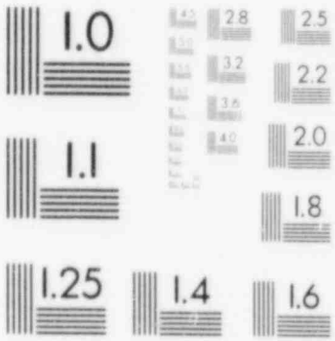
K5-98 INVALID GEOMETRY ENCOUNTERED FOR THE LAST GEOMETRY REGION. IGEO=_____

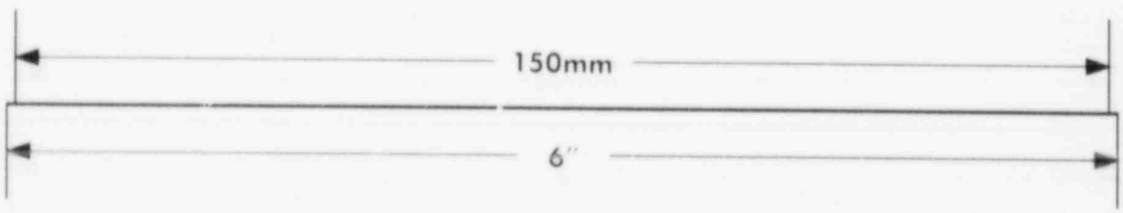
This message is from subroutine CORSIZ. It indicates that an unrecognized geometry word was entered for the last geometry region. This fact should have triggered several error messages prior to this time. This message is usually indicative of a code error.

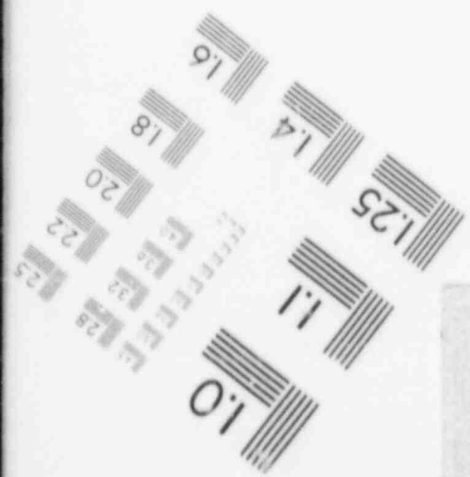
IMAGE EVALUATION
TEST TARGET (MT-3)

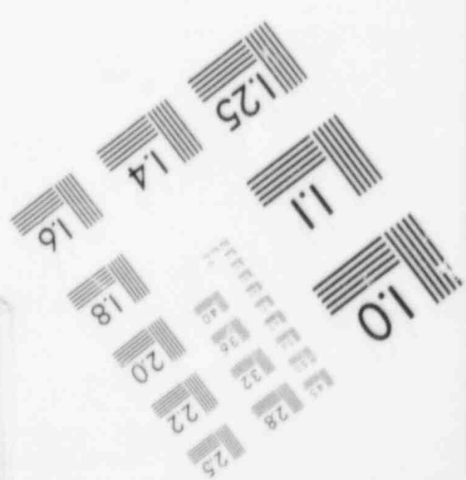












K5-99 *** ERROR IN UNIT _____ *** THE LAST GEOMETRY REGION OF A UNIT UTILIZED IN THE UNIT ORIENTATION DESCRIPTION OF THE ARRAY DATA MUST BE A CUBE OR CUBOID.

This message is from subroutine ARASIZ. It can occur when a single unit problem whose outer region is not a cube or cuboid is specified as a 1x1x1 array. To eliminate this message, add a cuboidal outer region containing void, or remove the array data. If the problem is an array problem, be sure each unit used in the unit orientation description ends with a cube or cuboid. See Sects. F11.4.4 and F11.4.5.

K5-100 THIS PROBLEM WILL NOT BE RUN BECAUSE ERRORS WERE ENCOUNTERED IN THE INPUT DATA.

This message is from subroutine MASTER and indicates that other error messages were printed in the problem output. Find these messages and correct the data accordingly. A STOP 129 is executed when this message is printed.

K5-101 *** ERROR *** NO FISSILE MATERIAL WAS FOUND IN SUBROUTINE START.

This message is from subroutine START. It indicates that none of the mixtures utilized in this problem have a fission spectrum associated with them. Either the geometry data did not specify a fissionable mixture number, the mixing table is incorrect, the wrong mixed cross section data set was mounted, or the mixed cross section data set was incorrectly or incompletely made. A STOP 128 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-102 THE START DATA SPECIFIES THAT NEUTRONS CAN BE STARTED IN THE REFLECTOR. HOWEVER NEUTRONS WILL NOT BE STARTED BECAUSE THE OUTER REGION OF THE REFLECTOR IS NOT A CUBE OR CUBOID. NEUTRONS CAN BE STARTED FOR THE EXISTING GEOMETRY IF XSM, XSP, YSM, YSP, ZSM AND ZSP ARE ENTERED AS START DATA.

This message is from subroutine START. Start type 0 allows starting points throughout noncuboidal regions. If a start type other than 0 or 6 is desired and the outermost region of the reflector is not a cube or cuboid, data must be input to specify an imaginary cube or cuboid within this outer region. See Sect. F11.4.8 for assistance in specifying this data.

K5-103 START TYPE _____ IS OUT OF RANGE.

This message from subroutine START indicates that the start type was less than zero or greater than 6. The start type is defined by entering the keyword NST= followed by the desired start type in the start data. The available starting options are given in Table F11.4.6, Sect. F11.4.8.

K5-104 A POSIT ERROR INDICATES THAT THE POINT X=_____ Y=_____ Z=_____ DOES NOT OCCUR WITHIN UNIT _____

This message from subroutine START may result from precision difficulties. It is allowed to occur a maximum of five times before being considered fatal. A code error may be the cause of this message if it becomes fatal.

K5-105 ***** WARNING, ONLY _____ INDEPENDENT STARTING POSITIONS WERE GENERATED. *****

This message is from subroutine START. KENO V must have *npb* (NPG=, see parameter data, Sect. F11.4.3) starting positions. This message is to inform the user that fewer than *npb* starting positions were generated. The remaining starting positions are randomly selected from those that were generated, thus giving duplicate starting positions. If the number of independent starting positions is nearly *npb*, the starting distribution is probably acceptable. If it is much smaller than *npb*, a different start type should be used to give a better starting distribution. See the start data in Sect. F11.4.8 for assistance.

K5-106 POSIT ERROR --- ILLEGAL GEOMETRY TYPE X=_____ Y=_____ Z=_____ K1=_____ K2=_____ K=_____ IGEO=_____

This message from subroutine POSIT is usually the result of an incorrectly specified starting point for the initial source distribution when NST=3, 4 or 6 is specified in the start data. The starting point may not be consistent with the unit's position in the global array. The message can also be caused by a code error that was introduced when changes were made to the code. X, Y, Z is the location of the neutron, K1 and K2 are the first and last regions of the unit, and K is the region whose geometry type is illegal. IGEO is the geometry type. If the geometry word (*fgeom*) for the region is correct, IGEO should be correct. See Sect. F11.4.4 for correct geometry words.

K5-107 POSIT ERROR X=_____ Y=_____ Z=_____ K1=_____ K2=_____

This message from subroutine POSIT may result from precision difficulties or a code error. X, Y, Z is the location of the neutron. K1 is the region it is in and K2 is the region it is trying to go to. K5-104 is a companion message.

K5-108 POSITION (___ _____) IS NOT VALID FOR THE POSITION OF THE SPIKE FOR START TYPE 2.

This message from subroutine START2 indicates that NXS, NYS or NZS was entered as zero. See Sect. F11.4.8 for the correct start data specification.

K5-109 ***** INVALID GEOMETRY TYPE IN START. IGEO=_____ *****

This message from subroutine STRTSU is probably the result of a code error.

K5-110 ***** ERROR ERROR THE PROBLEM WILL NOT BE EXECUTED BECAUSE NO FISSILE MATERIAL WAS FOUND. *****

This message from subroutine VOLFIS occurs when the volume of fissile material is found to be zero. Check to be sure the fissile material was correctly specified in the geometry data, check the volume of the fissile material in the printout to be sure it is nonzero, verify that the mixing table is correct or the correct ice mixed cross-section data set is used if a mixing table is not used. A STOP 131 is executed when this message is printed.

K5-111 ** RESTART DATA IS NOT AVAILABLE FOR RESTARTING WITH GENERATION _____ AS SPECIFIED IN THE INPUT DATA **
** HOWEVER, AVAILABLE RESTART DATA WAS ALLOWED RESTARTING WITH GENERATION _____ **

This message from subroutine RDCALC indicates that *nbas* (BEG=, Sect. F11.4.3) was not consistent with the set of restart data that was to be used. A set of restart data is written every *nrstrt* (RES=, Sect. F11.4.3). The value of *nbas* should be one greater than one of these generations.

K5-112 ERROR IN RESTART. PARAMETER DATA AND RESTART DATA DO NOT AGREE.
NUMBER PER GENERATION FROM RESTART, NPBT=_____
NUMBER PER GENERATION FROM INPUT DATA, NPB=_____
NUMBER OF ENERGY GROUPS FROM RESTART, NGPT=_____
NUMBER OF ENERGY GROUPS FROM INPUT DATA, NGP=_____

This message is from subroutine RDCALC. A restarted problem MUST use the same number per generation and the same number of energy groups as the parent problem that wrote the restart data. Verify that the correct data set is mounted on unit *rstrt*. (RST=, Sect. F11.4.3). This message can also be caused by a code error.

K5-113 ERROR IN RESTART. PARAMETER DATA AND RESTART DATA DO NOT AGREE. FISSION DENSITIES, FLUXES, OR REGION DEPENDENT FISSIONS AND ABSORPTIONS WERE REQUESTED, BUT THE GEOMETRY DATA IS INCONSISTENT.
NUMBER OF GEOMETRY REGIONS FROM RESTART, KREFT=_____
NUMBER OF GEOMETRY REGIONS FROM INPUT DATA, KREFM=_____

This message is from subroutine RDCALC. Verify that the correct data set is mounted on unit *rstrt* (RST=, Sect. F11.4.3). A code error can also cause this message.

K5-114 *** ERROR ***** ERROR *** PARAMETER DATA SPECIFIED FLUXES BUT THE RESTART DATA DID NOT INCLUDE FLUXES.

This message is from subroutine RDCALC. The restarted problem can turn off fluxes if the parent case that wrote the restart data set calculated fluxes. However, if the parent case did not calculate fluxes, the restarted problem cannot calculate fluxes either. If the correct restart data set was mounted on *rstrt* (RST=, Sect. F11.4.3), the parameter data FLX=YES must be removed from the input data or FLX=NO must be entered later in the parameter data of the restarted problem.

K5-115 *** ERROR ***** ERROR *** PARAMETER DATA SPECIFIED REGION DEPENDENT FISSIONS AND ABSORPTIONS, BUT THEY WERE NOT INCLUDED ON RESTART.

This message is from subroutine RDCALC. The restarted problem specified FAR=YES in the parameter data block but the parent case that wrote the restart data set did not calculate region-dependent fissions and absorptions. The restarted problem can turn off region-dependent data if the parent case calculated them, but cannot turn them on if they were not calculated by the parent case. Verify that the correct restart data set is mounted on *rstrt* (RST=, Sect. F11.4.3). Remove FAR=YES from the parameter data of the restarted problem or add FAR=NO later in the parameter data. Section F11.5.3 illustrates methods of changing the parameter input data.

K5-116 *** ERROR ***** ERROR *** EXECUTION IS TERMINATED *** ERROR *** *** ERROR ***

This message from subroutine RDCALC is a companion to messages K5-112 through K5-115. A STOP 121 is executed when this message is printed.

K5-117 THE CALCULATION WAS TERMINATED BECAUSE ERRORS WERE ENCOUNTERED IN THE START DATA.

This message is from subroutine GUIDE. It will be accompanied by one or more of messages K5-101 through K5-104 or K5-106 through K5-110. A STOP 130 is executed when this message is printed.

K5-118 EXECUTION TERMINATED. RAKBAR HAS BECOME ZERO OR NEGATIVE.

This message is from subroutine GUIDE. If this message appears without other error messages, a code error is the probable cause.

K5-119 JOB PULLED GENERATION=_____ NEUTRON=_____

This message from subroutine GUIDE indicates that the problem is looping, or the time allotted for each generation, *tbtch* (TBA=, Sect. F11.4.3) is too small. If *tbtch* (TBA=) is increased significantly and the message occurs again for the same generation and the same neutron, it is due to a code error.

K5-120 EXECUTION TERMINATED DUE TO INSUFFICIENT IO'S. APPROXIMATELY _____ IO'S ARE NEEDED PER GENERATION, BUT ONLY _____ REMAIN.

This message is from subroutine GUIDE. The problem can be resubmitted if more histories are desired. Be sure to change the job control language to request sufficient IO's to allow the problem to run.

K5-121 EXECUTION TERMINATED DUE TO INSUFFICIENT TIME IN THE JOB STEP. _____ SECONDS ARE NEEDED PER GENERATION, BUT ONLY _____ REMAIN IN THE JOB STEP.

This message is from subroutine GUIDE. If more histories are desired, change the job control language to allow adequate time and resubmit the problem.

K5-122 EXECUTION TERMINATED DUE TO EXCEEDING THE TIME SPECIFIED FOR THE PROBLEM.

This message is from subroutine GUIDE. If more histories are desired, increase *tmax* (TME=, Sect. F11.4.3) to allow computation of the desired number of histories.

K5-123 EXECUTION TERMINATED DUE TO COMPLETION OF THE SPECIFIED NUMBER OF GENERATIONS.

This message from subroutine GUIDE states that the requested number of histories have been completed. If more histories are desired, increase the number of generations (GEN=, Sect. F11.4.3).

K5-124 ***** ERROR ***** THE OPTION TO USE EXTRA 1-D'S WAS SPECIFIED BUT ID NO._____ WAS NOT FOUND IN THE EXTRA 1-D ARRAY.

This self-explanatory message is from function INDX. If extra 1-D's are specified in the parameter data (X1D=, Sect. F11.4.3), extra 1-D ID's must be entered as data. See Sect. F11.4.9. A STOP 107 is executed when this message is printed.

K5-125 EXCEEDED NEUTRON BANK SIZE

This message from subroutine BANKER indicates that the number of banked particles exceeds the bank size. This can be corrected by increasing *nbank* (NBK=, Sect. F11.4.3).

K5-126 ***** CROSS ERROR _____ _____ _____ _____ _____ _____ _____ _____

This message from subroutine CROS indicates a code error. The printed data, left to right, are: IGEO,K,X,Y,Z,X1,Y1,Z1. IGEO is the geometry type, K is the region number, X,Y,Z is the current position and X1,Y1,Z1 is the end point of the path. A STOP 103 is executed when this message is printed.

K5-127 ********** ERROR.....NHCYL_____ **********

This message from subroutine CROS indicates invalid hemicylinder information as the result of a code error. A STOP 104 is executed when this message is printed.

K5-128 NEUTRON BANK IS FULL. SPLITTING NOT ALLOWED.

This message from subroutine TRACK indicates that the neutron bank is too small to allow additional splitting. This can occur if the bank size, *nbank* (NBK=, Sect. F11.4.3) is too small, if the biasing or weighting data are incorrect (Sect. F11.4.7), or if the biasing data are incorrectly utilized in the geometry description (Sect. F11.4.4).

K5-129 *** ERROR IN SUBROUTINE ALBIN *** FACE NUMBER _____ USES _____ _____
ALBEDO NUMBER=_____ INCIDENT XSEC ENERGY GROUP=_____
INCIDENT ALBEDO ENERGY GROUP=_____
INCIDENT ANGLE INDEX=_____ RANDOM NUMBER=_____

This message from subroutine ALBIN indicates that a code error was encountered when trying to determine the output energy group during the albedo treatment.

K5-130 *** ERROR IN SUBROUTINE ALBIN *** FACE NUMBER_____ USES _____ _____
ALBEDO NUMBER=_____ RETURNING XSEC ENERGY GROUP=_____
INCIDENT ALBEDO ENERGY GROUP=_____
INCIDENT ANGLE INDEX=_____
RETURNING ALBEDO ENERGY GROUP=_____
RANDOM NUMBER=_____

This message from subroutine ALBIN indicates that a code error was encountered while trying to compute the returning angle in the albedo treatment.

K5-131 NO FISSIONS

This message from subroutine NSTART indicates that none of the generations encountered a fissile material so no fission points were generated.

K5-132 WARNING....ONLY _____. INDEPENDENT FISSION POINTS WERE GENERATED

This message from subroutine NSTART indicates that less than *npb* (NPG=, Sect. F11.4.3) fission points were generated during the previous generation. Because *npb* fission points are required by the code, the remaining fission points are randomly selected from those that were generated, thus utilizing duplicate fission points. If the number of fission points is considerably less than *npb* for most of the generations, the answer can be affected.

K5-133 NUMBER OF GENERATIONS RUN WAS INSUFFICIENT TO EDIT

This message from subroutine KEDIT occurs if the number of generations completed is less than *nskip+1* (NSK=, Sect. F11.4.3). In this instance, the summaries for the problem cannot be printed.

K5-134 FLUXES FOR UNIT _____ WILL NOT FIT IN CORE.

This message from subroutine FITFLX indicates that the flux array is too large for the available core space. More space can be allocated in the job control language if desired.

K5-135 GEOMETRY TYPE - IGEO - OUT OF RANGE IN CRMAX

This message from subroutine CRMAX occurs if the geometry indicator is invalid as the result of a code error. A STOP 102 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-136 GEOMETRY TYPE - IGEO - OUT OF RANGE IN SRMAX

This message from subroutine SRMAX occurs if the geometry indicator is invalid as the result of a code error. A STOP 127 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-137 INPUT DATA SAID TO PRINT WEIGHTS, BUT THEY WERE NOT PRINTED BECAUSE THE NUMBER OF ENERGY GROUPS EXCEEDS THE AVAILABLE SPACE.

This message is from subroutine MASTER. In order to accommodate the specified number of energy groups, more computer storage must be requested in the job control language.

K5-138 ********** A WEIGHT OF 0.0 INDICATES THAT WEIGHTS WERE NOT READ OR GENERATED FOR THE BIAS ID. **********
********** WEIGHTS OF 0.0 WILL BE DEFAULTED TO 0.5 PRIOR TO EXECUTION **********

This message is from subroutine PRTWTS. It is printed to alert the user that weights were not entered, defaulted, or generated. This message may appear as the result of a code error.

K5-139 ***** ERROR ***** NO VALID MIXTURES WERE FOUND IN THE GEOMETRY DESCRIPTION.

This message from subroutine ICEMIX indicates that the geometry data did not specify any valid mixtures. Check the geometry data (Sect. F11.4.4) and correct any errors that are found. This message can also be triggered if the unit orientation data description is not properly entered for geometry having more than one unit or box type.

K5-140 ***ERROR*** NOT ENOUGH STORAGE FOR START TYPE 6 DATA. 1ST IS AMT NEEDED, 2ND IS AMT AVAILABLE.
PERTINENT CONSTANTS (1) (2)

This message from subroutine SAVST6 indicates that more computer storage is necessary to run this problem. At least (1) words of storage are needed to run the problem, but only (2) words of storage were available. Increase the amount of computer storage requested in the job control language to correspond to the amount of storage needed. A STOP 126 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-141 DATA CANNOT BE CHANGED WHEN A PROBLEM IS RESTARTED AT A GENERATION GREATER THAN ONE.

This message from subroutine DATAIN is printed if data other than parameter data are entered for a problem being restarted at a generation greater than one. If data other than certain parameter data are to be changed, the problem must be restarted with the first generation. The error flag is set so the problem will not execute.

K5-142 NO GEOMETRY DATA HAS BEEN SPECIFIED IN THE INPUT DATA.

This message from subroutine DATAIN indicates that a geometry data block was not entered for the problem either as input data or from the restart unit. Correct the data and resubmit the problem. A STOP 135 is executed in conjunction with this message and a traceback may be printed from subroutine STOP.

K5-144 DUE TO INCONSISTENCIES BETWEEN INPUT AND RESTART DATA, MATRIX INFORMATION BY UNIT TYPE WILL BE CALCULATED BUT NOT PRINTED. INPUT DATA SET MKU=NO, BUT DATA FROM THE RESTART UNIT SPECIFIED YES.

This self-explanatory warning message is from subroutine PARAM. The matrix information by unit type cannot be eliminated if it was calculated by the original problem (parent case) that wrote the restart data. However, printing it can be avoided. Verify that the correct problem is being used for restarting the problem (the title is printed at the bottom of the parameter tables). Also verify the specification of the restart unit, RST, in the third table of the output.

K5-145 DUE TO INCONSISTENCIES BETWEEN INPUT AND RESTART DATA, MATRIX INFORMATION BY UNIT LOCATION WILL BE CALCULATED BUT NOT PRINTED. INPUT DATA SET MKP=NO, BUT DATA FROM THE RESTART UNIT SPECIFIED YES.

This self-explanatory warning message is from subroutine PARAM. The matrix information by unit location (also called array position or position index) cannot be eliminated if it was calculated by the original problem (parent case) that wrote the restart data. However, printing it can be avoided. Verify that the correct problem is being used for restarting the problem (the title is printed at the bottom of the parameter tables). Also verify the specification of the restart unit, RST, in the third table of the output.

K5-146 *** ERROR ***** ERROR *** PARAMETER DATA SPECIFIED MATRIX INFORMATION BY UNIT TYPE BUT IT WAS NOT FOUND ON THE RESTART UNIT.

This message from subroutine RDCALC is printed if a restarted problem requests matrix information by unit type when it was not requested and calculated by the original problem (parent case) that wrote the restart data. Verify that the correct restart data is being used and that the restart unit (RST) is correctly specified. Eliminate the request for matrix information by unit type (MKU=, in the parameter data) if it is not necessary. The problem must be restarted with the first generation (BEG=1, in the parameter data) if matrix information by unit type is required and was not calculated by the parent case. A STOP 121 is executed in conjunction with this message.

K5-147 *** ERROR ***** ERROR *** PARAMETER DATA SPECIFIED MATRIX INFORMATION BY UNIT LOCATION BUT IT WAS NOT FOUND ON THE RESTART UNIT.

This message from subroutine RDCALC is printed if a restarted problem requests matrix information by unit location (also called array position or position index) when it was not requested and calculated by the original problem (parent case) that wrote the restart data. Verify that the correct restart data is being used and that the restart unit (RST) is correctly specified in the first table following the parameter table. Eliminate the request for matrix information by unit location (MKP=, in the parameter data) if it is not necessary. If matrix information by unit location is required and it was not calculated by the parent case, the problem must be restarted with the first generation (BEG=1, in the parameter data). A STOP 121 is executed in conjunction with this message.

K5-149 A CROSS SECTION LIBRARY WAS SPECIFIED FOR A RESTARTED PROBLEM, BUT MIXING TABLE DATA WAS NOT AVAILABLE. THE PROBLEM WILL NOT EXECUTE.

This message from subroutine RDRST means that LIB= was entered in the parameter data block and a mixing table data block was not available. A flag is set to terminate execution when the data reading has been completed. If cross sections are to be used from the restart unit (RST=), eliminate the LIB= or XSC= from the parameter data. If new cross sections are to be mixed, LIB= must be specified in the parameter data. The ID's in the mixing table must be available on the cross-section library specified by LIB=. A problem can be restarted using a new mixed cross-section library by specifying XSC= in the parameter data.

K5-150 *** ERROR ****** ERROR *** TOO FEW ENTRIES WERE SUPPLIED IN THE REFLECTOR GEOMETRY DESCRIPTION.

This message from subroutine KENOG indicates that too few data entries were supplied for the geometry word REFLECTOR. The mixture ID, bias ID, one of the thickness/region specifications or the number of regions to be generated was omitted or incorrectly specified. Each REFLECTOR entry requires (1) a mixture ID, (2) a bias ID, (3) N entries for the thickness/region specifications, and (4) the number of regions to be generated. The thickness/region can be obtained from the Increment Thickness column of Table F11.4.5 for the material to be used in the regions generated by the REFLECTOR specification. N is the number of thickness/region specifications required by the geometry shape: N=1 for spheres or hemispheres, N=3 for cylinders and hemicylinders, and N=6 for cubes, cuboids, and cores. A flag is set to terminate the problem when the input data reading is completed.

K5-151 *** ERROR *** _____ IS AN INVALID PARAMETER NAME IN THE ARRAY DATA

This message from subroutine ARAYIN indicates that a parameter name was misspelled or the data was out of order. See Sect. F11.4.5 for a list of the array parameter names.

K5-152 *** ERROR *** IRET= _____ A PREMATURE TERMINATION WAS ENCOUNTERED WHILE READING ARRAY DATA.
IRET=1 INDICATES AN END WAS FOUND. IRET=2 INDICATES AN END OF FILE.

This message from subroutine ARAYIN indicates that an array number was specified without entering the corresponding UNIT ORIENTATION DESCRIPTION. See Sects. F11.4.5 and F11.5.6 for assistance.

K5-153 FIRST TWO NUMBERS ARE WORDS OF STORAGE NEEDED AND ALLOCATED. THIRD IS ADDITIONAL REGION SIZE NEEDED.

This message from subroutine SORTA indicates that the allocated computer storage is too small to allow loading the unit orientation data prior to determining the nesting level of the arrays defined by the unit orientation data. The first number printed is the amount of storage, in words, needed to hold this data. The second number is the amount of computer storage that was allocated, and the third number is the minimum additional region size, in units of K bytes, necessary to hold this data. Increase the region size for the "go step" in the job control language by the additional required region size (the third number) and resubmit the problem.

K5-154 *** ERROR *** LEVELA= _____ IS LARGER THAN _____ THE NUMBER OF ARRAYS.

This message from subroutine SORTA indicates that the array data specified in the problem is recursively nested. An example of this is:

array 1 contains array 3
array 2 contains array 1
array 3 contains array 2

Thus the definition of array 1 and array 3 are intertwined in a never ending loop. Correct the array data (Sect. F11.4.5) and resubmit the problem. If the input data did not specify recursive nesting, a code error has occurred.

K5-155 *** ERROR *** THE NESTING FLAG OR NUMBER OF ARRAY LEVELS HAS BEEN DESTROYED BY A CODE ERROR.
THE ORIGINAL NESTING FLAG WAS SET _____ IT IS NOW SET _____
THE ORIGINAL NESTING LEVEL WAS _____ IT IS NOW _____

This self-explanatory message is from subroutine LODARA. A STOP 140 is executed in conjunction with this message.

K5-156 *** ERROR *** MIXTURE _____ CONTAINS AT LEAST ONE ZERO VALUE FOR THE TOTAL CROSS SECTION.

This message is from subroutine XSEC. All the total cross sections must be positive. Zero total cross sections can occur if all the components of a mixture are mixed with a zero number density. Correct the mixing table for the specified mixture and resubmit the problem.

K5-157 *** ERROR *** THE FIRST HOLE IN A UNIT MUST FOLLOW A VALID GEOMETRY REGION.

A STOP 141 accompanies this message from subroutine READGM. If holes are to be utilized in the geometry region data, (Sect. F11.4.4) they must follow the region in which they are to be placed. This message indicates that HOLE was the first geometry description in a unit or was placed inside a CORE or ARRAY description. Correct the geometry region data and resubmit the problem.

K5-158 *** ERROR *** ERROR IN THE NUMBER OF HOLES. IHOL=_____ NUMHOL=_____

This message is from subroutine READGM. A code error is the probable cause of this error.

K5-159 *** ERROR *** ARRAY NUMBER _____ SPECIFIED IN THE GEOMETRY REGION DATA WAS NOT ENTERED IN THE ARRAY DATA.

This message from subroutine SORTA occurs if the array number specified for an ARRAY region description of the EXTENDED GEOMETRY data (Sect. F11.4.4) did not have the corresponding UNIT ORIENTATION DATA entered in the ARRAY DATA (Sect. F11.4.5). A STOP 142 is executed in conjunction with this message. Correct the data and resubmit the problem.

K5-160 THE HOLES ARE RECURSIVELY NESTED.

This message from subroutine HOLE indicates that the geometry region data description (Sect. F11.4.4) specifies holes that are recursively nested. This can happen if a unit contains a hole whose definition traces back to the same unit or are defined in terms of each other. A simple example of recursive nesting is:

UNIT 1 CUBE 0 1 2P10.0 HOLE 2 3*0.0
UNIT 2 CUBE 0 1 2P10.0 HOLE 1 3*0.0

Thus unit 1 contains unit 2 and unit 2 contains unit 1. Check the geometry region data for recursive nesting. In the absence of recursive nesting, a code error is the probable cause of this message. A STOP 143 is executed when this message occurs and a traceback is printed.

K5-161 *** ERROR *** THE GLOBAL ARRAY SPECIFIED IN THE GEOMETRY REGION DATA IS _____ BUT THE GLOBAL ARRAY SPECIFIED IN THE ARRAY DATA IS _____ EXECUTION IS TERMINATED.

This message is from subroutine FLDATA. A STOP 144 is executed in conjunction with it. The global array specified in the array data (Sect. F11.4.5) was entered using the keyword GBL=. It was not consistent with the implied global array from the geometry region data (Sect. F11.4.4). The global array number in the geometry region data is defined to be the array number of the last ARRAY or CORE description that does not immediately follow a UNIT or BOX TYPE definition, i.e., other geometry definitions occur between UNIT and CORE or ARRAY. Correct the data and resubmit the problem.

K5-162 *** ERROR *** THE GLOBAL ARRAY WAS NOT CONSISTENTLY SPECIFIED. THE ARRAY DATA SPECIFIED ARRAY NUMBER _____ AND THE GEOMETRY DATA SPECIFIED ARRAY NUMBER _____

This message from subroutine SORTA occurs if GBL= in the array data (Sect. F11.4.5) does not agree with the global array number implicitly set in the geometry region data (Sect. F11.4.4). The geometry region data defines the global array number to be the array number of the last ARRAY or CORE description that does not immediately follow a UNIT or BOX TYPE definition, i.e., other geometry definitions occur between UNIT and CORE or ARRAY. A flag is set to terminate execution when the data reading is completed.

K5-163 *** ERROR *** ARRAY NUMBER _____ IS _____ WORDS LONG, BUT ONLY _____ WORDS ARE AVAILABLE WHEN WRITING RESTART. THE RESTART FILE WILL BE INCOMPLETE. THE PROBLEM WILL NOT EXECUTE.

This message is from subroutine WRTARA. It indicates that the available computer storage is too small to hold the array data from the direct access device. This in turn will cause the restart data file to be incomplete. Increase the requested storage space in the job control language and resubmit the problem.

K5-164 *** WARNING *** ARRAY NUMBER _____ IS _____ WORDS LONG, BUT ONLY _____ WORDS ARE AVAILABLE WHEN READING RESTART. THIS ARRAY WILL NOT BE AVAILABLE. THE PROBLEM WILL NOT EXECUTE IF THIS ARRAY IS USED.

This message is from subroutine RDARA. It occurs when the restart data file is being loaded from the restart unit and the allocated space is insufficient to hold the data for the specified array. Resubmit the problem, requesting more storage space in the job control language.

K5-165 *** ERROR *** GEOMETRY TYPE OUT OF RANGE IN SUBROUTINE ADJUST. IGEOH=_____

This message from subroutine ADJUST indicates that the geometry type, IGEO, falls outside the allowable range of 1 through 19. This can occur if a HOLE references an undefined unit number. If all the unit numbers referenced by the holes are valid, a code error is the probable cause.

K5-166 HOLE NUMBER _____ INTERSECTS REGION NUMBER _____

This form of message K5-166 is from subroutine HOLEXT. It indicates that the specified hole intersects the region in which the hole was placed. A flag is set to terminate execution when the data reading is completed. Correct the geometry data (Sect. F11.4.4) and resubmit the problem.

K5-167 GEOMETRY TYPE - IGEO - OUT OF RANGE IN CRMIN

This message from subroutine CRMIN is accompanied by a STOP 145. The geometry type, IGEO, must be greater than zero and less than 20. If it does not fall in this range, a code error is the probable cause.

K5-168 GEOMETRY TYPE - IGEO - OUT OF RANGE IN SRMIN

This message from subroutine SRMIN is accompanied by a STOP 146. The geometry type, IGEO, must be greater than zero and less than 20. If it does not fall in this range, a code error is the probable cause.

K5-169 HOLE NUMBER _____ INTERSECTS REGION NUMBER _____

This message from subroutine HOLHOL occurs if a hole intersects the first region interior to the region in which the hole is implaced. This may be a real intersection or it may be due to roundoff in processing the holes. If it is due to roundoff, adjust *xhole*, *yhole* and/or *zhole* and/or the dimensions as appropriate. This message indicates that the problem will be terminated when the data reading is completed.

K5-170 ERROR IN KENO 5 PICTURE DATA - KEYWORD _____ IS NOT VALID.

This message from subroutine RDPLOT indicates that an invalid keyword was encountered in the plot data. The data may be out of order or spelled incorrectly. See Sect. F11.4.11 for a list of acceptable keywords.

K5-171 ERROR - DIRECTION COSINES DOWN THE PAGE WERE ALL INPUT AS ZERO.

This message from subroutine RDPLOT indicates that the values for UDN=, VDN= and WDN= were all zero. A zero value vector does not define a direction, so an error has occurred. See Sect. F11.4.11 for information concerning direction cosines down the page.

K5-172 ERROR - DIRECTION COSINES ACROSS THE PAGE WERE ALL INPUT AS ZERO.

This message from subroutine RDPLOT indicates that the values for UAX=, VAX= and WAX= were all zero. This is an error because a zero value vector does not define a direction. See Sect. F11.4.11 for assistance in defining direction cosines across the page.

K5-173 ERRORS WERE DETECTED IN THE INPUT DATA FOR THIS PICTURE. IT WILL NOT BE DRAWN.

This message from subroutine RDPLOT is a companion to messages K5-174, K5-170, K5-171 and K5-172. Correct the error that triggered the companion message and resubmit the problem.

K5-174 ERROR IN KENO 5 PICTURE DATA - _____ _____ SHOULD BE ENTERED AS _____ YES OR _____ NO.

This self-explanatory message is from subroutine RDPLOT. Correct the error and resubmit the problem. See Sect. F11.4.11 for assistance.

K5-175 DUE TO INCONSISTENCIES BETWEEN INPUT AND RESTART DATA, MATRIX INFORMATION BY HOLE WILL BE CALCULATED BUT NOT PRINTED. INPUT DATA SET MKH=NO, BUT DATA FROM THE RESTART UNIT SPECIFIED YES.

This self-explanatory warning message is from subroutine PARAM. The matrix information by hole cannot be eliminated if it was calculated by the original problem (parent case) that wrote the restart data. However, printing it can be avoided. Verify that the correct problem is being used for restarting the problem (the title is printed at the bottom of the parameter tables). Also verify the specification of the restart unit, RST, in the third table of the computer output.

K5-176 DUE TO INCONSISTENCIES BETWEEN INPUT AND RESTART DATA, MATRIX INFORMATION BY ARRAY WILL BE CALCULATED BUT NOT PRINTED. INPUT DATA SET MKA=NO, BUT DATA FROM THE RESTART UNIT SPECIFIED YES.

This self-explanatory warning message is from subroutine PARAM. The matrix information by array cannot be eliminated if it was calculated by the original problem (parent case) that wrote the restart data. However, printing it can be avoided. Verify that the correct problem is being used for restarting the problem (the title is printed at the bottom of the parameter tables). Also verify the specification of the restart unit, RST, in the third table in the computer output.

K5-177 *** ERROR ***** ERROR *** PARAMETER DATA SPECIFIED MATRIX INFORMATION BY HOLE BUT IT WAS NOT FOUND ON THE RESTART UNIT.

This message from subroutine RDCALC is printed if a restarted problem requests matrix information by hole when it was not requested and calculated by the original problem (parent case) that wrote the restart data. Verify that the correct restart data file is being used and that the restart unit (RST) is correctly specified. Eliminate the request for matrix information by hole (MKH=, in the parameter data) if it is not necessary. The problem must be restarted with the first generation (BEG=1, in the parameter data) if matrix information by hole is required and was not calculated by the parent case. A STOP 121 is executed in conjunction with this message.

K5-178 *** ERROR ****** ERROR *** PARAMETER DATA SPECIFIED MATRIX INFORMATION BY ARRAY BUT IT WAS NOT FOUND ON THE RESTART UNIT.

This message from subroutine RDCALC is printed if a restarted problem requests matrix information by array (also called array position or position index) when it was not requested and calculated by the original problem (parent case) that wrote the restart data. Verify that the correct restart data is being used and that the restart unit (RST) is correctly specified in the first table following the parameter tables. Eliminate the request for matrix information by array (MKA=, in the parameter data) if it is not necessary. If matrix information by array is required and it was not calculated by the parent case, the problem must be restarted with the first generation (BEG=1, in the parameter data). A STOP 121 is executed in conjunction with this message.

K5-179 HOLE NUMBER _____ INTERSECTS HOLE NUMBER _____ IN REGION NUMBER _____

This message from subroutine HOLHOL indicates that the specified holes intersect. Check the dimensions and origins of the units being placed in the region.

K5-180 ERROR IN KENO 5 PICTURE DATA - KEYWORD _____ IS NOT VALID.

This message from subroutine RDPLOT indicates that the plot or picture data are out of order or a keyword is incorrectly spelled. See Sect. F11.4.11 for a list of correct keywords.

K5-181 *** ERROR *** LPIC IS OUT OF RANGE. LPIC= _____

This message from subroutine PRTPLT indicates that a code error has occurred or the type of plot (PIC=) was not properly specified. *lpic*=1 for a mixture map, *lpic*=2 for a unit map and *lpic*=3 for a bias ID map. Any other values of *lpic* are invalid.

K5-182 INSUFFICIENT SPACE FOR PLOTTING ARRAYS. FIRST NUMBERS ARE STORAGE NEEDED AND ALLOCATED. LAST IS REQUIRED ADDITIONAL REGION SIZE.

This message from subroutine MASTER is accompanied by a STOP 147. It indicates that the allocated computer storage will not hold the plot data. The first number printed is the amount of storage, in words, needed to hold this data. The second number is the amount of computer storage that was allocated, and the third number is the minimum additional region size, in units of K bytes, necessary to hold the data. Increase the region size for the "go step" in the job control language by this third number and resubmit the problem. If additional computer storage is not available, eliminate the fluxes, fissions and absorptions by region, matrix information or other space consuming options (see Sect. F11.4.3).

K5-183 *** WARNING *** UNIT _____ WAS NOT CHECKED FOR GEOMETRIC CONSISTENCY. IT CONTAINS ARRAY _____ BUT WAS NOT USED IN THE PROBLEM.

This warning message is from subroutine JOMCHK. It means that a unit whose first region is an array was described in the extended geometry data but that unit was not referenced in the unit orientation data (see Sects. F11.4.4 and F11.4.5). KENO usually checks all the geometry region data to be sure it is correct, even when the unit is not used in the problem. The code is unable to make these checks when a unit containing an array is not used in the problem. It is not necessarily an error, but the user should double check to be sure that unit was intentionally omitted from all the arrays.

K5-184 ****** _____ WORDS OF STORAGE ARE NEEDED FOR THE WEIGHTS, BUT ONLY _____ WORDS ARE AVAILABLE. ******

This message from subroutine RDBIAS indicates that the allocated computer storage is not sufficient to process the biasing or weighting data for the problem. Increase the region size in the "go step" of the job control language. The additional required region size is four times (the difference of the two numbers plus 1023) divided by 1024. A STOP 151 is executed when this message is printed.

K5-185 *** ERROR *** THE NUMBER OF SETS OF BIASING CORRELATION DATA EXCEEDS THE NUMBER THAT WAS WRITTEN WHEN THE BIASING DATA WAS READ. _____ WERE WRITTEN, BUT AN ATTEMPT WAS MADE TO READ _____

This message from subroutine WAITIN is accompanied by a STOP 148. It is indicative of a code error, unless accompanied by error messages related to the biasing input data.

K5-186 *** ERROR *** THE NUMBER OF SETS OF BIASING AUXILIARY DATA EXCEEDS THE NUMBER THAT WAS WRITTEN WHEN THE BIASING DATA WAS READ. _____ WERE WRITTEN, BUT AN ATTEMPT WAS MADE TO READ _____

This message from subroutine WAITIN is accompanied by a STOP 149. It is indicative of a code error, unless accompanied by error messages related to the biasing input data.

K5-187 ***** WARNING - INTERVALS IN THE ABOVE RANGE WERE NOT USED. THIS COULD LEAD TO IMPROPER BIASING. *****

This message from subroutine LODWTS is printed to remind the user that at least one of the specified intervals was not utilized in the problem. This can result in improper biasing. Biasing should not be used between fissile units. When biasing is used, it should be flat or increasing as distance from the fissile material increases and flat or decreasing as a history moves toward fissile material. See Sect. F11.5.3 for additional assistance.

K5-188 *** ERROR *** HOLE NUMBER _____ REFERENCES UNDEFINED UNIT NUMBER NUMBER _____

This message is printed by subroutine READGM if the unit number referenced by the hole is less than one or greater than the largest unit number in the geometry data. The message is printed by subroutine HOLE if the unit number referenced by the hole is larger than zero and not greater than the largest unit number in the geometry data, but is a unit number for which all data are missing. The message is printed by subroutine HOLCHK if the unit number referenced by the hole is undefined. Message K5-165 may accompany this message. Specify a valid unit number (*Ihole* - see EXTENDED GEOMETRY DESCRIPTION, Sect. F11.4.4) and resubmit the problem.

K5-189 *** ERROR *** ALBEDOS WERE SPECIFIED FOR A PROBLEM WHOSE OUTER BOUNDARY IS NOT A CUBE OR CUBOID.
THE PROBLEM WILL NOT EXECUTE.

This message from subroutine JOMCHK is printed if the outer geometry region is not a cube or cuboid. A REFLECTOR or REPLICATE geometry description immediately following a CORE, ARRAY or CUBOID is considered to be a cuboid. If reflection is desired on a surface of some other shape, it will be necessary to describe actual geometry regions of the appropriate material.

K5-190 ERROR IN PICTURE DATA - OPTION _____ IS NOT VALID FOR KEYWORD PIC=.

This message from subroutine RDPLOT indicates an incorrect option associated with the keyword PIC=. See Sect. F11.4.11. Acceptable options include MAT, MIX, MIXT, MEDI, BOX, BOXT, UNT, UNIT, IMP, BIAS, WTS, WGT, WGTS or WEIGH.

K5-191 ***** ERROR ***** START TYPE _____ IS INVALID FOR A PROBLEM THAT DOES NOT HAVE A GLOBAL ARRAY. ***** ERROR *****

This message from subroutine DATAIN occurs if the start type (NST= in the start data, Sect. F11.4.8) is 2, 3, 4, or 5. A global array is required in order to use the specified start type.

K5-192 MATRIX INFORMATION BY ARRAY WAS SPECIFIED AS YES IN THE PARAMETER DATA (MKA=), BUT IS NOT ALLOWED BECAUSE ARRAY DATA WERE NOT SPECIFIED.

This warning message from subroutine DATAIN is self-explanatory. The code redefines the problem so matrix information will not be collected by array number.

K5-193 MATRIX INFORMATION BY HOLE WAS SPECIFIED AS YES IN THE PARAMETER DATA (MKH=), BUT IS NOT ALLOWED BECAUSE HOLE DATA WERE NOT SPECIFIED.

This warning message from subroutine DATAIN is self-explanatory. The code redefines the problem so matrix information will not be collected by hole number.

K5-194 MATRIX INFORMATION BY UNIT LOCATION WAS SPECIFIED AS YES IN THE PARAMETER DATA (MKP=), BUT IS NOT ALLOWED BECAUSE A GLOBAL ARRAY WAS NOT SPECIFIED.

This warning message from subroutine DATAIN is self-explanatory. The code redefines the problem so matrix information will not be collected by unit location.

K5-195 ***** ERROR ***** CHARACTER STRING EXCEEDS THE SPECIFIED LENGTH. CHECK FOR ENDING DELIMITER.

This error message is from subroutine RCHRS. It indicates that either the character string exceeds 132 characters or the ending delimiter was omitted for TTL= (plot title, Sect. F11.4.11) or for COM= (unit comment, Sect. F11.4.4, or array comment, Sect. F11.4.5). A STOP 153 is executed when this message is printed.

K5-196 A PROBLEM CANNOT BE RESTARTED WHEN RESTART DATA DO NOT EXIST FOR THE SPECIFIED GENERATION AND THE NEXT GENERATION FOR WHICH RESTART DATA ARE AVAILABLE IS LARGER THAN THE REQUESTED NUMBER OF GENERATIONS. EXECUTION IS TERMINATED.

This message from subroutine RDCALC indicates that a problem was to be restarted but the requested number of generations (GEN=, Sect. F11.4.3) was smaller than the beginning generation number (BEG=, Sect. F11.4.3). The beginning generation number for a restarted problem is the generation at which the calculation of k-effectives and associated information is resumed. Therefore, the number of generations to be run must be larger than the beginning generation number. Correct the data and resubmit the problem. A STOP 154 is executed when this message is printed.

K5-197 *** ERROR IN UNIT _____ *** A VALID GEOMETRY REGION MUST PRECEDE A REPLICATE REGION.

This message from subroutine KENOG indicates that a replicate specification follows an invalid geometry specification (for example, REPLICATE immediately follows a UNIT specification).

K5-198 *** ERROR *** ARRAY _____ CONTAINS AN ERROR IN THE INPUT DATA.

This message from subroutine ARAYIN is printed as the result of an error in the FILL input data for the specified array. One or more messages from the library routine YREAD should immediately precede this message and indicate the nature of the error. Correct the data and resubmit the problem. Messages K5-32, K5-33 and/or K5-85 may also print as a result of this error.

K5-199 *** ERROR *** AN ARRAY WAS SPECIFIED IN THE GLOBAL UNIT, BUT ARRAY DATA WERE NOT ENTERED.

This message from subroutine SORTA occurs if array data are not entered when the global unit geometry specifies an array by using either CORE or ARRAY. A STOP 156 is executed in conjunction with this message. Enter the appropriate array data and resubmit the problem.

F11.7.2 STOP CODES

The STOP codes that are encountered in KENO V are listed in tabular form, indicating the subroutine where they occur and the associated error message. A STOP is executed whenever a fatal error is recognized. Look up the associated message number to determine the appropriate corrective measures. A traceback may be generated whenever subroutine STOP is called to print a message. If no traceback is indicated in the STOP CODE table, a stop is executed as soon as the associated message is printed.

STOP NUMBER	SUBROUTINE	TRACEBACK	ASSOCIATED MESSAGE
100	ARAYIN	yes	K5-25
101	ARAYIN	no	K5-31
102	CRMAX	yes	K5-135
103	CROS	no	K5-126
104	CROS	no	K5-127
105	WRTALB	no	K5-71
106	ICEMIX	no	K5-68
107	INDX	no	K5-124
108	INITAL	no	K5-2
109	INITAL	no	K5-2
110	MAKANG	yes	K5-57
111	MIXCRS	yes	K5-52
112	MIXER	yes	K5-53
113	MIXER	yes	K5-54
114	MIXIT	yes	K5-15
115	NORMID	no	K5-56
116	NSUPG	yes	K5-79
117	NSUPG	no	K5-81
118	PARAM	no	K5-7 or K5-8
119	PRTLBA	no	K5-89
120	RDALB	no	K5-49
121	RDCALC	no	K5-116 or K5-146 or K5-147 or K5-177 or K5-178
122	RDICE	no	K5-50
123	RDRST	no	K5-48
124	RDTAPE	no	K5-69
125	READGM	no	K5-20
126	SAVST6	yes	K5-140
127	SRMAX	yes	K5-136
128	START	yes	K5-101
129	MASTER	no	K5-100
130	MASTER	no	K5-117 or K5-128
131	VOLFIS	no	K5-110
132	WRTICE	no	K5-72
133	WRTRST	no	K5-70
134	TRACK	no	K5-62
135	DATAIN	yes	K5-142
136	DATAIN	yes	K5-143
137	ARAYIN	no	K5-151
138	SORTA	yes	K5-153
139	SORTA	no	K5-154
140	LODARA	no	K5-155
141	READGM	no	K5-157
142	SORTA	no	K5-159
143	HOLE	no	K5-160
144	FLDATA	no	K5-161
145	CRMIN	yes	K5-167
146	SRMIN	yes	K5-168
147	MASTER	yes	K5-182
148	WAITIN	no	K5-185
149	WAITIN	no	K5-186
150	WATES	yes	K5-22
151	RDBIAS	no	K5-184
152	DATAIN	no	K5-46
153	RCHRS	yes	K5-195
154	RDCALC	no	K5-196
155	GEOMIN	yes	K5-25
156	SORTA	yes	K5-25

APPENDIX A

F11.A ALPHABETICAL INDEX OF SUBROUTINES

This section provides a convenient alphabetical index whereby the user can locate a brief summary of the function of a subroutine. It also provides a list of the subroutines from which the subroutine is called in addition to listing the subroutines and library routines it calls. A ** for a page number indicates a FORTRAN-supplied library routine.

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
ACOS	STRTSU		**
ADJUST	HOLCHK	MOVE	F11.3.34
ALBEDO	TRACK		F11.3.43
ALBIN	TRACK	AZIRN	F11.3.47
		FLTRN	
ALBRD	DIFALB	ALBUSE	F11.3.18
ALBUSE	ALBRD	IO	F11.3.18
		RITE	
ALOCAT	MAIN		M2.2.1
	Q#$009		
ALOG	CORRE		**
ANGLES	PRANG	FIND	F11.3.22
		Q	
ARALBA	JOMITY	PRTLBA	F11.3.30
ARASIZ	CORSIZ	LSCAN	F11.3.30
ARAYIN	DATAIN	AREAD	F11.3.7
		BOX	F11.3.9
		CLEAR	
		HLFWRD	
		IREAD	
		MOVE	
		RCHRS	
		RDBOX	
		STOP	
		YREAD	
AREAD	ARAYIN		M2.2.4
	DATAIN		
	INITAL		
	KENOG		
	MIXIT		
	PARAM		
	RCHRS		
	RDBIAS		
	RDORGN		
	RDPLOT		
	RDREF		
	RDSTRT		

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
ARSIN	START1		**
	VOLUME		
AZIRN	ALBIN		M2.2.60
	STRTSU		
	TRACK		
BADMOM	PRANG		F11.3.22
BANKER	GUIDE	MOVE	F11.3.40
			F11.3.41
BOX	ARAYIN		F11.3.9
BOXC	LOADIT		F11.3.32
CHKSTR	GUIDE	MOVE	F11.3.40
		SQRT	F11.3.41
		TRKWRT	
CHOOSE	START4	FLTRN	F11.3.45
	START5		F11.3.45
CLEAR	ARAYIN		M2.2.9
	CORRE		
	DATAIN		
	FILLSG		
	FLDATA		
	GEOMIN		
	GUIDE		
	HOLE		
	HUNTER		
	LODARA		
	LODWTS		
	MAKANG		
	MASTER		
	MATK		
	MESH		
	MIXCRS		
	NNITL		
	NSTART		
	PARAM		
	RDALB		
	RDICE		
	RDPLOT		
	RDTAPE		
	RGUSF[illegible]		
	RT[illegible]A		
	SAVST6		
	SORTA		
	START		
	UNTCRS		
	VOLUME		
CLOSDA	MAIN		M2.2.10
	MIXER		
	Q#$009		
CMPRS	MIXMIX		F11.3.21

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
CORRE	MASTER	ALOG	F11.3.5
		CLEAR	F11.3.25
		RATIO	
		REED	
		RITE	
CORSIZ	JOMITY	ARASIZ	F11.3.29
		SQRT	
COS	STRTSU		**
CRMAX	HOLEXT	CRSPRD	F11.3.34
	JOMCHK	DOTPRD	F11.3.35
		SQRT	
		STOP	
		VECADD	
		VECNRM	
CRMIN	HOLHOL	CRSPRD	F11.3.34
		DOTPRD	F11.3.35
		SQRT	
		STOP	
		VECADD	
		VECNRM	
CROS	TRACK	SQRT	F11.3.47
CRSPRD	CRMAX		F11.3.35
	CRMIN		F11.3.36
	SRMAX		F11.3.35
	SRMIN		F11.3.36
DATAIN	MASTER	ARAYIN	F11.3.4
		AREAD	F11.3.5
		CLEAR	F11.3.6
		EXTRA	
		FLDATA	
		GEOMIN	
		IDX1D	
		IOWRT	
		MIXIT	
		RDBIAS	
		RDPLOT	
		RDREF	
		RDRST	
		RDSTRT	
		RT	
		RTARA	
		SAVST6	
		STOP	
		WRTPLT	
DIFALB	FLDATA	ALBRD	F11.3.18
DOTPRD	CRMAX		F11.3.35
	CRMIN		F11.3.36
	SRMAX		F11.3.35
	SRMIN		F11.3.36

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
DSQRT	EDITOR		**
	KEDIT		
	MATK		
	PLTKEF		
DTASET	IOWRT		M2.2.20
EDITOR	KEDIT	DSQRT	F11.3.57
EXP	FREAK		**
EXPRN	TRACK		M2.2.60
EXTRA	DATAIN		F11.3.7
FILLSG	NSUPG	CLEAR	F11.3.26
		FIL2D	F11.3.28
		RD	
		REED	
		RITE	
		RT	
		SGALB	
		SGWT	
FIL2D	FILLSG	RD	F11.3.28
FIND	ANGLES	Q	F11.3.22
FINDBX	LOCATE		F11.3.45
FISFLX	GUIDE	JSTIME	F11.3.40
		LOOPER	F11.3.42
		MATK	F11.3.55
		SQRT	
		STATIS	
FITFLX	MAS1ER	PRTFLX	F11.3.58
		RD	
FLDATA	DATAIN	CLEAR	F11.3.8
		DIFALB	F11.3.17
		REED	
		RGUSED	
		RITE	
		SORTA	
		SORTR	
		WATES	
FLTRN	ALBIN		M2.2.60
	CHOOSE		
	NSTART		
	START		
	START0		
	START1		
	START5		
	STRTSU		
	TRACK		

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
FREAD	MIXIT		M2.2.27
	PARAM		
	RDBIAS		
	RDORGN		
	RDPLOT		
	RDSTRT		
	XXIN		
FREAK	MASTER	EXP	F11.3.4
		SQRT	F11.3.6
GEOMIN	DATAIN	CLEAR	F11.3.7
		KENOG	F11.3.11
		READGM	
GETMUS	PRANG		F11.3.22
GETPTR	RCHRS		M2.2.64
GOCURS	VOLUME	HUNTER	F11.3.37
GTISO	SORTBK		M2.2.60
	START		
	STRTSU		
	TRACK		
GTVOLS	VOLUME		F11.3.37
GUIDE	MASTER	BANKER	F11.3.4
		CHKSTR	F11.3.5
		CLEAR	F11.3.40
		FISFLX	
		INDX	
		IOLEFT	
		JSTIME	
		NSTART	
		PULL	
		RD	
		RDCALC	
		REED	
		RESET	
		RITE	
		SQRT	
		START	
		TRACK	
		WRTCAL	
HLFWRD	ARAYIN		F11.3.9
HOLCHK	JOMCHK	ADJUST	F11.3.34
		HOLEXT	
		HOLHOL	
HOLE	LODARA	CLEAR	F11.3.31
	SORTA	STOP	

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
HOLEXT	HOLCHK	CRMAX	F11.3.34
		SRMAX	
		XXLIM	
HOLEOL	HOLCHK	CRMIN	F11.3.34
		SRMIN	
		XRMIN	
		XXMIN	
HUNTER	GOCURS	CLEAR	F11.3.37
	VOLUME	MOVE	
ICEMIX	MASTER	RDTAPE	F11.3.4
			F11.3.5
			F11.3.23
ICOMPA	RTADJ		M2.2.37
	START6		
IDX1D	DATAIN	IREAD	F11.3.7
INDX	GUIDE		F11.3.41
INITAL	MAIN	AREAD	F11.3.1
	Q#$009	IOLEFT	
		JSTIME	
		MESAGE	
		PARAM	
		SCANON	
IO	ALBUSE		M2.2.39
	MAKTAP		
	MIXMIX		
	RDALB		
	RDARA		
	RDBIAS		
	RDCALC		
	RDICE		
	RDPLOT		
	RDRST		
	RDSTRT		
	RDTAPE		
	RDWTS		
	SAVST6		
	WAITIN		
	WATES		
	WRTALB		
	WRTARA		
	WRTCAL		
	WRTICE		
	WRTPLT		
	WRTRST		
	WRTWTS		
IOLEFT	GUIDE		M2.2.41
	INITAL		
	IOSDUN		
IONUMS	Q#$009		M2.2.64

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
IOSDUN	JOMITY	IOLEFT	F11.3.4
	MASTER		F11.3.5
IOWRT	DATAIN	DTASET	F11.3.8
IREAD	ARAYIN		M2.2.42
	IDX1D		
	KENOG		
	MIXIT		
	PARAM		
	RDBIAS		
	RDBOX		
	RDPLOT		
	RDSTRT		
	XXIN		
IXALB	LIMLN		F11.3.26
	NSUPG		
JLL2	MIXER		F11.3.19
JOMCHK	JOMITY	CRMAX	F11.3.30
		HOLCHK	F11.3.33
		SRMAX	
		XXLIM	
JOMITY	MASTER	ARALBA	F11.3.4
		CORSIZ	F11.3.5
		IOSDUN	F11.3.29
		JOMCHK	
		LOADIT	
		PRTJOM	
		VOLUME	
JSTIME	FISFLX		M2.2.47
	GUIDE		
	INITAL		
	KEDIT		
	MASTER		
	MATRIX		
	START		
KEDIT	MASTER	DSQRT	F11.3.4
		EDITOR	F11.3.6
		JSTIME	F11.3.56
		LOOPER	
		MATRIX	
		PLTKEF	
		RNDOUT	
		SQRT	
KENOG	GEOMIN	AREAD	F11.3.11
		IREAD	
		MOVE	
		RCHRS	
		RDORGN	
		XXIN	
		XXINA	
LABL	MATRIX		M2.2.48

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
LDWRT	TRACK		F11.3.47
LEGEND	PRANG		F11.3.22
LIMLN	NSUPG	IXALB	F11.3.26
LISTQA	INITAL		M2.2
LOADIT	JOMITY	BOXC	F11.3.29
		LODALB	F11.3.31
		LODARA	
		PRTARA	
		REED	
LOCATE	MESH	FINDBX	F11.3.39
	START	LOCBOX	F11.3.45
		MOVE	
		POSIT	
LOCBOX	LOCATE		F11.3.39
			F11.3.45
	LODARA		F11.3.32
	SORTA		F11.3.17
	START		F11.3.45
	TRACK		F11.3.48
LODALB	LOADIT	REED	F11.3.32
LODARA	LOADIT	CLEAR	F11.3.31
		HOLE	
		LOCBOX	
		REED	
LODRGC	RGUSED		F11.3.17
LODWTS	MASTER	CLEAR	F11.3.4
		PRTWTS	F11.3.5
		RD	F11.3.38
		REED	
LOOPER	[illegible]	REED	F11.3.55
	[illegible]	RITE	F11.3.57
LREAD	[illegible]		M2.2.50
LSCAN	ARASIZ		F11.3.30
	RGUSED		F11.3.17
MAIN		ALOCAT	
		CLOSDA	
		INITAL	
		MASTER	
MAKANG	MIXER	CLEAR	F11.3.19
		PRANG	F11.3.21
		REED	
		STOP	
MAKTAP	MIXER	IO	F11.3.19
		REED	
		SCOOT	

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
MASTER	MAIN	CLEAR	F11.3.4
	Q#$009	CORRE	
		DATAIN	
		FITFLX	
		FREAK	
		GUIDE	
		ICEMIX	
		IOSDUN	
		JOMITY	
		JSTIME	
		KEDIT	
		LODWTS	
		MIXER	
		NSUPG	
		OPENDA	
		POINT	
		PRTPLT	
		STOP	
		WRTRST	
MATK	FISFLX	CLEAR	F11.3.55
	MATRIX	DSQRT	F11.3.57
MATRIX	KEDIT	JSTIME	F11.3.57
		LABL	
		MATK	
		SQRT	
MESAGE	INITAL		M2.2.51
MESH	PRINT	CLEAR	F11.3.39
		LOCATE	
MGCWRD	MIX2D		M2.2.52
MIXCRS	MIXER	CLEAR	F11.3.19
		STOP	
MIXER	MASTER	CLOSDA	F11.3.4
		JLL2	F11.3.5
		MAKANG	F11.3.19
		MAKTAP	
		MIXCRS	
		MIXMIX	
		OPENDA	
		REED	
		STOP	
		XLNTHS	
MIXIT	DATAIN	AREAD	F11.3.7
		FREAD	F11.3.12
		IREAD	
		LREAD	
		RDMIXT	
		RITE	
		STOP	

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
MIXMIX	MIXER	CMPRS	F11.3.19
		IO	F11.3.20
		MIX1D	
		MIX2D	
		MIX2M	
		NNITL	
		NORM1D	
		NORM2D	
		PRTMIX	
		REED	
		RITE	
		SUMSCT	
MIX1D	MIXMIX		F11.3.20
MIX2D	MIXMIX	MGCWRD	F11.3.20
MIX2M	MIXMIX		F11.3.20
MOVE	ADJUST		M2.2.53
	ARAYIN		
	BANKER		
	CHKSTR		
	HUNTER		
	KENOG		
	LOCATE		
	NSTART		
	RDPLOT		
	RESET		
	RTARA		
	SAVST6		
	SORTBK		
	START		
	START6		
	TRACK		
	VOLUME		
	WATES		
NNITL	MIXMIX	CLEAR	F11.3.20
		RITE	
NORM1D	MIXMIX		F11.3.21
NORM2D	MIXMIX		F11.3.21
NSTART	GUIDE	CLEAR	F11.3.40
		FLTRN	F11.3.42
		MOVE	F11.3.54
		SORTBK	
NSUPG	MASTER	FILLSG	F11.3.4
		IXALB	F11.3.5
		LIMLN	F11.3.26
		OPENDA	
		POINT	
		PRTXS	
		REED	
		RITE	
		RTADJ	
		STOP	

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
OPENDA	MASTER		M2.2.56
	MIXER		
	NSUPG		
PARAM	INITAL	AREAD	F11.3.2
		CLEAR	F11.3.3
		FREAD	
		IREAD	
		RNDIN	
		RNDOUT	
		TIMFAC	
		ZREAD	
PLTKEF	KEDIT	DSQRT	F11.3.57
POINT	MASTER		F11.3.4
	NSUPG		F11.3.5
POSIT	LOCATE		F11.3.39
			F11.3.45
PRANG	MAKANG	ANGLES	F11.3.22
		BADMOM	
		GETMUS	
		LEGEND	
		RD	
		RITE	
PRINT	PRTPLT	MESH	F11.3.39
PRTARA	LOADIT		F11.3.32
PRTFLX	FITFLX	SQRT	F11.3.58
PRTJOM	JOMITY		F11.3.30
PRTLBA	ARALBA	STOP	F11.3.30
PRTMIX	MIXMIX		F11.3.20
PRTPLT	MASTER	PRINT	F11.3.4
		REED	F11.3.5
		RELATE	F11.3.38
		UNTCRS	
PRTWTS	LODWTS		F11.3.38
PRTXS	NSUPG	PRT1D	F11.3.26
		PRT1DS	F11.3.27
		PRT2DS	
		REED	
PRT1D	PRTXS		M2.2.57
PRT1DS	PRTXS		F11.3.27
PRT2DS	PRTXS		F11.3.27
PULL	GUIDE		F11.3.40
			F11.3.42

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
Q	ANGLES		F11.3.22
	FIND		F11.3.22
Q#$009		ALOCAT	
		CLOSDA	
		INITAL	
		IONUMS	
		MASTER	
		SETBIN	
RATIO	CORRE		F11.3.25
RCHRS	ARAYIN	AREAD	F11.3.10
	KENOG	GETPTR	F11.3.11
	RDPLOT	RCRDLN	F11.3.13
		RSTPTR	
		STOP	
RCRDLN	RCHRS		M2.2.64
RD	FILLSG		M2.2.15
	FIL2D		
	FITFLX		
	GUIDE		
	LODWTS		
	PRANG		
	SGALB		
	SGWT		
RDALB	RDRST	CLEAR	F11.3.15
		IO	
		RITE	
RDARA	RDRST	IO	F11.3.16
		RITE	
RDBIAS	DATAIN	AREAD	F11.3.8
		FREAD	F11.3.14
		IO	
		IREAD	
		WAITIN	
RDBOX	ARAYIN	IREAD	F11.3.9
RDCALC	GUIDE	IO	F11.3.40
		RDGRP	F11.3.41
		RITE	F11.3.43
		RNDIN	
RDGRP	RDCALC		F11.3.43
RDICE	RDRST	CLEAR	F11.3.16
		IO	
		RITE	
RDMIXT	MIXIT		F11.3.12
RDORGN	KENOG	AREAD	F11.3.11
		FREAD	

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
RDPLOT	DATAIN	AREAD	F11.3.8
		CLEAR	F11.3.13
		FREAD	
		IO	
		IREAD	
		MOVE	
		RCHRS	
		SQRT	
RDREF	DATAIN	AREAD	F11.3.7
RDRST	DATAIN	IO	F11.3.8
		RDALB	F11.3.15
		RDARA	
		RDICE	
		RDWTS	
		RITE	
RDSTRT	DATAIN	AREAD	F11.3.8
		FREAD	
		IO	
		IREAD	
RDTAPE	ICEMIX	CLEAR	F11.3.23
		IO	
		RITE	
		SQRT	
		XSECID	
		XTENDA	
RDWTS	RDRST	IO	F11.3.16
		RITE	
READGM	GEOMIN		F11.3.11
REED	CORRE		M2.2.15
	FILLSC		
	FLDATA		
	GUIDE		
	LOADIT		
	LOALB		
	LODARA		
	LODWTS		
	LOOPER		
	MAKANG		
	MAKTAP		
	MIXER		
	MIXMIX		
	NSUPG		
	PRTPLT		
	PRTXS		
	SGALB		
	SORTA		
	START6		
	WATES		
	WRTALB		
	WRTARA		
	WRTCAL		
	WRTICE		
	WRTRST		
	WRTWTS		

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
RELATE	PRTPLT		F11.3.38
RESET	GUIDE	MOVE	F11.3.40 F11.3.41
RGUSED	FLDATA	CLEAR LODRGC LSCAN	F11.3.17
RITE	ALBUSE CORRE FILLSG FLDATA GUIDE LOOPER MIXIT MIXMIX NNITL NSUPG PRANG RDALB RDARA RDCALC RDICE RDRST RDTAPE RDWTS RT RTARA SAVST6 WAITIN WATES WRTPLT		M2.2.15
RNDIN	PARAM RDCALC		M2.2.60
RNDOUT	KEDIT PARAM TRKWRT WRTCAL		M2.2.60
RSTPTR	RCHRS		M2.2.64
RT	DATAIN FILLSG	RITE	F11.3.6 F11.3.28
RTADJ	NSUPG	ICOMPA	F11.3.26
RTARA	DATAIN	CLEAR MOVE RITE	F11.3.7
SAVST6	DATAIN	CLEAR IO MOVE RITE STOP	F11.3.8

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
SCANON	INITAL		M2.2.64
SCOOT	MAKTAP		F11.3.19
SETBIN	Q#$009		M2.2.64
SFLRA	STRTSU TRACK		M2.2.60
SGALB	FILLSG	RD REED	F11.3.28
SGWT	FILLSG	RD	F11.3.28
SORTA	FLDATA	CLEAR HOLE LOCBOX REED STOP	F11.3.17
SORTBK	NSTART	GTISO MOVE	F11.3.54
SORTR	FLDATA		F11.3.17
SQRT	CHKSTR CORSIZ CRMAX CRMIN CROS FISFLX FREAK GUIDE KEDIT MATRIX PRTFLX RDPLOT RDTAPE SRMAX SRMIN STRTSU TRACK VOLUME XRMIN XXLIM XXMIN		**
SRMAX	HOLEXT JOMCHK	CRSPRD DOTPRD SQRT STOP VECADD VECNRM	F11.3.34 F11.3.35
SRMIN	HOLHOL	CRSPRD DOTPRD SQRT STOP VECADD VECNRM	F11.3.34 F11.3.35

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
START	GUIDE	CLEAR	F11.3.40
		FLTRN	F11.3.40
		GTISO	F11.3.44
		JSTIME	
		LOCATE	
		LOCBOX	
		MOVE	
		START0	
		START1	
		START2	
		START3	
		START4	
		START5	
		START6	
		STOP	
		STRTSU	
		VOLFIS	
START0	START	FLTRN	F11.3.44
START1	START	ARSIN	F11.3.45
	START2	FLTRN	
START2	START	START1	F11.3.45
		START5	
START3	START		F11.3.45
START4	START	CHOOSE	F11.3.45
START5	START	CHOOSE	F11.3.45
	START2	FLTRN	
START6	START	ICOMPA	F11.3.45
		MOVE	
		REED	
STATIS	FISFLX		F11.3.56
STOP	ARAYIN		M2.2.67
	CRMAX		
	CRMIN		
	DATAIN		
	HOLE		
	MAKANG		
	MASTER		
	MIXCRS		
	MIXER		
	MIXIT		
	NSUPG		
	PRTLBA		
	RCHRS		
	SAVST6		
	SORTA		
	SRMAX		
	SRMIN		
	START		
	WATES		

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
STRTSU	START	ACOS	F11.3.44
		AZIRN	
		COS	
		FLTRN	
		GTISO	
		SFLRA	
		SQRT	
SUMSCT	MIXMIX		F11.3.21
TIMFAC	PARAM		M2.2.68
TRACK	GUIDE	ALBEDO	F11.3.40
		ALBIN	F11.3.41
		AZIRN	F11.3.47
		CROS	
		EXPRN	
		FLTRN	
		GTISO	
		LDWRT	
		LOCBOX	
		MOVE	
		SFLRA	
		SQRT	
		TRKWRT	
TRKWRT	CHKSTR	RNDOUT	F11.3.41
	TRACK		
UNTCRS	PRTPLT	CLEAR	F11.3.38
VECADD	CRMAX		F11.3.36
	CRMIN		
	SRMAX		
	SRMIN		
VECNRM	CRMAX		F11.3.36
	CRMIN		
	SRMAX		
	SRMIN		
VOLFIS	START		F11.3.44
VOLUME	JOMITY	ARSIN	F11.3.30
		CLEAR	F11.3.37
		GOCURS	
		GTVOLS	
		HUNTER	
		MOVE	
		SQRT	
WAITIN	RDBIAS	IO	F11.3.14
		RITE	
WATES	FLDATA	IO	F11.3.18
		MOVE	
		REED	
		RITE	
		STOP	

SUBROUTINE NAME	CALLING SUBROUTINE	CALLED SUBROUTINE	PAGE NUMBER
WRTALB	WRTRST	IO	F11.3.24
		REED	
WRTARA	WRTRST	IO	F11.3.24
		REED	
WRTCAL	GUIDE	IO	F11.3.40
		REED	F11.3.42
		RNDOUT	
		WRTGRP	
WRTGRP	WRTCAL		F11.3.42
WRTICE	WRTRST	IO	F11.3.24
		REED	
WRTPLT	DATAIN	IO	F11.3.8
		RITE	
WRTRST	MASTER	IO	F11.3.5
		REED	F11.3.24
		WRTALB	
		WRTARA	
		WRTICE	
		WRTWTS	
WRTWTS	WRTRST	IO	F11.3.24
		REED	
XLNTHS	MIXER		F11.3.19
XRMIN	HOLHOL	SQRT	F11.3.34
XSEC1D	RDTAPE		F11.3.23
XTENDA	RDTAPE		M2.2.72
XXIN	KENOG	FREAD	F11.3.11
		IREAD	
XXINA	KENOG		F11.3.11
XXLIM	HOLEXT	SQRT	F11.3.34
	JOMCHK		
XXMIN	HOLHOL	SQRT	F11.3.34
YREAD	ARAYIN		M2.2.73
ZREAD	PARAM		M2.2.76

APPENDIX B

F11.B ALPHABETICAL INDEX OF COMMONS

This section lists the labeled common blocks used in KENO V.a and an alphabetical listing of the subroutines that reference them.

COMMON NAME	SUBROUTINE
ALBNAM	ALBIN
	ALBRD
	ALBUSE
	CORRE
	DATAIN
	DIFALB
	FILLSG
	FLDATA
	GUIDE
	KEDIT
	LIMLN
	LOADIT
	MASTER
	NSUPG
	PARAM
	POINT
	RDALB
	RDREF
	RDRST
	SGALB
	TRACK
	WRTALB
	WRTRST
ANGLE	ANGLES
	BADMOM
	FIND
	GETMUS
	LEGEND
	PRANG
	Q
BLKLNC	MASTER
	PARAM
	RDTAPE

COMMON NAME	SUBROUTINE
DIMEN	ARASIZ
	ARAYIN
	BOX
	BOXC
	CORRE
	CORSIZ
	DATAIN
	EDITOR
	FILLSG
	FIL2D
	FINDBX
	FISFLX
	FITFLX
	FLDATA
	FREAK
	GEOMIN
	GUIDE
	HOLCHK
	ICEMIX
	INITAL
	JOMCHK
	JOMITY
	KEDIT
	KENOG
	LIMLN
	LOCATE
	LOCBOX
	LODARA
	LOOPER
	MASTER
	MATRIX
	MESH
	MIXER
	MIXIT
	NSTART
	NSUPG

COMMON NAME	SUBROUTINE
DIMEN(cont'd)	PARAM
	POINT
	POSIT
	PRINT
	PRTJOM
	PRTPLT
	RDBIAS
	RDBOX
	RDCALC
	RDGRP
	RDICE
	RDREF
	RDRST
	RDSTRT
	RDTAPE
	READGM
	RESET
	RGUSED
	RTARA
	SORTA
	SORTR
	START
	STATIS
	TRACK
	VOLFIS
	VOLUME
	WRTCAL
	WRTGRP
	WRTICE
	WRTRST
	XSEC1D
DRTACS	ALBUSE
	CORRE
	DATAIN
	FILLSG
	FIL2D
	FISFLX
	FITFLX
	FLDATA
	GUIDE
	ICEMIX
	INITAL
	LOADIT
	LODARA
	LODWTS

COMMON NAME	SUBROUTINE
DRTACS(cont'd)	LOOPER
	MASTER
	MIXER
	MIXIT
	NNITL
	NSUPG
	PRANG
	PRTPLT
	RDALB
	RDARA
	RDCALC
	RDICE
	RDRST
	RDTAPE
	RDWTS
	RT
	RTARA
	SAVST6
	SGALB
	SGWT
	SORTA
	START6
	WAITIN
	WATES
	WRTALB
	WRTCAL
	WRTICE
	WRTPLT
	WRTRST
	WRTWTS
FINAL	EDITOR
	FISFLX
	GUIDE
	INITAL
	KEDIT
	MASTER
	MATRIX
	START
	STATIS
	TRACK
	LOOPER

COMMON NAME	SUBROUTINE
LIFETM	EDITOR
	FISFLX
	GUIDE
	KEDIT
	MASTER
	RDCALC
	STATIS
	TRACK
	WRTCAL
LOGIC	ALBRD
	ARASIZ
	ARAYIN
	CHKSTR
	CORRE
	CORSIZ
	DATAIN
	EDITOR
	FILLSG
	FIL2D
	FISFLX
	FITFLX
	FLDATA
	GUIDE
	ICXMIX
	IDX1D
	INITAL
	JOMCHK
	JOMITY
	KEDIT
	KENOG
	LIMLN
	LOCATE
	LOOPER
	MASTER
	MATRIX
	MESH
	MIXER
	NSUPG

COMMON NAME	SUBROUTINE
LOGIC(cont'd)	PARAM
	POINT
	POSIT
	PRTJOM
	PRTPLT
	PRTXS
	RATIO
	RDBIAS
	RDBOX
	RDCALC
	RDGRP
	RDREF
	RDRST
	RDTAPE
	READGM
	RESET
	RTARA
	SORTA
	SORTR
	START
	STATIS
	TRACK
	VOLFIS
	VOLUME
	WATES
	WRTCAL
	WRTGRP
LPNT	MASTER
	START6
MATRX	FLDATA
	MATRIX
	NSUPG
	RESET
	SORTA
	TRACK
	PARAM
	POINT

COMMON NAME	SUBROUTINE
NUTRON	ALBIN
	CHKSTR
	CHOOSE
	CROS
	FINDBX
	FLDATA
	GUIDE
	LOCATE
	MESH
	NSTART
	RESET
	START
	START0
	START1
	START2
	START3
	START4
	START5
	START6
	STRTSU
	TRACK
	TRKWRT
PICT	MESH
	PRINT
	PRTPLT
	RDPLOT
	RELATE
POINTR	DATAIN
	FILLSG
	ICEMIX
	JOMITY
	KEDIT
	LOADIT
	MASTER
	NSUPG
	POINT
	RDCALC
	WRTCAL
RUNTYP	DATAIN
	INITAL
	MAIN
	PARAM
	Q#$009

COMMON NAME	SUBROUTINE
STDATA	CHKSTR
	DATAIN
	RDRST
	RDSTRT
	START
	STAPT0
	START1
	START2
	START3
	START4
	START5
	START6
	VOLFIS
	WRTRST
TITL	ARASIZ
	CORSIZ
	DATAIN
	EDITOR
	FITFLX
	FREAK
	GEOMIN
	GUIDE
	HOLCHK
	HOLEXT
	HOLHOL
	ICEMIX
	INITAL
	JOMCHK
	KEDIT
	KENOG
	MASTER
	MATRIX
	NSUPG
	PARAM
	POINT
	PRTJOM
	PRTLBA
	PRTMIX
	PRTPLT
	PRTWTS
	PRTXS
	RDBOX
	RDPLOT
	RDTAPE
	READGM
	TRACK
	VOLUME

COMMON NAME	SUBROUTINE
UNIT	ALBIN
	ALBRD
	ALBUSE
	ANGLES
	ARASIZ
	ARAYIN
	BOX
	CHKSTR
	CORRE
	CORSIZ
	CROS
	DATAIN
	DIFALB
	EDITOR
	FILLSG
	FIL2D
	FIND
	FISFLX
	FITFLX
	FLDATA
	FREAK
	GEOMIN
	GETMUS
	GUIDE
	ICEMIX
	IDX1D
	INDX
	INITAL
	JOMCHK
	JOMITY
	KEDIT
	KENOG
	LDWRT
	LIMLN
	LOADIT
	LODARA
	LODWTS
	LOOPER
	MAIN
	MAKANG
	MAKTAP
	MASTER
	MATK
	MATRIX
	MIXER
	MIXIT
	MIXMIX
	NNITL
	NSTART
	NSUPG

COMMON NAME	SUBROUTINE
UNIT(cont'd)	PARAM
	POINT
	POSIT
	PRANG
	PRINT
	PRTARA
	PRTFLX
	PRTJOM
	PRTLBA
	PRTPLT
	PRTXS
	Q#$009
	RATIO
	RDALB
	RDBIAS
	RDBOX
	RDCALC
	RDGRP
	RDICE
	RDPLOT
	RDREF
	RDRST
	RDSTRT
	RDTAPE
	RDWTS
	READGM
	RT
	RTARA
	SAVST6
	SGALB
	SGWT
	SORTA
	SORTR
	START
	START2
	START6
	STATIS
	STRTSU
	TRACK
	TRKWRT
	VOLFIS
	VOLUME
	WAITIN
	WATES
	WRTALB
	WRTCAL
	WRTGRP
	WRTICE
	WRTRST
	WRTWTS

APPENDIX C

F11.C KENO-V INPUT SUMMARY

This appendix consists of a summary of the KENO V input data requirements.

Table F11.C.1. Summary of parameter data

TITLE The title must be entered first (80 columns) See Section F11.4.3

PARAMETERS Format: READ PARAM enter parameter data here END PARAM
If parameters are entered, they must follow the title. See Sections F11.4.3, F11.5.2 and F11.5.3.

KEY	STD.	DEFINITION
RND=	given	random number
TME=	30.min	execution time(min)
TBA=	0.5min	batch time (min)
WTA=	0.5	average weight
WTH=	3.0	wt. for splitting
WTL=	1/WTH	Russian Roulette wt
GEN=	103	No. of generations
NPG=	300	No. per generation
NSK=	3	generations skipped
RES=	0	gens. between restart
NBK=	GEN+25	neutron bank positions
XNB=	0	extra bank entries
NFB=	NPG	fission bank positions
XFB=	0	extra bank entries
X1D=	0	No. of extra 1-D's
LNG=	4000000	words of storage
BEG=	1	restart at this gen.
NB8=	200	blocks for d.a. unit
NL8=	789	length of d.a. block

KEY	STD.	DEFINITION
RUN=	YES	execute problem
FLX=	NO	fluxes
FDN=	NO	fission densities
ADJ=	NO	adjoint calculation
AMX=	NO	all mixture xsecs
XAP=	NO	xsec angles & probs.
XS1=	NO	1-D xsecs
XS2=	NO	2-D xsecs
PKI=	NO	fission spectrum
P1D=	NO	extra 1-D xsecs
FAR=	NO	fiss & abs.
MKP=	NO	matrix by location
CKP=	NO	cofactor k by loc.
FMP=	NO	fiss. prod. by loc.
MKU=	NO	matrix by unit
CKU=	NO	cofactor k by unit
FMU=	NO	fiss. prod. by unit

KEY	STD.	DEFINITION
MKH=	NO	matrix by hole
CKH=	NO	cofactor k by hole
FMH=	NO	fiss. prod. by hole
MHL=	NO	MKH at highest level
MKA=	NO	matrix by array
CKA=	NO	cofactor k by array
FMA=	NO	fiss. prob. by array
MAL=	NO	MKA at higest level
PLT=	YES	printer plots tracks
BUG=	NO	debug print
TRK=	NO	print neutron tracks
PWT=	NO	print avg. weight
PGM=	NO	unprocessed geometry
SMU=	NO	self-multiplication
NUB=	NO	neutrons per fission
PAX=	NO	albedo-xsec array

KEY	STD.	DEFINITION
XSC=	16	mixed xsecs
ALB=	79	albedo
WTS=	80	weights
LIB=	0	working xsecs
SKT=	16	scratch
RST=	0	read restart
WRS=	0	write restart

Table F11.C.2. Summary of array data

ARRAY

Format: READ ARRAY array parameters data type orientation data END ARRAY
See Sections F11.5.5, F11.5.6 and F11.5.7

Repeat the sequence ARRAY PARAMETERS DATA TYPE ORIENTATION DATA for each array used in the problem.

ARRAY PARAMETERS

KEYWORD	DEFAULT	DEFINITION
ARA=	1	No. defining the array
NUX=	1	No. of units in X direction
NUY=	1	No. of units in Y direction
NUZ=	1	No. of units in Z direction
GBL=	maxara	The global, universe or overall array number**
COM=	none	delim comment delim optional comment is a maximum of 132 characters.

** Can be defaulted by the code. If specified, it need be entered only once per problem.

DATA TYPE

FILL
LOOP

ORIENTATION DATA FOR FILL

Enter unit numbers to define every position in the array. When entering data utilizing the options in this table, the count field and option field must be adjacent with no imbedded blanks. The operand field may be separated from the option field by one or more blanks. Fill data for an array ends with END FILL.

COUNT FIELD	OPTION FIELD	OPERAND FIELD		COMMENTS
		j		stores j at the current position in the array
i	R	j		stores j in the next i positions in the array
i	*	j		stores j in the next i positions in the array
i	$	j		stores j in the next i positions in the array
	F	j		fills remainder of the array with unit No. j starting with the current array position
	A	j		sets the current position in the array to j
i	S			increments current position in the array by i allows skipping i positions. The value of i may be positive or negative.
i	Q	j		repeats the previous j entries i times. The default value of i is 1.
i	M	j		repeats the previous j entries i times, inverting the sequence each time. The default value of i is 1.
i	B	j		Starting with the entry at -i from the current position, store entries in inverse order until position -(i+j) is reached. Default value of i=1.
i	P	j		alternately stores j and -j in the next i positions of the array.
i	I	j	k	provides i entries linearly interpolated between j and k, and the end points (i.e. a total of i+2 points.) j and k must be separated by at least one blank.
	T			terminates the data reading for the array.

ORIENTATION DATA FOR LOOP

Enter the unit number and nine numbers that define the position(s) of that unit. Data for each of these ten entries are repeated until every position in the array has been defined. Data for an array is terminated by entering END LOOP.

ENTER DATA IN THE FORM:

DATA ENTRY	COMMENTS
LTYPE	The unit or box type. LTYPE must be greater than 1.
IX1	Starting position in the X direction. IX1 must be at least 1 and no larger than the value entered for NUX.
IX2	Ending position in the X direction. IX2 must be at least 1 and no larger than the value of NUX.
INCX	The number of units by which increments are made in the X direction.
IY1	The starting position in the Y direction. IY1 must be at least 1 and less than the value entered for NUY.
IY2	Ending position in the Y direction. IY2 must be at least 1 and no larger than the value of NUY.
INCY	The number of units by which increments are made in the positive Y direction.
IZ1	Starting position in the Z direction. IZ1 must be at least 1 and no larger than NUZ. in the positive Y direction.
IZ2	Ending position in the Z direction. IZ2 must be at least 1 and no larger than NUZ.
INCZ	The number of units by which increments are made in the positive Z direction.

Table F11.C.3. Summary of biasing data

BIAS (weighting)

Format: READ BIAS keyword correlation data auxiliary END BIAS
See Sections F11.4.7 and F11.5.8

KEYWORD	DESCRIPTION
ID=	CORRELATION DATA will be read next.
	id material ID. Enter from table at right. to use weighting data from the library.
	ibgn beginning bias ID
	iend ending bias ID
WT=	AUXILIARY DATA will be read next.
WTS=	AUXILIARY DATA will be read next.
	wttitl material title
	iq material ID.
	nsets number of sets of group structures
	REPEAT THKINC, NUMINC, NGPWT, WTAVG NSETS TIMES
	thkinc thickness per increment
	numinc number of increments
	ngpwt number of energy groups for this set of wts
	wtavg enter numinc* ngpwt values of wtavg.

MATERIAL	ID	ENERGY GROUPS	THICKNESS/INCREMENT
concrete	301	16, 27, 123	5 cm
paraffin	400	16, 27, 123	3 cm
water	500	16, 27, 123, 218	3 cm
graphite	6100	16, 27, 123	20 cm

For CORRELATION DATA, the Material ID is chosen from material ID column above (the keyword is ID=).
For AUXILIARY DATA, the material ID is chosen by the user and the keyword is WT= or WTS=.

Beginning and ending bias ID's are defined by the user. The geometry specification that has the bias ID equal to the beginning bias ID utilizes the wt avg's from the first interval of material ID.

Table F11.C.4. Summary of boundary condition data

BNDS (albedo or boundary conditions)

Format: READ BNDS face code albedo name END BNDS
See Section F11.4.7

The sequence FACE CODE ALBEDO NAME is entered as many times as necessary to define the appropriate albedo boundary conditions. The default for all faces is vacuum.

FACE CODES FOR ENTERING BOUNDARY (ALBEDO) CONDITIONS

FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION	FACE CODE	DEFINITION
+XB=	positive X face	XFC=	both X faces	+YX=	positive X and Y faces	&ZY=	positive Y and Z faces
&XB=	positive X face	YFC=	both Y faces	&XY=	positive X and Y faces	-XY=	negative X and Y faces
-XB=	negative X face	ZFC=	both Z faces	&YX=	positive X and Y faces	-XZ=	negative X and Z faces
+YB=	positive Y face	+FC=	all positive faces	+XZ=	positive X and Z faces	-YZ=	negative Y and Z faces
&YB=	positive Y face	&FC=	all positive faces	+ZX=	positive X and Z faces	YXF=	all X and Y faces
-YB=	negative Y face	-FC=	all negative faces	&XZ=	positive X and Z faces	ZXF=	all X and Z faces
+ZB=	positive Z face	XYF=	all X and Y faces	&ZX=	positive X and Z faces	ZYF=	all Y and Z faces
&ZB=	positive Z face	XZF=	all X and Z faces	+YZ=	positive Y and Z faces	-YX=	negative X and Y faces
-ZB=	negative Z face	YZF=	all Y and Z faces	+ZY=	positive Y and Z faces	-ZX=	negative X and Z faces
ALL=	all 6 faces	+XY=	positive X and Y faces	&YZ=	positive Y and Z faces	-ZY=	negative Y and Z faces

ALBEDO NAMES AVAILABLE ON THE KENO V ALBEDO LIBRARY, FOR USE WITH THE FACE CODES

ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION
DP0H2O	12 inch double P0 water	CONC-4	4 inch concrete differential	VACUUM	vacuum condition
DP0H2O	differential albedo with	CON4	albedo with 4 incident	VOID	
DP0	4 incident angles	CONC4	angles	VACU	
DPO				VAC	
		CONC-8	8 inch concrete differential		
H2O	12 inch water differential	CON8	albedo with 4 incident	SPECULAR	mirror image reflection
WATER	albedo with 4 incident	CONC8	angles	MIRROR	
	angles			MIRR	
				SPEC	
PARAFFIN	12 inch paraffin differen-	CONC-12	12 inch concrete differential	SPE	
PARA	tial albedo with 4 incident	CON12	albedo with 4 incident	MIR	
WAX	angles	CONC12	angles		
CARBON	200 cm. carbon differential	CONC-16	16 inch concrete differential	PERIODIC	periodic boundary condition
GRAPHITE	albedo with 4 incident angles	CON16	albedo with 4 incident	PERI	
C		CONC16	angles	PER	
ETHYLENE	12 inch polyethylene	CONC-24	24 inch concrete differential		
POLY	differential albedo with	CON24	albedo with 4 incident		
CH2	4 incident angles	CONC24	angles		

Table F11.C.5. Summary of geometry data

GEOMETRY (region)

Format: READ GEOM enter geometry region data here END GEOM
See Sections F11.4.4, F11.5.1.2, F11.5.6 and F11.5.7

GEOMETRY REGION DATA consists of SIMPLE GEOMETRY REGION DATA and EXTENDED GEOMETRY REGION DATA.

ENTER GEOMETRY REGION DATA IN THE FOLLOWING FORM:

```
OPTIONAL GLOBAL SPECIFICATION
UNIT n
OPTIONAL GEOMETRY COMMENT
SIMPLE GEOMETRY REGION DATA
  and/or
EXTENDED GEOMETRY REGION DATA
```

ENTER SIMPLE GEOMETRY REGION DATA IN THE FOLLOWING FORM:

```
GLOBAL   Enter only to specify this unit as the global unit.
UNIT n
COM-delim comment delim  This optional comment can be up to 132 characters.
                         It must begin and end with a delimiter.
fgeom  mix no.  bias ID  dimensions optional origin data (ORIGN coordinates) optional chord data (CHORD)
```

Enter as many geometry description specifications as necessary to describe the unit and as many units as necessary to describe the system.

SIMPLE GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA	TYPE 4 DATA	TYPE 5 DATA	TYPE 6 DATA
fgeom	SPHERE HEMISPHERE HEMISPHERE+X HEMISPHERE-X HEMISPHERE+Y HEMISPHERE-Y HEMISPHERE+Z HEMISPHERE-Z	XCYLINDER XHEMICYL+Y XHEMICYL-Y XHEMICYL+Z XHEMICYL-Z	YCYLINDER YHEMICYL+X YHEMICYL-X YHEMICYL+Z YHEMICYL-Z	CYLINDER ZCYLINDER ZHEMICYL+X ZHEMICYL-X ZHEMICYL+Y ZHEMICYL-Y	CUBE	CUBOID
dimensions	R (radius)	R +H -H	R +H -H	R +H -H	+X -X	+X -X +Y -Y +Z -Z
optional origin coord.	Enter the X Y Z coord. of origin	Enter the Y Z coord. of center-line	Enter the X Z coord. of center-line	Enter the X Y coord. of center-line	omit	omit
optional chord data**	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	Enter the dist. to plane	omit	omit

**Chord data is not applicable for SPHERE XCYLINDER YCYLINDER CYLINDER or ZCYLINDER

Table F11.C.5 (continued)

GEOMETRY (region) (cont.)

ENTER EXTENDED GEOMETRY DATA IN THE FOLLOWING FORM:

fgeom ref. ID bias ID thickness per region origin coordinates nreg

EXTENDED GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	TYPE 1 DATA	TYPE 2 DATA	TYPE 3 DATA
fgeom	ARRAY CORE COREBDY COREBNDS COREBOUN	HOLE	REPLICATE REFLECTOR
ref. ID	array no.	emplaced unit number	mixture no. in generated regions
bias ID	omit for ARRAY	omit	first bias ID
thick./reg.	omit	omit	variable**
origin coord.	Enter the X Y Z coord. of most neg. pt. of array	Enter the X Y Z coord of origin	omit
nreg	omit	omit	No. of regions to be generated

**The number of dimensions to be entered is the same as the region preceding the replicate or reflector specification because the generated regions have that shape. The value of the dimensions is the thickness of each generated region of material on that surface.

Table F11.C.6. Summary of mixing table data

MIXTURES

Format: READ MIXT xsec parameters mixing table END MIXT
This data is entered only if an AMPX working format library is being used. (LIB=) in the parameter data. Do not enter if an ICE mixed library is used, (XSC=) in the parameter data.
See Sections F11.4.10 and F11.5.5

XSEC PARAMETERS consists of keywords and associated values. These parameters, if entered, need be entered only once.

KEYWORD	DEFAULT	DEFINITION
SCT=	1	No. of discrete scattering angles. 0 is isotropic 1 is P1 2 is P3 3 is P5
EPS=	.00003	Cross section message cutoff value. Use to suppress message No. K5-60

MIXING TABLE DATA consists of (1) a keyword and mixture ID for the mixture
The keyword is MIX=
The desired mixture number follows the keyword.
(2) nuclide ID**
(3) number density**
** the sequence (2) (3) is repeated for each nuclide in the mixture.

REPEAT the sequence (1) (2)'s (3)'s until all the mixtures have been described.

Table F11.C.7. Summary of plot data

PLOT Format: READ PLOT plot parameters END PLOT
plot parameters must be entered for each plot that is to be made.
See Sections F11.4.11 and F11.5.9

KEYWORD	DEFAULT	DEFINITION
TTL=	prob. title	delim ptitl delim A one character delimiter signals the beginning and end of the title. ptitl is the plot title.
PIC=	MAT	Type of picture: MIXTURE, UNIT NO. or BIAS ID NO.
		MIXTURE ------- MAT
		MIX
		MIXT
		MIXTURE
		MEDI
		MEDIA
		UNIT NO. ------- BOX
		BOXT
		BOXTYPE
		UNT
		UNIT
		UNITTYPE
		BIAS ID NO. ---- IMP
		BIAS
		BIASID
		WTS
		WEIG
		WEIGHTS
		WGT
		WGTS
XUL=	prev. plot	X coord. of upper left corner of plot
YUL=	prev. plot	Y coord. of upper left corner of plot
ZUL=	prev. plot	Z coord. of upper left corner of plot
XLR=	prev. plot	X coord. of lower right corner of plot
YLR=	prev. plot	Y coord. of lower right corner of plot
ZLR=	prev. plot	Z coord. of lower right corner of plot

KEYWORD	DEFAULT	DEFINITION
UAX=	prev. plot 0 IF VAX OR WAX is read	X component of direction cosine for the AX axis of the plot (across)
VAX=	prev. plot 0 IF UAX OR WAX is read	Y component of direction cosine for the AX axis of the plot (across)
WAX=	prev. plot 0 IF UAX OR VAX is read	Z component of direction cosine for the AX axis of the plot (across)
UDN=	prev. plot 0 IF VDN OR WDN is read	X component of direction cosine for the DN axis of the plot (down)
VDN=	prev. plot 0 IF UDN OR WDN is read	Y component of direction cosine for the DN axis of the plot (down)
WDN=	prev. plot 0 IF UDN OR VDN is read	Z component of direction cosine for the DN axis of the plot (down)
DLX=		horizontal spacing between points on plot
DLD=		vertical spacing between points on plot
MAX=		no. of intervals to be printed across page
MDN=		no. of intervals to be printed down page
LPI=	8	lines per inch printed down the page
NCH=	CHRS**	delim CHRS delim a one character delimiter signals the beginning and end of the character string.
RUN=	YES	YES allows the problem to execute NO terminates problem after data checking
PLT=	YES	YES allows the plot(s) to be made NO allows reading the plot data without making a plot.

PLOT ORIGIN:

(1) SINGLE UNIT-coincides with origin of geometry description.

(2) BARE ARRAY-at the most negative point of the array, (lower left-hand back corner of the global array).

(3) REFLECTED ARRAY- coincides with the origin of the CORE or ARRAY description of the global array.

**default values of CHRS are given below.

MEDIA	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
CHRS		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M
MEDIA	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42			
CHRS	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	#	,	$	-	+	)	(			
MEDIA	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58							
CHRS	&	>	:	;	.	-	%	*	"	=	!	(	@	<	/	0							

Table F11.C.8. Summary of starting data

START

Format: READ START enter start data here END START
The default value of start type is zero.
See Section F11.4.8

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
0	none	NST	uniform
		XSM	
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
1	NST	XSM	cosine
		XSP	
		YSM	
		YSP	
		ZSM	
		ZSP	
		RFL	
		PSP	
2	NST	XSM	cosine with
	NXS	XSP	fraction in
	NYS	YSM	specified
	NZS	YSP	unit
	FCT	ZSM	
		ZSP	
		RFL	
		PSP	

START TYPE	REQUIRED DATA	OPTIONAL DATA	STARTING DISTRIBUTION
3	NST	KFS	spike
	TFX	PSP	
	TFY		
	TFZ		
	NXS		
	NYS		
	NZS		
4	NST	KFS	multiple
	TFX	PSP	spikes
	TFY		
	TFZ		
	NBX		
5	NST	PSP	in specified
	NBX		units
6	NST	NXS	arbitrary
	TFX	NYS	points
	TFY	NZS	
	TFZ	KFS	
	LNU*	PS6	
		PSP	

*LNU must be the last entry for each set of start 6 data. The LNU of each successive set of data must be larger than the last.

KEYWORD	DEFAULT	DEFINITION
NST=	0	start type
TFX=	0.0	X coordinate
TFY=	0.0	Y coordinate
TFZ=	0.0	Z coordinate
NXS=	0	X index of unit pos.
NYS=	0	Y index of unit pos.
NZS=	0	Z index of unit pos.
KFS=		fission spectra
LNU=	0	number of last neutron
NBX=	0	source unit number
FCT=	0	fraction
XSM=	-X	-X of source cuboid
XSP=	+X	+X of source cuboid
YSM=	-Y	-Y of source cuboid
YSP=	+Y	+Y of source cuboid
ZSM=	-Z	-Z of source cuboid
ZSP=	+Z	+Z of source cuboid
RFL=	NO	start in reflector
PS6=	NO	print start 6 input
PSP=	NO	print starting points

APPENDIX D

F11.D SAMPLE PROBLEMS

This section contains sample problems to demonstrate some of the options available in KENO V. All the options of KENO IV except generalized geometry and the search capability are available in KENO V. A brief problem description and the associated card input data are included for each sample problem. Different options may be easily activated by making changes in the data. These problems are set up using the nuclide identifiers from the Hansen-Roach 16-group AMPX working format library which is specified on Unit 41 of the job control language. The nuclide identifiers for this library are listed in Table F11.D.1 at the end of this section. The unit number is defined by the parameter LIB= in the parameter data.

SAMPLE PROBLEM 1 2C8 BARE

This is a simple 2x2x2 array of uranium metal cylinders as described in the article, "Critical Three-Dimensional Arrays of U(93.2)-Metal Cylinders,"[1] by J. T. Thomas. This critical experiment is designated in Table II of that article as cylinder index 11 and reflector index 1. Figure F11.D.1 shows the critical experiment.

CARD INPUT

```
SAMPLE PROBLEM 1 CASE 2C8 BARE
READ PARAMETERS TME=3.0 FLX=YES FDN=YES FAR=YES LIB=41 END PARAMETERS
READ MIXT SCT=1 MIX=1 92500 4.48006-2 92800 2.6578-3 92400 4.827-4
92600 9.57-5 END MIXT
READ GEOMETRY
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID 0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
END GEOMETRY
READ ARRAY NUX=2 NUY=2 NUZ=2 END ARRAY
END DATA
```

SAMPLE PROBLEM 2 CASE 2C8 BARE WITH 8 UNIT TYPES MATRIX CALCULATION

This problem is the same as sample problem 1 except it is set up as a mixed box problem with each unit of the array defined as a different unit type. Matrix k-effectives will be calculated for this problem by both unit type and array position. The print flags are set to print all matrix data.

CARD INPUT

```
SAMPLE PROBLEM 2  2C8 BARE WITH 8 UNIT TYPES MATRIX CALCULATION
READ PARAM TME=3.0 LIB=41 FLX=YES FDN=YES
MKU=YES FMU=YES MKP=YES FMP=YES
END PARAM  READ GEOMETRY  UNIT 1
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
UNIT 2
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
UNIT 3  CYLINDER  1 1 5.748 5.3825 -5.3825 CUBOID  0 1 6.87 -6.87 6.87
-6.87 6.505 -6.505
BOX TYPE 4  CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
BOX TYPE  5
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
UNIT 6
CYLINDER  1 1 5.748 5.3825 -5.3825  CUBOID  0 1 6.87 -6.87 6.87 -6.87
6.505 -6.505 UNIT 7  CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
UNIT 8
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505 END GEOM
READ MIXT  SCT=1  MIX=1 92500 4.48006-2  92800 2.6578-3  92400 4.827-4
92600 9.57-5  END MIXT
READ ARRAY  NUX=2 NUY=2 NUZ=2  LOOP  10*1 3*2 7*1  3 1 1 1 2 2 1
1 1 1  4 2 2 1 2 2 1 1 1 1  5 6*1 2 2 1  6 2 2 1 1 1 1 2 2 1
7  1 1 1 2 2 1 2 2 1  8 2 2 1 2 2 1 2 2 1 END ARRAY
END DATA
```

SAMPLE PROBLEM 3 2C8 15.24 CM PARAFFIN REFL

A 2x2x2 array of uranium metal cylinders is reflected by 6 in. of paraffin on all faces. This critical experiment[1] is designated as cylinder index 11 and reflector index 5 in Table II of Ref. 1. Figure F11.D.2 shows half of the critical experiment, which consisted of the half shown and the mirror image of it. These two assemblies were moved together to achieve criticality. The top reflector is missing in Fig. F11.D.2 but was present when criticality was achieved.

CARD INPUT

```
SAMPLE PROBLEM 3  2C8  15.24 CM PARAFFIN REFL
READ PARAM  LIB=41 FLX=YES FDN=YES TME=10.0  PWT=YES END PARAM
READ ARRAY NUX=2 NUY=2 NUZ=2 END ARRAY
READ MIX  SCT=1  MIX=1 92500 4.48006-2 92800 2.6578-3  92400 4.827-4
92600 9.57-5  MIX=2  1101 8.2581-2  6100 3.9702-2 END MIXT
READ GEOM
CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 11.74 -11.74 11.74 -11.74 11.375 -11.375
CORE  0 1 -23.48 -23.48 -22.75
CUBOID  2 2 26.48 -26.48 26.48 -26.48 25.75 -25.75
CUBOID  2 3 29.48 -29.48 29.48 -29.48 28.75 -28.75
CUBOID  2 4 32.48 -32.48 32.48 -32.48 31.75 -31.75
CUBOID  2 5 35.48 -35.48 35.48 -35.48 34.75 -34.75
CUBOID  2 6 38.72 -38.72 38.72 -38.72 37.99 -37.99
END GEOM
READ BIAS  ID=400 2 6  END BIAS  END DATA
```

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

This problem is the same as sample problem 3 except it is set up using the automatic reflector option instead of describing the reflector manually.

CARD INPUT

```
SAMPLE PROBLEM 4  2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL
READ PARAM TME=10.0 PWT=YES LIB=41 FLX=YES FDN=YES END PARAM
READ GEOMETRY  CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 11.74 -11.74 11.74 -11.74 11.375 -11.375
CORE  0 1 -23.48 -23.48 -22.75
REFLECTOR  2 2 6*3.0 5
REFLECTOR  2 7 6*.24 1
END GEOM
READ MIXT  MIX=1 92500 4.48006-2  92800 2.6578-3  92400 4.827-4
92600 9.57-5  MIX=2  1101 8.2581-2  6100 3.9702-2  SCT=1 END MIXT
READ ARRA  NUX=2 NUY=2 NUZ=2 END ARRAY
READ BIAS  ID=400 2 7  END BIAS
END DATA
```

SAMPLE PROBLEM 5 2C8 12 INCH PARAFFIN ALBEDO REFLECTOR

This problem is the same as sample problems 3 and 4 except the reflector is represented by a 12-in. paraffin albedo. Note the decrease in execution time when using an albedo reflector instead of doing actual tracking. Note also that k-effective is somewhat higher for this system, probably due to the small edge size of the system.[2]

CARD INPUT

```
SAMPLE PROBLEM 5  2C8 12 INCH PARAFFIN ALBEDO REFLECTOR
READ PARA
FLX=YES FAR=YES FDN=YES TME=5.0  LIB=41 END PARA
READ ARRAY  NUX=2 NUY=2 NUZ=2 END ARRAY
READ MIXT  SCT=1 MIX=1 92500 4.48006-2 92800 2.6578-3  92400 4.827-4
92600 9.57-5  END MIXT
READ BOUNDS  ALL=PARAFFIN  END BOUNDS
READ GEOM  CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 11.74 -11.74 11.74 -11.74 11.375 -11.375  END GEOM
END DATA
```

SAMPLE PROBLEM 6 ONE 2C8 UNIT (SINGLE UNIT)

One of the 2C units is described and run as a single-unit problem, and its k-effective is calculated.

CARD INPUT

```
SAMPLE PROBLEM 6  ONE 2C8 UNIT (SINGLE UNIT)
READ PARA LIB=41 FLX=YES FDN=YES FAR=YES TME=3.0  END PARA
READ MIXT  SCT=1  MIX=1  92600 9.57-5  92400 4.827-4  92800 2.6578-3
92500 4.48006-2  END MIXT
READ GEOMETRY  CYLINDER  1 1 5.748 5.3825 -5.3825
END GEOMETRY  END DATA
```

SAMPLE PROBLEM 7 BARE 2C8 USING SPECULAR REFLECTION

One of the 2C8 units[1] is described and the 2x2x2 array is simulated by using specular reflection on the positive x, y and z faces of the unit. This is a simulation of sample problem 1.

CARD INPUT

```
SAMPLE PROBLEM 7  BARE 2C8 USING SPECULAR REFLECTION
READ PARA  LIB=41  FLX=YES FDN=YES  FAR=YES  TME=3.0 END PARAMETERS
READ MIX  SCT=1  MIX=1  92800 2.6578-3  92500 4.48006-2  92600 9.57-5
92400 4.827-4 END MIX
READ GEOM  CYLINDER  1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
END GEOM
READ ARRAY END ARRAY
READ BOUNDS  +FC=SPECULAR END BOUNDS  END DATA
```

SAMPLE PROBLEM 8 INFINITELY LONG CYLINDER FROM 2C8 UNIT

The fuel and cylinder radius from sample problem 1 is used. The length of the cylinder is arbitrarily chosen to be 20 cm and the unit is specularly reflected on the top and bottom to create an infinitely long cylinder.

CARD INPUT

```
SAMPLE PROBLEM 8 INFINITELY LONG CYLINDER FROM 2C8 UNIT
READ PARAM LIB=41 TME=3.0 END PARAM
READ MIXT SCT=1 MIX=1 92800 2.6578-3 92600 9.57-5 92500 4.48006-2
92400 4.827-4 END MIXTURES
READ GEOMETRY CYLINDER 1 1 5.748 10.0 -10.0
CUBOID 0 1 6.87 -6.87 6.87 -6.87 10.0 -10.0
END GEOMETRY
READ BOUNDS ZFC=MIRROR END BOUNDS
READ ARRAY END ARRAY
END DATA
```

SAMPLE PROBLEM 9 INFINITE ARRAY OF 2C8 UNITS

The geometry description from sample problem 1 is used and the cuboid is specularly reflected on all faces to create an infinite array of 2C8 units having an edge-to-edge spacing of 2.244 cm in the x and y directions and 2.245 cm in the z direction.

CARD INPUT

```
SAMPLE PROBLEM 9 INFINITE ARRAY OF 2C8 UNITS
READ PARAM TME=5.0 LIB=41 END PARAM
READ MIXTURES SCT=1 MIX=1 92500 4.48006-2 92800 2.6578-3
92600 9.57-5 92400 4.827-4
END MIXT
READ ARRAY END ARRAY
READ BOUN ALL=MIR END BOUN
READ GEOM
CYLINDER 1 1 5.748 5.3825 -5.3825 CUBOID 0 1 6.87 -6.87 6.87 -6.87
6.505 -6.505 END GEOM END DATA
```

SAMPLE PROBLEM 10 2C8 BARE WRITE RESTART

This problem is the same as sample problem 1, a 2x2x2 array of metal cylinders. Restart information is written on Unit 95 after the completion of every fifth generation.

CARD INPUT

```
SAMPLE PROBLEM 10  CASE 2C8 BARE  WRITE RESTART
READ PARAMETERS  TME=2.0 FLX=YES FDN=YES FAR=YES LIB=41 RES=5 WRS=95
END PARAMETERS
READ MIXT  SCT=1  MIX=1 92500 4.48006-2  92800 2.6578-3  92400 4.827-4
92600 9.57-5  END MIXT
READ GEOMETRY
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID  0 1 6.87 -6.87 6.87 -6.87 6.505 -6.505
END GEOMETRY
READ ARRAY  NUX=2 NUY=2 NUZ=2  END ARRAY
END DATA
```

SAMPLE PROBLEM 11 2C8 BARE READ RESTART DATA

This problem is a restart of sample problem 10. The problem is restarted from the 10th set of restart data that was written by sample problem 10 (i.e., it restarts with the fifty-first generation).

CARD INPUT

```
SAMPLE PROBLEM 11  2C8 BARE  READ RESTART DATA
READ PARAM  BEG=51  RST=95 RES=0 END PARAM
END DATA
```

SAMPLE PROBLEM 12 4 AQUEOUS 4 METAL

This problem is a critical experiment consisting of a composite array[1,3] of four highly enriched uranium metal cylinders and four cylindrical plexiglas containers filled with uranyl nitrate solution. The metal units in this experiment are designated in Table II of Ref. 1 as cylinder index 11 and reflector index 1. A photograph of the experiment is given in Fig. F11.D.3.

CARD INPUT

```
SAMPLE PROBLEM 12 4 AQUEOUS 4 METAL MIXED UNITS
READ PARAM LIB=41 FLX=YES FDN=YES TME=5.0 NUB=YES SMU=YES MKP=YES
MKU=YES FMP=YES FMU=YES END PARAM
READ MIXT SCT=1 MIX=1 92860 3.2275-3 92501 4.4802-2 MIX=2 1102 5.81-2
7100 1.9753-3 8100 3.6927-2 92501 9.8471-4 92860 7.7697-5
MIX=3 6100 3.5552-2 1102 5.6884-2 8100 1.4221-2 END MIXT
READ GEOM
BOX TYPE 1
CYLINDER 2 1 9.525 8.89 -8.89
CYLINDER 3 1 10.16 9.525 -9.525
CUBOID 0 1 10.875 -10.875 10.875 -10.875 10.24 -10.24
BOX TYPE 2
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID 0 1 6.59 -15.16 6.59 -15.16 6.225 -14.255
BOX TYPE 3
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID 0 1 6.59 -15.16 15.16 -6.59 6.225 -14.255
BOX TYPE 4
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID 0 1 6.59 -15.16 6.59 -15.16 14.255 -6.225
BOX TYPE 5
CYLINDER 1 1 5.748 5.3825 -5.3825
CUBOID 0 1 6.59 -15.16 15.16 -6.59 14.255 -6.225
END GEOM READ ARRAY NUX=2 NUY=2 NUZ=2 LOOP
1 3R2 1 2 1 1 2 1 2 9R1 3 3R1 2 2 1 3R1 4 6R1 2 2 1 5 3R1 2 2 1 2 2 1
END ARRAY
END DATA
```

SAMPLE PROBLEM 13 TWO CUBOIDS IN A CYLINDRICAL ANNULUS

This critical experiment[4] consists of two assemblies of 93.2% ^{235}U-enriched uranium metal (ρ =18.69g/cc) stacked vertically. The bottom assembly contains a uranium metal cuboid offset to the left within a uranium metal cylindrical annulus. The top assembly contains a uranium metal cuboid offset to the right within a uranium metal cylindrical annulus. The cuboid extends above the annulus. A drawing of the two sections and the total assembly is given in Fig. F11.D.4.

CARD INPUT

```
SAMPLE PROBLEM 13  TWO CUBOIDS IN A CYLINDRICAL ANNULUS
READ PARAM  TME=3.0 LIB=41 END PARAM
READ GEOM
UNIT 1
CUBOID  1 1 6.35 -6.35 6.35 -6.35 7.62 0.0
CYLINDER  0 1 13.97 7.62 0.0 ORIG -6.0934 0.0
CYLINDER  1 1 19.05 7.62 0.0 ORIG -6.0934 0.0
CUBOID  0 1 12.9566 -25.1434 19.05 -19.05 7.62 0.0
UNIT 2
CUBOID  1 1 6.35 -6.35 6.35 -6.35 8.56 0.0
CYLINDER  0 1 13.97 8.56 0.0 ORIGIN 6.0934 0.0
CYLINDER  1 1 19.05 8.56 0.0 ORIGIN 6.0934 0.0
CUBOID  0 1 25.1434 -12.9566 19.05 -19.05 8.56 0.0
UNIT 3
CUBOID  1 1 6.35 -6.35 6.35 -6.35 2.616 0.0
CUBOID  0 1 25.1434 -12.9566 19.05 -19.05 2.616 0.0
END GEOM
READ MIXT  SCT=1 MIX=1 92500 4.48006-2 92800 2.6578-3 92400 4.827-4
92600 9.57-5 END MIXT
READ ARRAY NUX=1 NUY=1 NUZ=3  FILL 1 2 3 T END ARRAY END DATA
```

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

This critical experiment[4] consists of a 93.2% ^{235}U-enriched uranium metal cylinder within a cylindrical annulus of the same material as shown in Fig. F11.D.5. The uranium metal specification is identical to that used in sample problem 13.

CARD INPUT

```
SAMPLE PROBLEM 14  U METAL CYLINDER IN AN ANNULUS
READ PARAM LIB=41 TME=3.0 END PARAM
READ MIXT SCT=1 MIX=1 92500 4.48006-2 92800 2.6578-3 92400 4.827-4
92600 9.57-5 END MIXT
READ GEOM
CYLINDER  1 1 8.89 10.109 0.0 ORIG 5.08 0.0
CYLINDER  0 1 13.97 10.109 0.0
CYLINDER  1 1 19.05 10.109 0.0
END GEOM
END DATA
```

SAMPLE PROBLEM 15 SMALL WATER REFLECTED SPHERE ON PLEXIGLAS COLLAR

This critical experiment[5] is a small highly enriched uranium sphere supported by a plexiglas doughnut in a tank of water. The sphere extends down through the hole of the doughnut. However, the KENO V geometry package cannot rigorously describe a doughnut. Therefore, the KENO V mock-up of this problem describes the doughnut as an annular cylindrical plate and the sphere is supported by it. Both are contained in a cylindrical tank of water. A drawing of the experiment is given in Fig. F11.D.6. This drawing shows the sphere above the cylindrical collar for the sake of clarity. The sphere is actually supported by the collar and extends into the opening in its center. The actual experiment utilized a torus or doughnut instead of a cylindrical collar.

CARD INPUT

```
SAMPLE PROBLEM 15  SMALL WATER REFLECTED SPHERE ON PLEXIGLAS COLLAR
READ PARAM  LIB=41 TME=8.0 FLX=YES FDN=YES END PARAM
READ MIXT  SCT=1  MIX=1 92500 4.7048-2  92600 9.5932-5
92400 5.6717-4  92800 4.6611-4  MIX=2  602 1.0  MIX=3 502 1.0
END MIXT
READ GEOM
UNIT 1
HEMISPHE-Z 1 1 6.5537 CHORD -5.09066
CYLINDER  3 1 4.1275 -5.09066 -7.63065
CYLINDER  2 1 12.7   -5.09066 -7.63065
CUBOID  3 1 4P12.7 -5.09066 -7.63065
UNIT 2
HEMISPHE+Z 1 1 6.5537 CHORD 5.09066
CUBOID  3 1 4P12.7 6.5537 -5.09066
CORE  0 1 -12.7 -12.7 -7.092175
CYLINDER  3 1 17.97 2P7.0922
REPLICATE 3 2 3*3.0 5
END GEOM
READ BIAS  ID=500 2 6 END BIAS
READ ARRAY NUX=1 NUY=1 NUZ=2  FILL 1 2 END ARRAY
READ PLOT TTL='X-Z SLICE THROGH THE CENTER OF THE SPHERE'
XUL=-20.0 ZUL=10.0 YUL=0.0  XLR=20.0 YLR=0.0 ZLR=-10.0
UAX=1.0 WDN=-1.0 NAX=130    NCH=' *0-'
END PLOT
END DATA
```

SAMPLE PROBLEM 16 UO2F2 INFINITE SLAB K-INFINITY

This problem solves for the k-infinity of an infinite number of slabs of uranyl fluoride solution contained in pyrex glass and separated by borated uranyl flouride solution. The uranyl fluoride slab is 4.958 cm thick, 93.2% enriched, and has a density of 578.7 g U/liter. The pyrex glass is 1.27 cm thick and is present on both faces of the uranyl fluoride solution. A total of 27.46 cm of borated solution separates the pyrex glass of adjacent slabs of solution. 1.482×10^{-27} atoms of boron per milliliter is present in the borated solution.

CARD INPUT

```
SAMPLE PROBLEM 16 UO2F2 INFINITE SLAB K-INFINITY
READ PARAMETERS TME=5.0 LIB=41 AMX=YES XAP=NO  END PARAMETERS
READ MIXT  SCT=1
MIX=1 92508 1.382-3 92851 4.978-5 92852 4.978-5 9100 2.963-3
8100 3.340-2 1102 6.088-2  MIX=2  13100 5.3-4 5100 4.59-3
8100 4.492-2 14100 1.802-2 11100 2.380-3  MIX=3  1 1.0 5100 1.482-3
END MIXT
READ GEOMETRY
CUBOID 1 1 2.479 -2.479 100 -100 100 -100
CUBOID 2 1 3.749 -3.749 100 -100 100 -100
CUBOID 3 1 17.479 -17.479 100 -100 100 -100
END GEOM
READ BOUNDS  ALL=MIRROR  END BOUNDS   READ ARRAY  END ARRAY
END DATA
END
```

SAMPLE PROBLEM 17 93% UO2F2 SOLUTION SPHERE ADJOINT CALCULATION

A single 93% enriched uranyl fluoride sphere is run as an adjoint calculation. The result for the forward and adjoint k-effectives should be the same within statistical error when the problem is run both ways.

CARD INPUT

```
SAMPLE PROBLEM 17 93% UO2F2 SOLUTION SPHERE ADJOINT CALCULATION
READ PARAMETERS TME=5.0 LIB=41 ADJ=YES AMX=YES XAP=NO END PARAMETERS
READ MIXT  SCT=1
MIX=1 92500 3.206-4 92800 2.339-5 1102 6.52?6-2 8100 3.3306-2
9100 6.879-4
END MIXT
READ GEOMETRY
SPHERE 1 1 16.0
END GEOM
END DATA
END
```

SAMPLE PROBLEM 18 1F27 DEMONSTRATION OF OPTIONS

A reflected cubic array of 27 cylinders of aqueous uranyl nitrate in Plexiglas bottles.[6] The walls of the bottles were 0.64-cm thick and each bottle was filled with 5 liters of 92.6% enriched solution at an $H/^{235}U$ atomic ratio of 59 and an $N/^{235}U$ atomic ratio of 2.006. The 3x3x3 array was surrounded by a 6-in. paraffin reflector. Most of the print options available in KENO V.a are exercised in this problem. A perspective of this critical experiment is shown in Fig. F11.D.7. A photograph of one of the experiments utilizing 27 of the Plexiglas bottles is shown in Fig. F11.D.8. Sample problem 18 has 15.24 cm of paraffin on all 6 faces rather than the 2.54-cm Plexiglas shown on 5 faces.

CARD INPUT

```
SAMPLE PROBLEM 18   1F27 DEMONSTRATION OF OPTIONS PROBLEM
READ PARA TME=10.  GEN=53 NPG=350 FDN=YES NUB=YES LIB=41
MKU=YES FMU=YES MKH=YES FMH=YES MKP=YES FMP=YES MKA=YES FMA=YES
PWT=YES FAR=YES FLX=YES AMX=YES PAX=YES PGM=YES LNG=20000
END PARA
READ MIXT SCT=1 MIX=1 92508 9.8435-4 92852 .000077660 7100 .0019746
8100 .037086 1102 .058076  MIX=2 602 1  MIX=3 1102 .082571 6100
READ BOUNDS  -ZB= H2O  END BOUNDS
READ GEOM UNIT 1 CYLINDER 1 1 9.52 8.7804 -8.7804
CYLINDER 0 1 9.52 8.9896 -8.7804
CYLINDER 2 1 10.16 9.6296 -9.4204
CUBOID 4 1 18.45 -18.45 18.45 -18.45 17.8946 -17.6854
UNIT 2 ARRAY 1 3*0.0
UNIT 3 ARRAY 2 3*0.0
UNIT 4 ARRAY 3 3*0.0
UNIT 5 ARRAY 4 3*0.0
GLOBAL
UNIT 6 CUBOID 4 1 55.3501 -55.3501 55.3501 -55.3501 53.3701 -53.3701
HOLE 2 -55.35 -18.45 -17.79
HOLE 3 -55.35 -18.45 -53.37
HOLE 4  18.45 -18.45 -53.37
HOLE 5 -55.35 -55.35 -53.37
REPLICATE 3 2 6*3 5  REPLICATE 3 7 6*0.24 1  END GEOM
READ BIAS  ID=400 2 7  END BIAS
READ ARRAY
ARA=1 NUX=2 NUY=2 NUZ=2 FILL F1 END FILL
ARA=2 NUX=2 NUY=2 NUZ=1 FILL F1 END FILL
ARA=3 NUX=1 NUY=2 NUZ=3 FILL F1 END FILL
ARA=4 NUX=3 NUY=1 NUZ=3 FILL F1 END FILL
END ARRAY
READ START NST=6 TFX=0.0 TFY=0.0 TFZ=0.0
LNU=350 PS6=YES
END START
READ PLOT
TTL=?  1F27 XY PLOT AT Z=0.0 ?  XUL=0.0 YUL=141.18 ZUL=68.61
XLR=141.18 YLR=0.0 ZLR=68.61 UAX=1 VDN=-1 NAX=130 NCH=?.*-3 ?
RUN=YES END      TTL=?UNIT MAP 1F27 XY PLOT AT Z=0.0?
PIC=UNIT NCH=? 123456? END PLOT    END DATA
END
```

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

This critical experiment was described previously as SAMPLE PROBLEM 12. The input data given below utilizes the array of arrays option. See Fig. F11.D.3.

CARD INPUT

```
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)
READ PARAM  LIB=41  FLX=YES FDN=YES  TME=2.0 NUB=YES SMU=YES MKP=YES
MKU=YES FMP=YES FMU=YES END PARAM
READ MIXT SCT=1  MIX=1  92860 3.2275-3  92501 4.4802-2  MIX=2  1102 5.81-2
7100 1.9753-3  8100 3.6927-2  92501 9.8471-4  92860 7.7697-5
MIX=3  6100 3.5552-2  1102 5.6884-2  8100 1.4221-2  END MIXT
READ GEOM
UNIT 1
COM='URANYL NITRATE SOLUTION IN A PLEXIGLAS CONTAINER'
CYLINDER  2 1 9.525 2P8.89
CYLINDER  3 1 10.16 2P9.525
CUBOID  0 1 4P10.875 2P10.24
UNIT 2
COM='URANIUM METAL CYLINDER'
CYLINDER  1 1 5.748 2P5.3825
CUBOID  0 1 4P6.59 2P6.225
UNIT 3
COM='1x2x2 ARRAY OF SOLUTION UNITS'
ARRAY 1 3*0.0
UNIT 4
COM='1x2x2 ARRAY OF METAL UNITS PADDED TO MATCH SOLUTION ARRAY'
ARRAY 2 3*0.0
REPLICATE 0 1 2*0.0 2*8.57 2*8.03 1
END GEOM
READ ARRAY ARA=1 NUX=1 NUY=2 NUZ=2 FILL F1 END FILL
ARA=2 NUX=1 NUY=2 NUZ=2 FILL F2 END FILL GBL=3 ARA=3 NUX=2 NUY=1 NUZ=1
COM='COMPOSITE ARRAY OF SOLUTION AND METAL UNITS'
FILL 4 3 END FILL
END ARRAY
END DATA
```

SAMPLE PROBLEM 20 TRIANGULAR PITCHED ARRAY

This problem is a critical experiment[7] consisting of seven cylinders in a triangular pitched unreflected array. The central cylinder has six cylinders arranged around it. The surface-to-surface separation between the units is 0.15 cm. Each unit consists of a 60-mil-thick aluminum can with an 8-in. inside diameter, filled with a solution of 93.2% enriched uranyl nitrate with a $H/^{235}U$ atomic ratio of 44.3 and a density of 576.87 gU/l. The apparatus for conducting this experiment is shown in Fig. F11.D.9.

CARD INPUT

```
SAMPLE PROBLEM 20 TRIANGULAR PITCHED ARRAY
READ PARAM TME=5 LIB=41 END PARAM
READ MIXT SCT=1 MIX=1 92508 1.3773-3 92852 9.9222-5
8100 3.3246-2 9100 2.9531-3 1102 6.0586-2 MIX=2 13100 6.0242-2 END MIXT
READ GEOM
UNIT 1
CYLINDER   1 1 10.16 18.288 0
CYLINDER   2 1 10.312 18.288 -.152
UNIT 2
CUBOID     0 1 4P50 50 -.152
HOLE       1 3R0
HOLE       1 21.006 2R0
HOLE       1 -21.006 2R0
HOLE       1 10.503 18.192 0
HOLE       1 -10.503 18.192 0
HOLE       1 10.503 -18.192 0
HOLE       1 -10.503 -18.192 0
END GEOM
READ ARRAY NUX=1 NUY=1 NUZ=1 FILL 2 END FILL END ARRAY
READ MIXT SCT=1 MIX=1 92508 1.3773-3 92852 9.9222-5
8100 3.3246-2 9100 2.9531-3 1102 6.0586-2 MIX=2 13100 6.0242-2 END MIXT
READ PLOT TTL='HEX ARRAY' PIC=MIX
XUL=0 YUL=100 ZUL=10
XLR=100 YLR=0 ZLR=10 UAX=1 VDN=-1 NAX=120
NCH=' 12' END PLOT
END DATA
```

SAMPLE PROBLEM 21 PARTIALLY FILLED SPHERE

This critical experiment[8] consisted of a partially filled, unreflected spherical container. This aluminum container had an inside diameter of 27.244 in. and a wall thickness of 1/16 in. It is referred to in the report as the 27.3-in.-diameter vessel. The sphere was 98% filled with uranyl fluoride at an enrichment of 4.89% with an $H/^{235}U$ atomic ratio of 1099. The height of the solution in the sphere was 64.6 cm above the bottom of the sphere. A schematic diagram of the apparatus used in the experiment is given in Fig. F11.D.10. The steel tank was ignored.

CARD INPUT

```
SAMPLE PROBLEM 21  PARTIALLY FILLED SPHERE
READ PARAM TME=5 LIB=41 END PARAM
READ GEOM
HEMISPHE-Z  1 1 34.6   CHORD 30.
SPHERE      0 1 34.6
SPHERE      2 1 34.759
END GEOM
READ MIXT SCT=1 MIX=1 92400 2.54-7 92500 6.1741-5 92843 7.6067-4
92844 4.2788-4 8100 3.3260-2 9100 2.4949-3 1102 6.1531-2
MIX=2 13100 6.0242-2  END MIXT
END DATA
```

F11.D.1 REFERENCES

1. J. T. Thomas, "Critical Three-Dimensional Arrays of U(93.2)-Metal Cylinders," *Nucl. Sci. Eng. 52,* 350 (November 1973).

2. G. E. Whitesides and J. T. Thomas, "The Use of Differential Current Albedos in Monte Carlo Criticality Calculations," *Trans. Am. Nucl. Soc. 12,* 889 (November 1969).

3. J. T. Thomas, "Critical Three-Dimensional Arrays of Neutron-Interacting Units, Part II," ORNL/TM-868 (July 1964).

4. D. C. Irving and J. T. Mihalczo, "Monte Carlo Calculations for Enriched Uranium Metal Assemblies," *Trans. Am. Nucl. Soc. 7,* 284 (November 1964).

5. Cleo C. Byers et al., "Critical Measurements of a Water-Reflected Enriched Uranium Sphere," *Trans. Am. Nucl. Soc. 27,* 412 (November 1977).

6. J. T. Thomas, "Critical Three-Dimensional Arrays of Neutron-Interacting Units," ORNL/TM-719 (1963).

7. J. K. Fox, L. W. Gilley, D. Callihan, "Critical Mass Studies, Part IX Aqueous U-235 Solutions," ORNL-2367 (February 1958).

8. D. F. Cronin, "Critical Mass Studies, Part X Uranium of Intermediate Enrichment," ORNL-2968 (September 1960).

Table F11.D.1. KENO 16-group Hansen-Roach cross-section library

Mixture Title	ID
CARBON STEEL 7.82g/cc C=.003921, FE=.083491	100
TYPE 304 STAINLESS STEEL 7.9g/cc CONTAINS C,CR,MN,FE,NI,SI	200
OAK RIDGE CONCRETE 2.3g/cc	300
ORDINARY CONCRETE 2.37g/cc	301
MAGNUSON CONCRETE 2.15g/cc	302
X(E) POLYETHYLENE 0.92g/cc H=.079433, C=.039716	401
DE/E POLYETHYLENE 0.92g/cc H=.079433, C=.039716	402
X(E) WATER 0.9982g/cc H=.066742, O=.033371	501
DE/E WATER 0.9982g/cc H=.066742, O=.033371	502
X(E) PLEXIGLAS 1.182g/cc H=.056884, C=.035552, O=.014221	601
DE/E PLEXIGLAS 1.182g/cc H=.056884, C=.035552, O=.014221	602
MAGNUSON VERMICULITE	701

Element Title	Source	ID
HYDROGEN X(E)	Hansen Roach	1101
HYDROGEN DE/E	Hansen Roach	1102
DEUTERIUM X(E)	Hansen Roach	1201
LITHIUM-6	Hansen Roach	3100
LITHIUM-7	Hansen Roach	3200
BERYLLIUM	Hansen Roach	4100
BORON (natural)	Hansen Roach	5100
CARBON	Hansen Roach	6100
NITROGEN	Hansen Roach	7100
OXYGEN	Hansen Roach	8100
FLUORINE	Hansen Roach	9100
SODIUM	Hansen Roach	11100
MAGNESIUM	XSDRN	12100
ALUMINUM	Hansen Roach	13100
SILICON	XSDRN	14100
SULFUR	XSDRN	16100
CHLORINE	Hansen Roach	17100
POTASSIUM	Hansen Roach	19100
CALCIUM	GAM-2	20100
TITANIUM	GAM-2	22100
VANADIUM	GAM-2	23100
CHROMIUM	AEROJET	24100
MANGANESE	XSDRN	25100
IRON	Hansen Roach	26100
COBALT	Hansen Roach	27100
NICKEL	Hansen Roach	28100
COPPER	XSDRN	29100
ZINC	GAM-2	30100
ZIRCONIUM	Hansen Roach	40100
NIOBIUM	Hansen Roach	41100
MOLYBDENUM	Hansen Roach	42100
CADMIUM	GAM-2	48100
INDIUM	GAM-2	49100
CERIUM	Hansen Roach	58100
SAMARIUM	GAM-2	62100

Table F11.D.1. (cont.)

Element Title	Source	ID
EUROPIUM	GAM-2	63100
GADOLINIUM	GAM-2	64100
TANTALUM	Hansen Roach	73100
TUNGSTEN	GAM-2	74100
LEAD	XSDRN	82100
TH-232 INFINITE DILUTION	Hansen Roach	90200
TH-232 SIG P = 20	Hansen Roach	90202
TH-232 SIG P = 40	Hansen Roach	90204
TH-232 SIG P = 50	Hansen Roach	90104
TH-232 SIG P = 60	Hansen Roach	90206
TH-232 SIG P = 80	Hansen Roach	90208
TH-232 SIG P = 100	Hansen Roach	90210
TH-232 SIG P = 150	Hansen Roach	90212
TH-232 SIG P = 200	Hansen Roach	90214
TH-232 SIG P = 300	Hansen Roach	90216
TH-232 SIG P = 400	Hansen Roach	90218
TH-232 SIG P = 600	Hansen Roach	90220
TH-232 SIG P = 800	Hansen Roach	90222
TH-232 SIG P = 1000	Hansen Roach	90224
TH-232 SIG P = 1250	Hansen Roach	90109
TH-232 SIG P = 1500	Hansen Roach	90226
TH-232 SIG P = 1750	Hansen Roach	90111
TH-232 SIG P = 2000	Hansen Roach	90228
TH-232 SIG P = 2500	Hansen Roach	90113
TH-232 SIG P = 3000	Hansen Roach	90230
TH-232 SIG P = 3500	Hansen Roach	90115
TH-232 SIG P = 4000	Hansen Roach	90232
TH-232 SIG P = 6000	Hansen Roach	90234
TH-232 SIG P = 10K	Hansen Roach	90236
TH-232 SIG P = 20K	Hansen Roach	90238
U-233	Hansen Roach	92300
U-233-1 SIG P = 20	Hansen Roach	92301
U-233-2 SIG P = 40	Hansen Roach	92302
U-233-3 SIG P = 60	Hansen Roach	92303
U-233-4 SIG P = 100	Hansen Roach	92304
U-233-5 SIG P = 200	Hansen Roach	92305
U-233-6 SIG P = 400	Hansen Roach	92306
U-233-7 SIG P = 600	Hansen Roach	92307
U-233-8 SIG P = 1000	Hansen Roach	92308
U-233-9 SIG P = 2000	Hansen Roach	92309
U-233-10 SIG P = 4000	Hansen Roach	92310
U-233-11 SIG P = 6000	Hansen Roach	92311
U-233-12 SIG P = 10000	Hansen Roach	92312
U-234	Mihalczo Mod of H-R U-238	92400
U-235 YR	Hansen Roach	92500
U-235-1R SIG P = 20	Hansen Roach	92501
U-235-2R SIG P = 40	Hansen Roach	92502
U-235-3R SIG P = 60	Hansen Roach	92503
U-235-4R SIG P = 100	Hansen Roach	92504
U-235-5R SIG P = 200	Hansen Roach	92505
U-235-6R SIG P = 400	Hansen Roach	92506

Table F11.D.1. (cont.)

Element Title	Source	ID
U-235-7R SIG P = 600	Hansen Roach	92507
U-235-8R SIG P = 1000	Hansen Roach	92508
U-235-9R SIG P = 2000	Hansen Roach	92509
U-235-10R SIG P = 4000	Hansen Roach	92510
U-235-11R SIG P = 6000	Hansen Roach	92511
U-235-12R SIG P = 10000	Hansen Roach	92512
U-236	Mihalczo Mod of H-R U-238	92600
U-238 Y	Hansen Roach	92800
U-238 SIG P = 12	Hansen Roach JRK Mod	92801
U-238 SIG P = 15	Hansen Roach JRK Mod	92802
U-238 SIG P = 20	Hansen Roach JRK Mod	92803
U-238 SIG P = 25	Hansen Roach JRK Mod	92804
U-238 SIG P = 30	Hansen Roach JRK Mod	92805
U-238 SIG P = 35	Hansen Roach JRK Mod	92806
U-238 SIG P = 40	Hansen Roach JRK Mod	92807
U-238 SIG P = 45	Hansen Roach JRK Mod	92808
U-238 SIG P = 50	Hansen Roach JRK Mod	92809
U-238 SIG P = 55	Hansen Roach JRK Mod	92810
U-238 SIG P = 60	Hansen Roach JRK Mod	92811
U-238 SIG P = 65	Hansen Roach JRK Mod	92812
U-238 SIG P = 70	Hansen Roach JRK Mod	92813
U-238 SIG P = 75	Hansen Roach JRK Mod	92814
U-238 SIG P = 80	Hansen Roach JRK Mod	92815
U-238 SIG P = 85	Hansen Roach JRK Mod	92816
U-238 SIG P = 90	Hansen Roach JRK Mod	92817
U-238 SIG P = 95	Hansen Roach JRK Mod	92818
U-238 SIG P = 100	Hansen Roach JRK Mod	92819
U-238 SIG P = 110	Hansen Roach JRK Mod	92820
U-238 SIG P = 120	Hansen Roach JRK Mod	92821
U-238 SIG P = 130	Hansen Roach JRK Mod	92822
U-238 SIG P = 140	Hansen Roach JRK Mod	92823
U-238 SIG P = 160	Hansen Roach JRK Mod	92824
U-238 SIG P = 180	Hansen Roach JRK Mod	92825
U-238 SIG P = 200	Hansen Roach JRK Mod	92826
U-238 SIG P = 220	Hansen Roach JRK Mod	92827
U-238 SIG P = 240	Hansen Roach JRK Mod	92828
U-238 SIG P = 260	Hansen Roach JRK Mod	92829
U-238 SIG P = 280	Hansen Roach JRK Mod	92830
U-238 SIG P = 300	Hansen Roach JRK Mod	92831
U-238 SIG P = 330	Hansen Roach JRK Mod	92832
U-238 SIG P = 360	Hansen Roach JRK Mod	92833
U-238 SIG P = 400	Hansen Roach	92834
U-238 SIG P = 450	Hansen Roach	92835
U-238 SIG P = 500	Hansen Roach	92836
U-238 SIG P = 550	Hansen Roach	92837
U-238 SIG P = 600	Hansen Roach	92838
U-238 SIG P = 650	Hansen Roach	92839
U-238 SIG P = 700	Hansen Roach	92840
U-238 SIG P = 800	Hansen Roach	92841
U-238 SIG P = 900	Hansen Roach	92842

Table F11.D.1. (cont.)

Element Title	Source	ID
U-238 SIG P = 1000	Hansen Roach	92843
U-238 SIG P = 1500	Hansen Roach	92844
U-238 SIG P = 2000	Hansen Roach	92845
U-238 SIG P = 3000	Hansen Roach	92846
U-238 SIG P = 4000	Hansen Roach	92847
U-238 SIG P = 5000	Hansen Roach	92848
U-238 SIG P = 6000	Hansen Roach	92849
U-238 SIG P = 8000	Hansen Roach	92850
U-238 SIG P = 10000	Hansen Roach	92851
U-238 SIG P = 20000	Hansen Roach	92852
U-238 SIG P = 40000	Hansen Roach	92853
U-238 SIG P = 60000	Hansen Roach	92854
U-238 SIG P = 100000	Hansen Roach	92855
U-238-1R SIG P = 20	Hansen Roach	92856
U-238-2R SIG P = 40	Hansen Roach	92857
U-238-3R SIG P = 60	Hansen Roach	92858
U-238-4R SIG P = 100	Hansen Roach	92859
U-238-5R SIG P = 200	Hansen Roach	92860
U-238-6R SIG P = 400	Hansen Roach	92861
U-238-7R SIG P = 600	Hansen Roach	92862
PU-240	Hansen Roach	94000
PU-240-1 SIG P = 50	Hansen Roach	94001
PU-240-2 SIG P = 100	Hansen Roach	94002
PU-240-3 SIG P = 200	Hansen Roach	94003
PU-240-4 SIG P = 400	Hansen Roach	94004
PU-240-5 SIG P = 600	Hansen Roach	94005
PU-240-6 SIG P = 1000	Hansen Roach	94006
PU-240-7 SIG P = 2000	Hansen Roach	94007
PU-240-8 SIG P = 4000	Hansen Roach	94008
PU-240-9 SIG P = 6000	Hansen Roach	94009
PU-240-10 SIG P = 10000	Hansen Roach	94010
PU-240-11 SIG P = 20000	Hansen Roach	94011
PU-240-12 SIG P = 40000	Hansen Roach	94012
PU-240-13 SIG P = 60000	Hansen Roach	94013
PU-240-14 SIG P = 100000	Hansen Roach	94014
PU-240-15 SIG P = 200000	Hansen Roach	94015
PU-240-16 SIG P = 400000	Hansen Roach	94016
PU-240-17 SIG P = 600000	Hansen Roach	94017
PU-240-18 SIG P = 1000000	Hansen Roach	94018
PU-241	GAM-2	94100
PU-242	GAM-2	94200
PU-238	Hansen Roach	94800
PU-238-1 SIG P = 100	Persimmon	94801
PU-238-2 SIG P = 1000	Persimmon	94802
PU-238-3 SIG P = 10000	Persimmon	94803
PU-238 Y	Persimmon	94804
PU-239	Hansen Roach	94900
PU-239-1 SIG P = 20	Hansen Roach	94901
PU-239-2 SIG P = 40	Hansen Roach	94902
PU-239-3 SIG P = 60	Hansen Roach	94903

Table F11.D.1. (cont.)

Element Title	Source	ID
PU-239-4 SIG P = 100	Hansen Roach	94904
PU-239-5 SIG P = 200	Hansen Roach	94905
PU-239-6 SIG P = 400	Hansen Roach	94906
PU-239-7 SIG P = 600	Hansen Roach	94907
PU-239-8 SIG P = 1000	Hansen Roach	94908
PU-239-9 SIG P = 2000	Hansen Roach	94909
PU-239-10 SIG P = 4000	Hansen Roach	94910
PU-239-11 SIG P = 6000	Hansen Roach	94911
PU-239-12 SIG P = 10000	Hansen Roach	94912
PU-239-13 SIG P = 20000	Hansen Roach	94913
PU-239-14 SIG P = 40000	Hansen Roach	94914
PU-239-15 SIG P = 60000	Hansen Roach	94915
PU-239-16 SIG P = 100000	Hansen Roach	94916

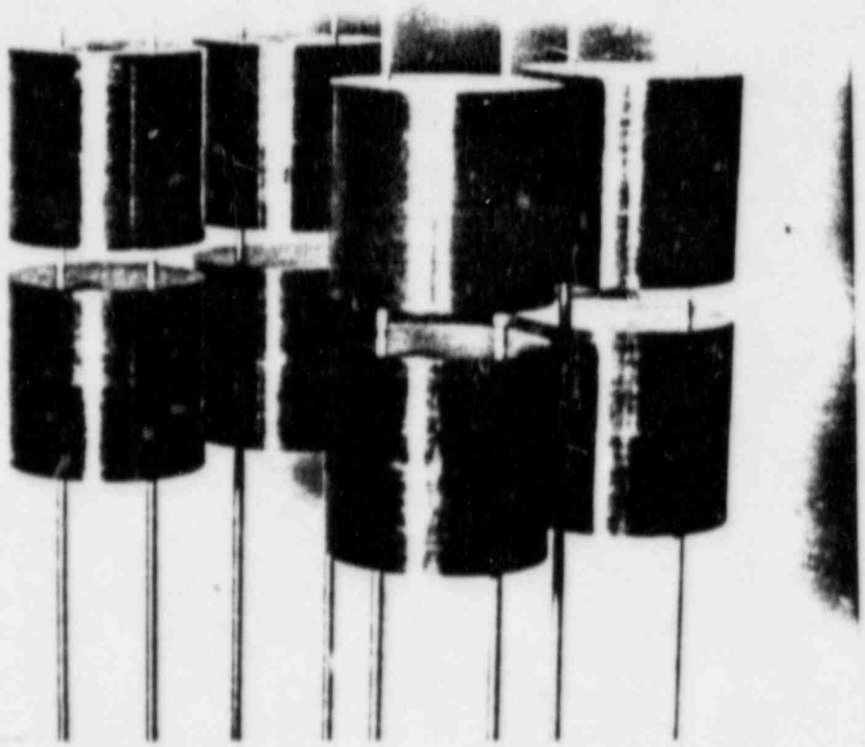


Fig. F11.D.1. Critical 2C8 bare assembly

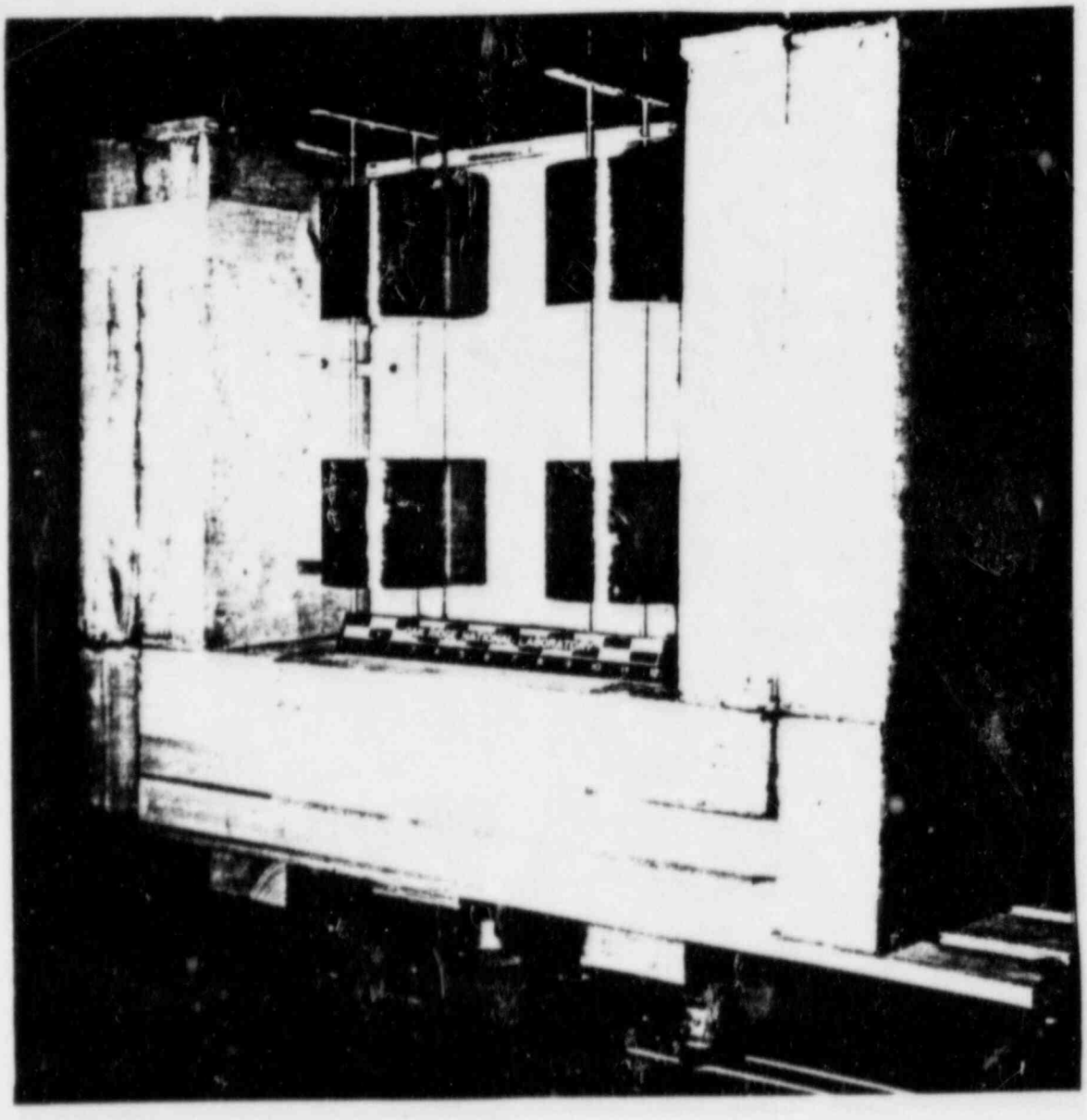


Fig. F11.D.2. Half of the paraffin reflected 2C8 assembly before the top reflector was added

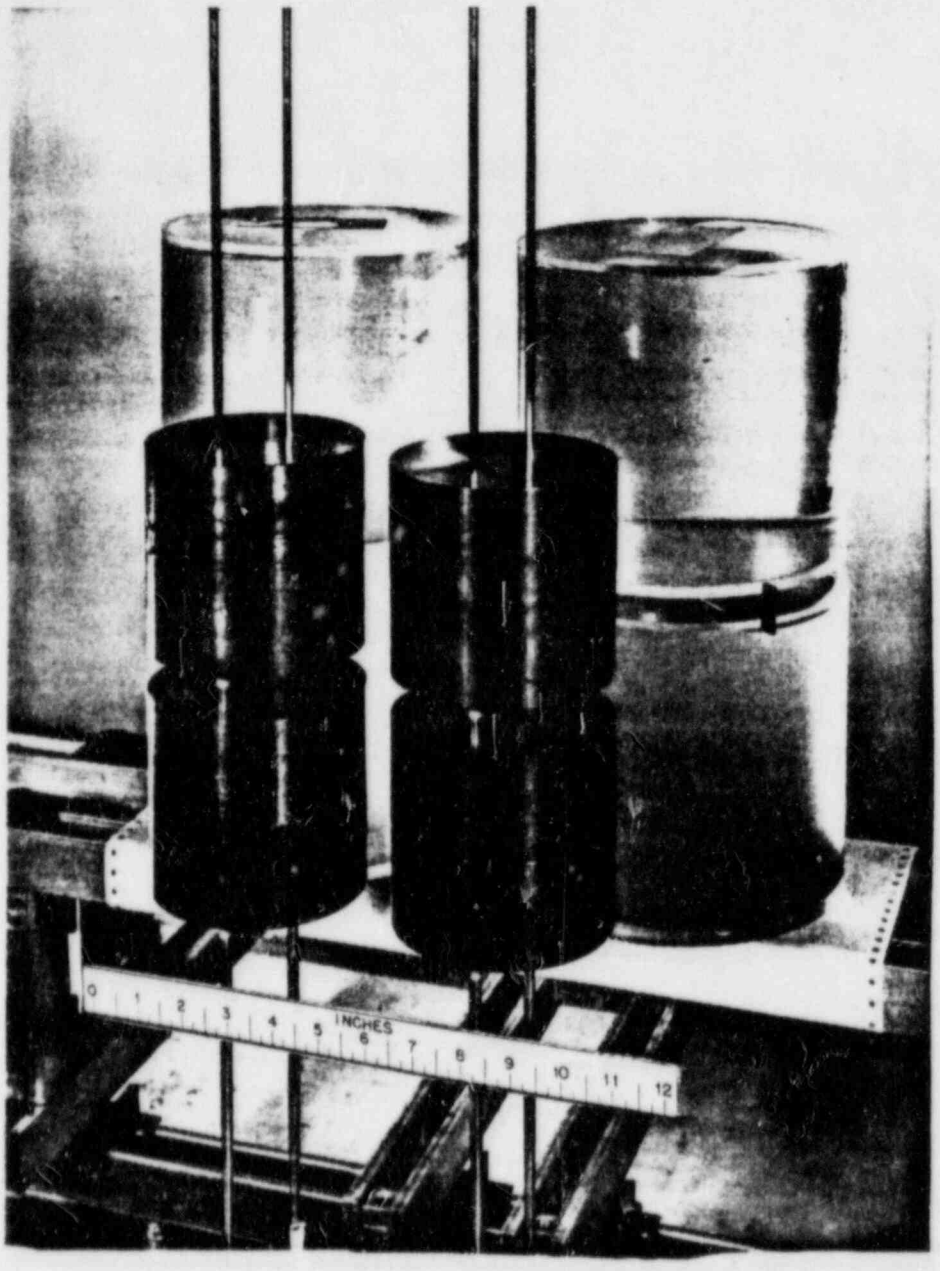


Fig. F11.D.3. Critical assembly of 4 solution units and 4 metal units

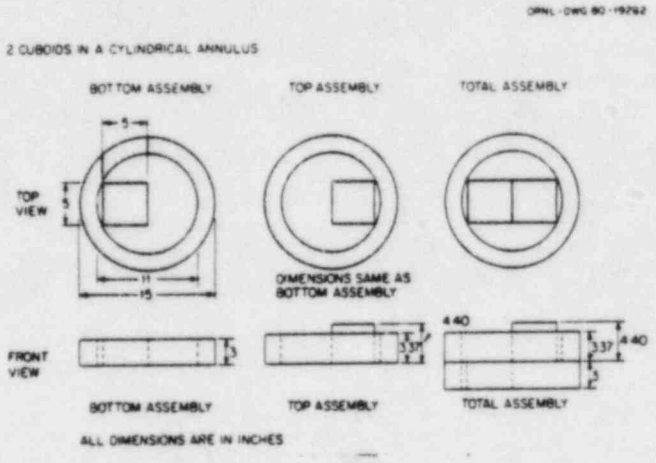


Fig. F11.D.4. Drawing of two cuboids in an annulus critical assembly

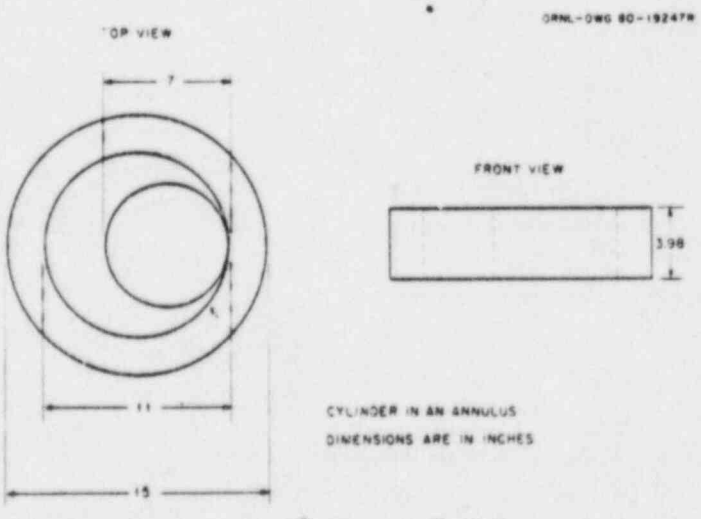


Fig. F11.D.5. Drawing of the cylinder in an annulus critical assembly

ORNL-DWG 80-10537

REFLECTED SPHERE OF U METAL

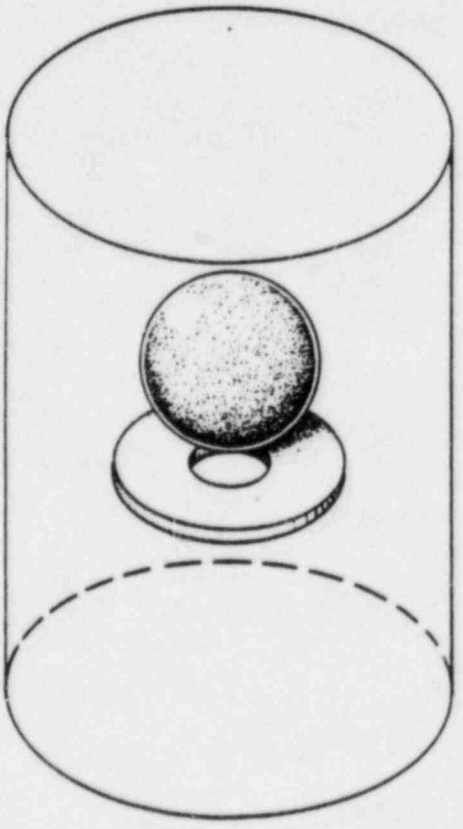


Fig. F11.D.6. Drawing of a critical assembly consisting of a uranium sphere on a plexiglas collar with a cylindrical water reflector

ORNL-DWG 83-12191

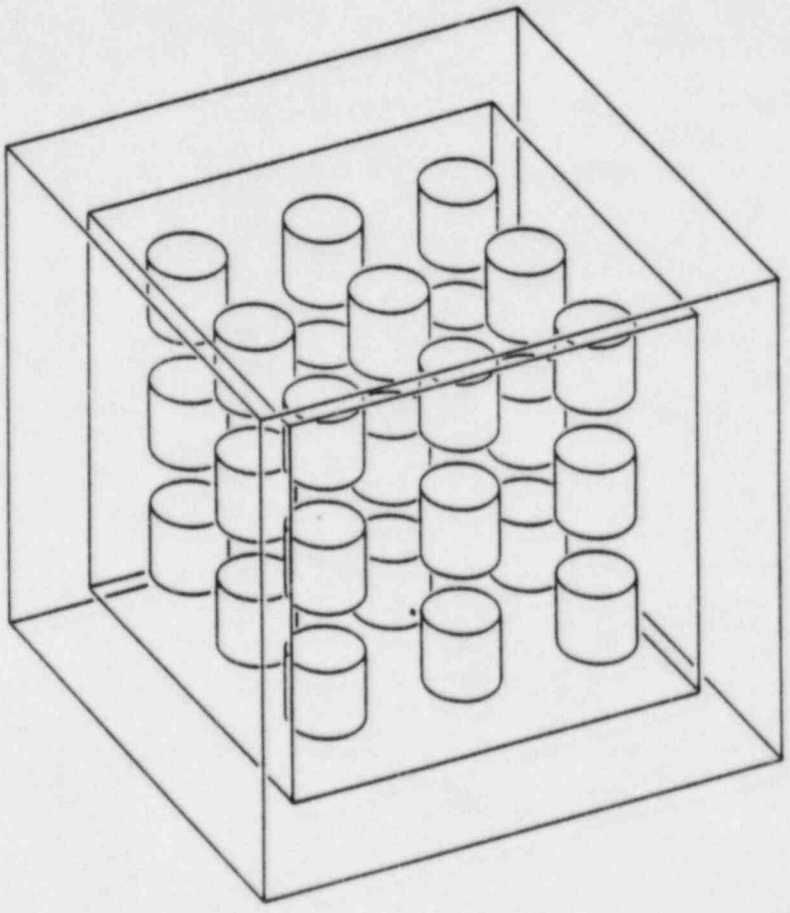


Fig. F11.D.7. Perspective of critical 1F27 experiment

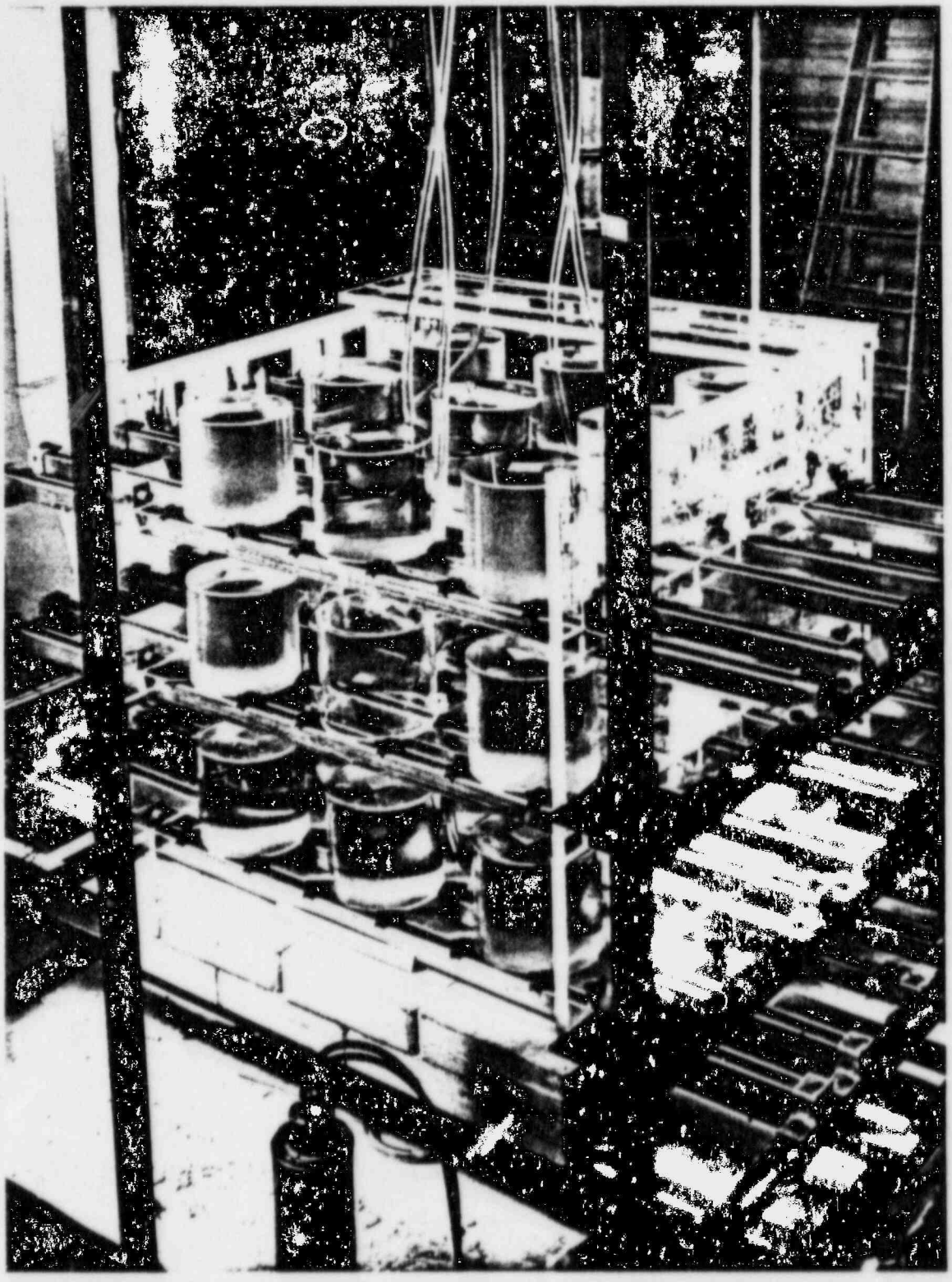


Fig. F11.D.8. View of a 27-unit cubic array with 2.54-cm-thick plexiglas reflector on five sides and a 15.24-cm-thick paraffin base

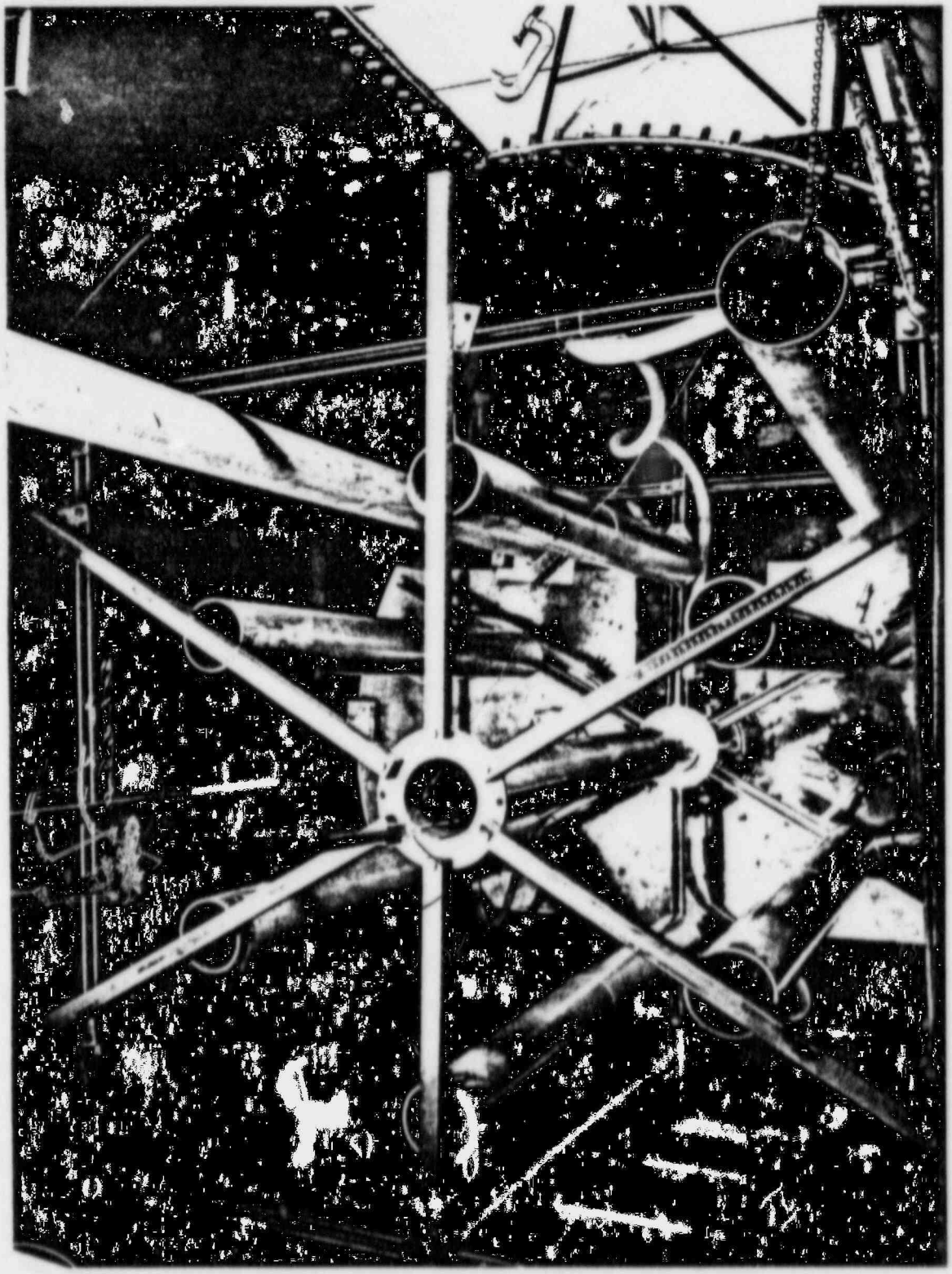


Fig. F11.D.9. Typical arrangement for critical experiments with interacting arrays of aluminum cylinders containing enriched ^{235}U solutions

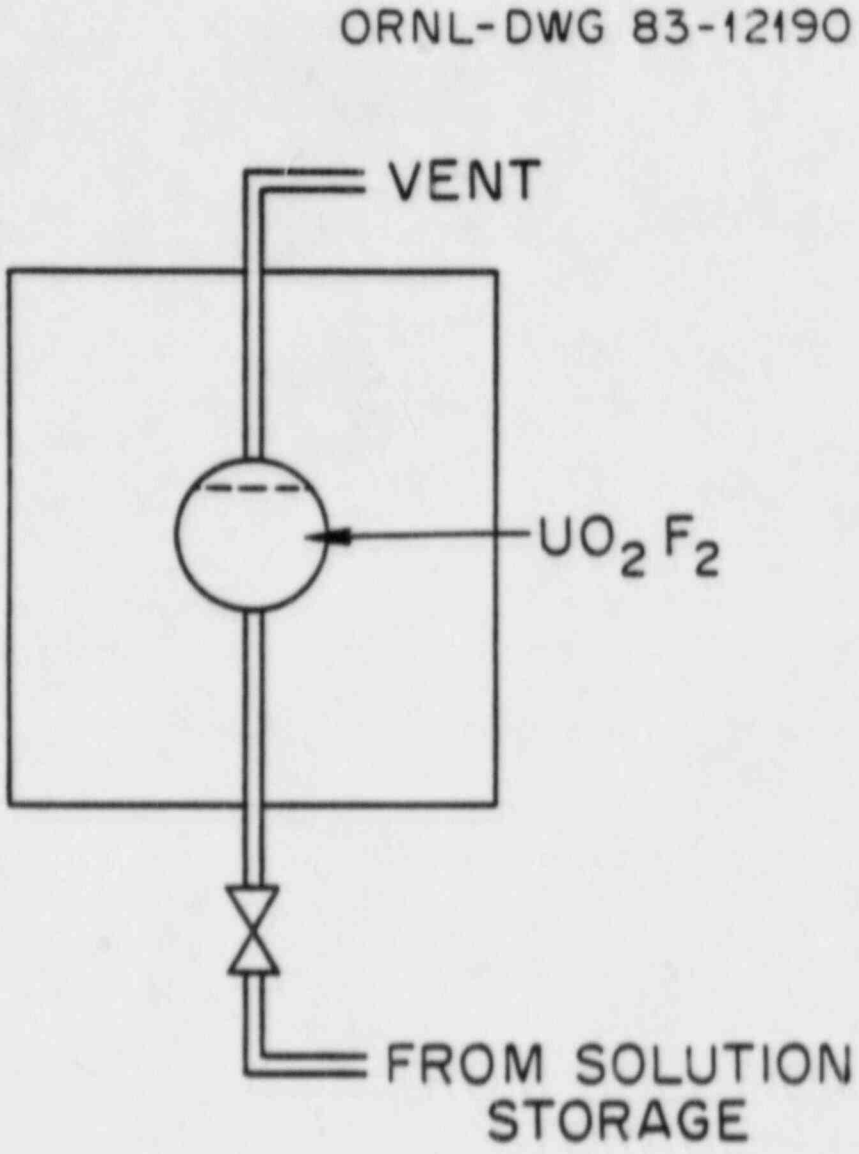


Fig. F11.D.10. Schematic of bare partially filled sphere experiment inside a 9.5-ft-diameter, 9-ft-high steel tank

APPENDIX E

F11.E KENO-V SAMPLE PROBLEM OUTPUT

F11.E.1 SELECTED OUTPUT FROM SAMPLE PROBLEMS

The computer output on the following pages is for sample problems 4, 14, and 19 described in Sect. F11.D

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

****** NUMERIC PARAMETERS ******

TME	MAXIMUM PROBLEM TIME (MIN)	5.00
TBA	TIME PER GENERATION (MIN)	0.50
GEN	NUMBER OF GENERATIONS	103
NPG	NUMBER PER GENERATION	300
NSK	NUMBER OF GENERATIONS TO BE SKIPPED	3
BEG	BEGINNING GENERATION NUMBER	1
NES	GENERATIONS BETWEEN CHECKPOINTS	0
XID	NUMBER OF EXTRA 1-D CROSS SECTIONS	0
NBK	NEUTRON BANK SIZE	325
XNB	EXTRA POSITIONS IN NEUTRON BANK	0
NFB	FISSION BANK SIZE	300
XFB	EXTRA POSITIONS IN FISSION BANK	0
WIA	DEFAULT VALUE OF WEIGHT AVERAGE	0.5000
WTH	WEIGHT HIGH FOR SPLITTING	3.0000
WTL	WEIGHT LOW FOR RUSSIAN ROULETTE	0.3333
RND	STARTING RANDOM NUMBER	00071AFD498D
NBS	NUMBER OF D.A. BLOCKS ON UNIT 8	200
NLS	LENGTH OF D.A. BLOCKS ON UNIT 8	512
ADJ	MODE OF CALCULATION	FORWARD
	INPUT DATA WRITTEN ON RESTART UNIT	NO
	BINARY DATA INTERFACE	NO

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

****** LOGICAL PARAMETERS ******

RUN	EXECUTE PROBLEM AFTER CHECKING DATA	YES	PLT	PLOT PICTURE MAP(S)	YES
FLX	COMPUTE FLUX	YES	FDN	COMPUTE FISSION DENSITIES	YES
ANU	COMPUTE ANG UNIT SELF-MULTIPLICATION	NO	MUB	COMPUTE MU-BAR & AVG FISSION GROUP	NO
MXU	COMPUTE MATRIX K-EFF BY UNIT NUMBER	NO	MXP	COMPUTE MATRIX K-EFF BY UNIT LOCATION	NO
CXU	COMPUTE COFACTOR K-EFF BY UNIT NUMBER	NO	CXP	COMPUTE COFACTOR K-EFF BY UNIT LOCATION	NO
FMU	PRINT FISS PROD MATRIX BY UNIT NUMBER	NO	FMP	PRINT FISS PROD MATRIX BY UNIT LOCATION	NO
MXH	COMPUTE MATRIX K-EFF BY HOLE NUMBER	NO	MXA	COMPUTE MATRIX K-EFF BY ARRAY NUMBER	NO
CXH	COMPUTE COFACTOR K-EFF BY HOLE NUMBER	NO	CXA	COMPUTE COFACTOR K-EFF BY ARRAY NUMBER	NO
FMH	PRINT FISS PROD MATRIX BY HOLE NUMBER	NO	FMA	PRINT FISS PROD MATRIX BY ARRAY NUMBER	NO
HHL	COLLECT MATRIX BY HIGHEST HOLE LEVEL	NO	AAL	COLLECT MATRIX BY HIGHEST ARRAY LEVEL	NO
AMX	PRINT ALL MIXED CROSS SECTIONS	NO	FAR	PRINT FIS. AND ABS. BY REGION	NO
XS1	PRINT 1-D MIXTURE X-SECTIONS	NO	PAX	PRINT XSEC-ALBEDO CORRELATION TABLES	NO
XS2	PRINT 2-D MIXTURE X-SECTIONS	NO	PWT	PRINT WEIGHT AVERAGE ARRAY	YES
XAP	PRINT MIXTURE ANGLES & PROBABILITIES	NO	PGM	PRINT INPUT GEOMETRY	NO
PXI	PRINT FISSION SPECTRUM	NO	DUG	PRINT DEBUG INFORMATION	NO
PXD	PRINT EXTRA 1-D CROSS SECTIONS	NO	TRK	PRINT TRACKING INFORMATION	NO

PARAMETER INPUT COMPLETED

-------- 208 IO'S WERE USED READING THE PARAMETER DATA --------

********** DATA READING COMPLETED **********

SAMPLE PROBLEM 4 2CS 15.24 CM PARAFFIN REFL AUTOMATIC REFL

	UNIT NUMBER	DATA SET NAME	VOLUME NAME	UNIT FUNCTION
XSC	14	SYS83314.T133920.RA000.WPCF11D.R0000013	X3A1C0	MIXED CROSS SECTIONS
ALB	79	E.LWP00000.KENO5.ALBEDOS	NE3330	INPUT ALBEDOS
WTS	80	E.LWP00000.KENO4.WGTS	NE3330	INPUT WEIGHTS
SKT	16	SYS83314.T133920.RA000.WPCF11D.R0000015	X3A1C1	WRITE SCRATCH DATA
LIB	41	E.LWP00000.AMPX.NBXSCS	NE3330	INPUT AMPX WORKING LIBRARY
	8	SYS83314.T133920.RA000.WPCF11D.R0000007	NVSDA2	INPUT DATA DIRECT ACCESS
	9	SYS83314.T133920.RA000.WPCF11D.R0000008	X3A1C1	SUPER GROUPED DIRECT ACCESS
	10	SYS83314.T133920.RA000.WPCF11D.R0000009	X3A1C0	XSEC MIXING DIRECT ACCESS

-------- 45 IO'S WERE USED PREPARING INPUT DATA --------

F11.E.4

SAMPLE PROBLEM 4 2CS 15.24 CM PARAFFIN REFL AUTOMATIC REFL

MIXING TABLE

NUMBER OF SCATTERING ANGLES = 1
CROSS SECTION MESSAGE THRESHOLD =3.0E-05

ENTRY	MIXTURE	NUCLIDE	DENSITY
1	1	92500	4.48006E-02
2	1	92800	2.65780E-03
3	1	92400	4.82700E-04
4	1	92600	9.57000E-05
5	2	1101	8.25810E-02
6	2	6100	3.97020E-02

CROSS SECTIONS READ FROM THE AMPX WORKING LIBRARY ON UNIT 41

1101	HYDROGEN H(H)	HANSEN ROACH
6100	CARBON	HANSEN ROACH
92460	U-234	MINALCEO MOD OF H-R U-238
92500	U-235 YR	HANSEN ROACH
92600	U-236	MINALCEO MOD OF H-R U-238
92800	U-238 Y	HANSEN ROACH

-------- 169 IO'S WERE USED MIXING CROSS-SECTIONS --------

-------- 21 IO'S WERE USED PREPARING THE CROSS SECTIONS --------

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

****** ADDITIONAL INFORMATION ******

NUMBER OF ENERGY GROUPS	16	USE LATTICE GEOMETRY	YES
NO. OF FISSION SPECTRUM SOURCE GROUP	1	GLOBAL ARRAY NUMBER	1
NO. OF SCATTERING ANGLES IN XSECS	1	NUMBER OF UNITS IN THE GLOBAL X DIR.	2
ENTRIES/NEUTRON IN THE NEUTRON BANK	16	NUMBER OF UNITS IN THE GLOBAL Y DIR.	2
ENTRIES/NEUTRON IN THE FISSION BANK	9	NUMBER OF UNITS IN THE GLOBAL Z DIR.	2
NUMBER OF MIXTURES USED	2	USE A GLOBAL REFLECTOR	YES
NUMBER OF BIAS ID'S USED	7	USE NESTED HOLES	NO
NUMBER OF DIFFER \L ALBEDOS USED	0	NUMBER OF HOLES	0
TOTAL INPUT GE .RY REGIONS	9	MAXIMUM HOLE NESTING LEVEL	0
NUMBER OF GEOMETRY REGIONS USED	9	USE NESTED ARRAYS	NO
NUMBER OF GEOMETRY UNITS	2	NUMBER OF ARRAYS USED	1
LARGEST ARRAY NUMBER	1	MAXIMUM ARRAY NESTING LEVEL	1
+X BOUNDARY CONDITION	VACUUM	-X BOUNDARY CONDITION	VACUUM
+Y BOUNDARY CONDITION	VACUUM	-Y BOUNDARY CONDITION	VACUUM
+Z BOUNDARY CONDITION	VACUUM	-Z BOUNDARY CONDITION	VACUUM

FILE5

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

****** SPACE AND SUPERGROUP INFORMATION ******

116224 WORDS IS THE TOTAL SPACE AVAILABLE.

8817 WORDS WERE USED FOR NON-SUPERGROUP STORAGE.

107407 WORDS OF STORAGE ARE AVAILABLE FOR SUPERGROUPED DATA.

116137 WORDS OF STORAGE ARE AVAILABLE FOR CONSTRUCTING THE SUPERGROUPS.

107369 WORDS OF STORAGE ARE AVAILABLE TO EACH SUPERGROUP.

123 WORDS ARE NEEDED FOR THE LARGEST GROUP.

9068 WORDS OF STORAGE IS SUFFICIENT TO RUN THIS PROBLEM.

10203 WORDS OF STORAGE WILL ALLOW THE PROBLEM TO RUN WITH ONE SUPERGROUP.

SUPERGROUP	STARTING GROUP	ENDING GROUP	XSEC LENGTH	ALBEDO LENGTH	TOTAL LENGTH
1	1	16	122	0	1348

..... 39 IO'S WERE USED IN SUPERGROUPING .....

..... 3 IO'S WERE USED LOADING THE DATA .....

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

REGION	MEDIA NUM	BIAS ID	GEOMETRY DESCRIPTION FOR THOSE UNITS UTILIZED IN THIS PROBLEM							
			----- UNIT 1 -----							
1 CYLINDER	1	1	RADIUS = 5.7480	+Z = 5.3825	-Z = -5.3825	CENTERLINE IS AT	X = .0	Y = .0		
2 CUBOID	0	1	+X = 11.740	-X = -11.740	+Y = 11.740	-Y = -11.740	+Z = 11.375	-Z = -11.375		
			********** GLOBAL **********							
			----- UNIT 2 EXTERNAL TO LATTICE 1 -----							
1 ARRAY NUMBER	1		+X = 23.480	-X = -23.480	+Y = 23.480	-Y = -23.480	+Z = 22.750	-Z = -22.750		
2 CUBOID	2	2	+X = 26.480	-X = -26.480	+Y = 26.480	-Y = -26.480	+Z = 25.750	-Z = -25.750		
3 CUBOID	2	3	+X = 29.480	-X = -29.480	+Y = 29.480	-Y = -29.480	+Z = 28.750	-Z = -28.750		
4 CUBOID	2	4	+X = 32.480	-X = -32.480	+Y = 32.480	-Y = -32.480	+Z = 31.750	-Z = -31.750		
5 CUBOID	2	5	+X = 35.480	-X = -35.480	+Y = 35.480	-Y = -35.480	+Z = 34.750	-Z = -34.750		
6 CUBOID	2	6	+X = 38.480	-X = -38.480	+Y = 38.480	-Y = -38.480	+Z = 37.750	-Z = -37.750		
7 CUBOID	2	7	+X = 38.720	-X = -38.720	+Y = 38.720	-Y = -38.720	+Z = 37.990	-Z = -37.990		

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL
VOLUMES FOR THOSE UNITS UTILIZED IN THIS PROBLEM

UNIT	REGION	GEOMETRY REGION	VOLUME	CUMULATIVE VOLUME
1	1	1	1.11737E+03 CM**3	1.11737E+03 CM**3
	2	2	1.14249E+04 CM**3	1.25423E+04 CM**3
SURROUNDING GEOMETRY VOLUMES -		GEOMETRY REGION	3 IS AN ARRAY PLACEMENT BOUNDARY REGION	
2	1	3	1.00338E+05 CM**3	1.00338E+05 CM**3
	2	4	4.41067E+04 CM**3	1.44445E+05 CM**3
	3	5	5.54410E+04 CM**3	1.99886E+05 CM**3
	4	6	6.80709E+04 CM**3	2.67957E+05 CM**3
	5	7	8.19977E+04 CM**3	3.49955E+05 CM**3
	6	8	9.72196E+04 CM**3	4.47174E+05 CM**3
	7	9	8.47369E+03 CM**3	4.55648E+05 CM**3

UNIT	USES	REGION	TOTAL VOLUME
1	8	1	8.93896E+03 CM**3
		2	9.13994E+04 CM**3
2	1	1	1.00338E+05 CM**3
		2	4.41067E+04 CM**3
		3	5.54410E+04 CM**3
		4	6.80709E+04 CM**3
		5	8.19977E+04 CM**3
		6	9.72196E+04 CM**3
		7	8.47369E+03 CM**3

BIASING INFORMATION

WEIGHTING INTERVALS 1 TO 6 FOR PARAFFIN ,MAT ID= 400 WILL BE USED FOR BIAS ID'S 2 TO 7

A DEFAULT WEIGHT OF 0.500 WILL BE USED FOR ALL OTHER BIAS ID'S.

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

GROUP DEPENDENT WEIGHTS

ENERGY GROUP	BIAS ID 1	BIAS ID 2	BIAS ID 3	BIAS ID 4	BIAS ID 5	BIAS ID 6	BIAS ID 7
1	5.00000E-01	5.96000E-01	8.46000E-01	1.28000E+00	2.08000E+00	3.51000E+00	6.08000E+00
2	5.00000E-01	5.89000E-01	8.73000E-01	1.50000E+00	2.87000E+00	5.88000E+00	1.26000E+01
3	5.00000E-01	5.83000E-01	9.15000E-01	1.80000E+00	4.17000E+00	1.07000E+01	2.94000E+01
4	5.00000E-01	5.77000E-01	9.72000E-01	2.17000E+00	5.80000E+00	1.73000E+01	5.53000E+01
5	5.00000E-01	5.94000E-01	1.14000E+00	2.99000E+00	9.42000E+00	3.28000E+01	1.20000E+02
6	5.00000E-01	6.40000E-01	1.44000E+00	4.46000E+00	1.61000E+01	6.19000E+01	2.42000E+02
7	5.00000E-01	6.71000E-01	1.63000E+00	5.38000E+00	2.02000E+01	7.86000E+01	3.09000E+02
8	5.00000E-01	6.84000E-01	1.75000E+00	6.02000E+00	2.30000E+01	9.01000E+01	3.54000E+02
9	5.00000E-01	6.87000E-01	1.84000E+00	6.57000E+00	2.54000E+01	9.97000E+01	3.93000E+02
10	5.00000E-01	6.73000E-01	1.88000E+00	6.95000E+00	2.71000E+01	1.07000E+02	4.20000E+02
11	5.00000E-01	6.48000E-01	1.89000E+00	7.12000E+00	2.79000E+01	1.10000E+02	4.32000E+02
12	5.00000E-01	6.31000E-01	1.95000E+00	7.48000E+00	2.94000E+01	1.16000E+02	4.55000E+02
13	5.00000E-01	7.68000E-01	2.65000E+00	1.03000E+01	4.05000E+01	1.59000E+02	6.28000E+02
14	5.00000E-01	8.18000E-01	2.97000E+00	1.16000E+01	4.56000E+01	1.80000E+02	7.07000E+02
15	5.00000E-01	8.24000E-01	3.05000E+00	1.19000E+01	4.70000E+01	1.85000E+02	7.29000E+02
16	5.00000E-01	8.91000E-01	3.52000E+00	1.39000E+01	5.46000E+01	2.15000E+02	8.47000E+02

..... 487 IO'S WERE USED IN KENO-V BEFORE TRACKING .....

..... 0.01517MINUTES WERE USED PROCESSING DATA. .....

VOLUME FRACTION OF FISSILE MATERIAL IN THE CORE= 0.89088E-01

START TYPE 0 WAS USED.

THE NEUTRONS WERE STARTED WITH A FLAT DISTRIBUTION IN A CUBOID DEFINED BY:
+X= 2.34800E+01 -X=-2.34800E+01 +Y= 2.34800E+01 -Y=-2.34800E+01 +Z= 2.27500E+01 -Z=-2.27500E+01

0.00767 MINUTES WERE REQUIRED FOR STARTING. TOTAL ELAPSED TIME IS 0.02283 MINUTES.

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

GENERATION	GENERATION K-EFFECTIVE	ELAPSED TIME MINUTES	AVERAGE K-EFFECTIVE	AVG K-EFF DEVIATION	MATRIX K-EFFECTIVE	MATRIX K-EFF DEVIATION
1	1.01565E+00	5.03333E-02	1.00000E+00	0.0	0.0	0.0
2	9.58337E-01	7.96666E-02	1.00000E+00	0.0	0.0	0.0
3	1.08134E+00	1.05000E-01	1.08134E+00	0.0	0.0	0.0
4	9.93345E-01	1.34833E-01	1.03734E+00	4.39887E-02	0.0	0.0
5	1.08738E+00	1.68333E-01	1.05402E+00	3.03913E-02	0.0	0.0
6	9.84501E-01	1.91667E-01	1.03664E+00	2.76501E-02	0.0	0.0
7	9.55051E-01	2.21667E-01	1.02032E+00	2.69175E-02	0.0	0.0
8	1.05417E+00	2.47667E-01	1.02596E+00	2.26933E-02	0.0	0.0
9	9.95418E-01	2.78500E-01	1.02160E+00	1.96691E-02	0.0	0.0
10	1.03861E+00	3.11833E-01	1.02373E+00	1.71664E-02	0.0	0.0
11	9.39917E-01	3.44500E-01	1.01441E+00	1.77771E-02	0.0	0.0
12	9.76407E-01	3.74833E-01	1.01061E+00	1.63507E-02	0.0	0.0
13	1.05314E+00	4.04667E-01	1.01448E+00	1.52856E-02	0.0	0.0
14	8.98528E-01	4.35833E-01	1.00448E+00	1.71639E-02	0.0	0.0
15	9.59359E-01	4.61500E-01	1.00101E+00	1.61674E-02	0.0	0.0
16	9.88133E-01	4.91667E-01	1.00009E+00	1.49925E-02	0.0	0.0
17	9.89243E-01	5.21333E-01	9.99369E-01	1.39755E-02	0.0	0.0
18	1.05609E+00	5.50833E-01	1.00291E+00	1.35504E-02	0.0	0.0
19	1.03455E+00	5.80833E-01	1.00477E+00	1.28632E-02	0.0	0.0
20	9.61870E-01	6.10667E-01	1.00350E+00	1.21973E-02	0.0	0.0
21	9.67911E-01	6.40000E-01	1.00163E+00	1.16916E-02	0.0	0.0
22	9.64995E-01	6.70500E-01	9.99795E-01	1.12397E-02	0.0	0.0
23	1.00624E+00	7.05333E-01	1.00010E+00	1.06977E-02	0.0	0.0
24	1.01009E+00	7.39167E-01	1.00056E+00	1.02112E-02	0.0	0.0
25	1.03669E+00	7.65333E-01	1.00213E+00	9.88475E-03	0.0	0.0
26	1.02542E+00	7.95667E-01	1.00310E+00	9.51182E-03	0.0	0.0
27	1.11912E+00	8.32500E-01	1.00774E+00	1.02365E-02	0.0	0.0
28	1.03849E+00	8.58167E-01	1.00892E+00	9.90525E-03	0.0	0.0
29	9.85307E-01	8.88333E-01	1.00805E+00	9.57215E-03	0.0	0.0
30	9.63828E-01	9.24167E-01	1.00647E+00	9.35743E-03	0.0	0.0
31	9.10580E-01	9.61167E-01	1.00316E+00	9.61803E-03	0.0	0.0
32	9.34550E-01	9.84500E-01	1.00087E+00	9.57004E-03	0.0	0.0
33	9.59738E-01	1.01733E+00	9.99545E-01	9.34934E-03	0.0	0.0
34	9.96881E-01	1.04833E+00	9.99461E-01	9.05414E-03	0.0	0.0
35	9.07315E-01	1.07267E+00	9.96668E-01	9.20984E-03	0.0	0.0
36	9.29537E-01	1.10600E+00	9.94694E-01	9.15012E-03	0.0	0.0
37	1.06221E+00	1.14400E+00	9.96623E-01	9.09278E-03	0.0	0.0
38	1.00260E+00	1.17417E+00	9.96788E-01	8.83776E-03	0.0	0.0
39	9.36503E-01	1.20300E+00	9.95159E-01	8.74932E-03	0.0	0.0
40	9.85262E-01	1.23267E+00	9.94898E-01	8.51972E-03	0.0	0.0
41	1.00099E+00	1.26633E+00	9.95054E-01	8.30040E-03	0.0	0.0
42	1.03959E+00	1.29367E+00	9.96167E-01	8.16667E-03	0.0	0.0
43	9.54605E-01	1.31867E+00	9.95153E-01	8.02978E-03	0.0	0.0
44	8.82557E-01	1.34917E+00	9.92472E-01	8.20228E-03	0.0	0.0
45	1.02539E+00	1.37333E+00	9.93237E-01	8.12435E-03	0.0	0.0
46	1.06578E+00	1.40383E+00	9.94886E-01	8.10724E-03	0.0	0.0
47	9.77493E-01	1.42967E+00	9.94499E-01	7.93525E-03	0.0	0.0
48	9.76965E-01	1.46500E+00	9.94118E-01	7.77026E-03	0.0	0.0
49	1.02992E+00	1.48850E+00	9.94879E-01	7.64111E-03	0.0	0.0
50	9.84225E-01	1.51400E+00	9.94658E-01	7.48318E-03	0.0	0.0
51	1.04253E+00	1.54083E+00	9.95634E-01	7.39408E-03	0.0	0.0
52	1.03073E+00	1.57317E+00	9.96336E-01	7.27890E-03	0.0	0.0

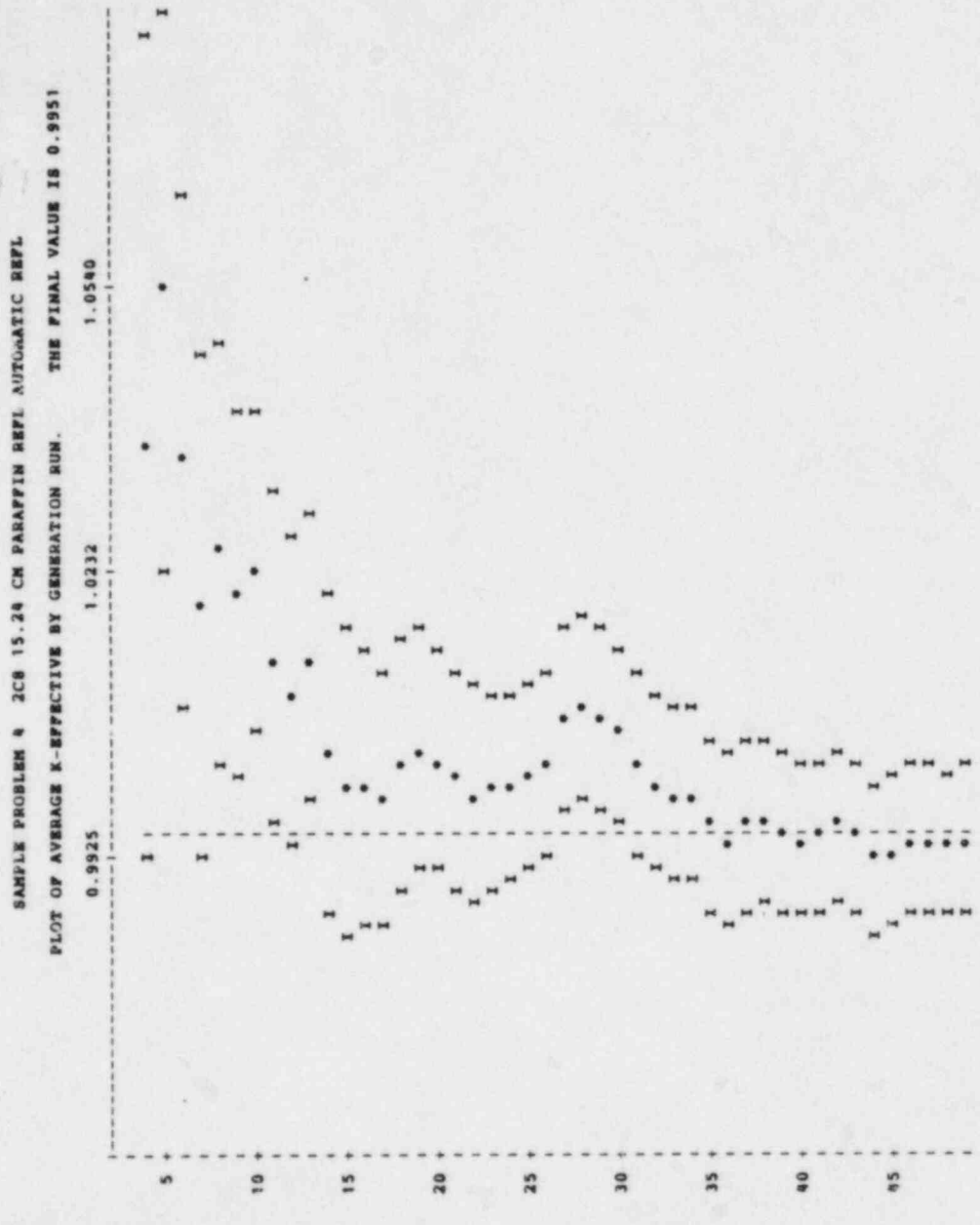
53	9.84319E-01	1.60700E+00	9.96102E-01	7.13860E-03	0.0	0.0
54	9.58511E-01	1.63683E+00	9.95377E-01	7.03777E-03	0.0	0.0
55	9.95938E-01	1.66900E+00	9.95387E-01	6.90410E-03	0.0	0.0
56	9.28779E-01	1.69650E+00	9.94154E-01	6.88666E-03	0.0	0.0
57	9.38168E-01	1.72817E+00	9.93136E-01	6.83691E-03	0.0	0.0
58	1.07117E+00	1.75967E+00	9.94529E-01	6.85691E-03	0.0	0.0
59	9.28319E-01	1.78750E+00	9.93367E-01	6.83500E-03	0.0	0.0
60	1.00511E+00	1.82033E+00	9.93570E-01	6.71922E-03	0.0	0.0
61	1.09966E+00	1.85067E+00	9.95368E-01	6.84487E-03	0.0	0.0
62	1.07089E+00	1.88150E+00	9.96626E-01	6.84671E-03	0.0	0.0
63	9.45070E-01	1.91100E+00	9.95781E-01	6.78613E-03	0.0	0.0
64	8.81176E-01	1.95400E+00	9.93932E-01	6.92724E-03	0.0	0.0
65	1.02855E+00	1.98867E+00	9.94482E-01	6.83889E-03	0.0	0.0
66	1.01401E+00	2.01483E+00	9.94787E-01	6.73787E-03	0.0	0.0
67	1.08307E+00	2.04833E+00	9.96145E-01	6.77098E-03	0.0	0.0
68	9.77982E-01	2.07600E+00	9.95870E-01	6.67302E-03	0.0	0.0
69	1.00994E+00	2.10600E+00	9.96080E-01	6.57588E-03	0.0	0.0
70	9.35182E-01	2.13350E+00	9.95184E-01	6.54004E-03	0.0	0.0
71	1.01508E+00	2.16150E+00	9.95472E-01	6.45107E-03	0.0	0.0
72	1.02613E+00	2.19233E+00	9.95910E-01	6.37332E-03	0.0	0.0
73	9.55288E-01	2.21900E+00	9.95338E-01	6.30924E-03	0.0	0.0
74	1.09101E+00	2.25350E+00	9.96667E-01	6.36126E-03	0.0	0.0
75	1.01554E+00	2.28083E+00	9.96925E-01	6.27895E-03	0.0	0.0
76	9.88208E-01	2.30683E+00	9.96807E-01	6.19454E-03	0.0	0.0
77	1.06142E+00	2.33967E+00	9.97669E-01	6.17202E-03	0.0	0.0
78	9.21536E-01	2.37567E+00	9.96667E-01	6.17220E-03	0.0	0.0
79	1.02894E+00	2.40950E+00	9.97086E-01	6.10615E-03	0.0	0.0
80	9.88233E-01	2.43617E+00	9.96972E-01	6.02861E-03	0.0	0.0
81	1.05682E+00	2.46350E+00	9.97729E-01	5.99997E-03	0.0	0.0
82	8.71890E-01	2.49083E+00	9.96156E-01	6.12985E-03	0.0	0.0
83	9.46059E-01	2.52183E+00	9.95538E-01	6.08536E-03	0.0	0.0
84	1.02841E+00	2.55200E+00	9.95939E-01	6.02415E-03	0.0	0.0
85	9.79316E-01	2.58367E+00	9.95738E-01	5.95456E-03	0.0	0.0
86	1.09493E+00	2.60517E+00	9.96919E-01	6.00052E-03	0.0	0.0
87	9.58743E-01	2.63500E+00	9.96470E-01	5.94641E-03	0.0	0.0
88	1.04522E+00	2.66400E+00	9.97037E-01	5.90414E-03	0.0	0.0
89	9.88842E-01	2.69100E+00	9.96942E-01	5.83686E-03	0.0	0.0
90	9.52501E-01	2.72117E+00	9.96437E-01	5.79223E-03	0.0	0.0
91	1.05262E+00	2.75850E+00	9.97069E-01	5.76146E-03	0.0	0.0
92	9.81846E-01	2.78283E+00	9.96899E-01	5.69975E-03	0.0	0.0
93	1.01493E+00	2.81383E+00	9.97097E-01	5.64048E-03	0.0	0.0
94	1.01321E+00	2.84483E+00	9.97272E-01	5.58160E-03	0.0	0.0
95	9.61908E-01	2.87417E+00	9.96892E-01	5.53447E-03	0.0	0.0
96	9.21153E-01	2.90033E+00	9.96086E-01	5.53441E-03	0.0	0.0
97	9.20692E-01	2.93133E+00	9.95293E-01	5.53316E-03	0.0	0.0
98	1.01148E+00	2.96400E+00	9.95461E-01	5.47774E-03	0.0	0.0
99	9.68195E-01	2.99733E+00	9.95180E-01	5.42830E-03	0.0	0.0
100	1.00658E+00	3.02583E+00	9.95296E-01	5.37390E-03	0.0	0.0
101	1.03864E+00	3.05933E+00	9.95734E-01	5.33741E-03	0.0	0.0
102	8.80392E-01	3.08300E+00	9.94581E-01	5.40823E-03	0.0	0.0
103	1.05008E+00	3.11683E+00	9.95130E-01	5.38284E-03	0.0	0.0

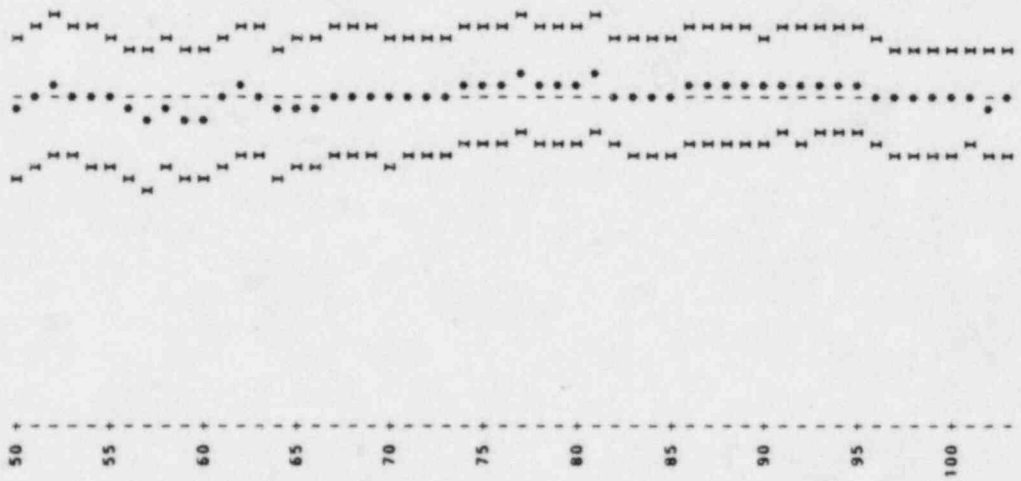
KENO MESSAGE NUMBER K5-123 EXECUTION TERMINATED DUE TO COMPLETION OF THE SPECIFIED NUMBER OF GENERATIONS.

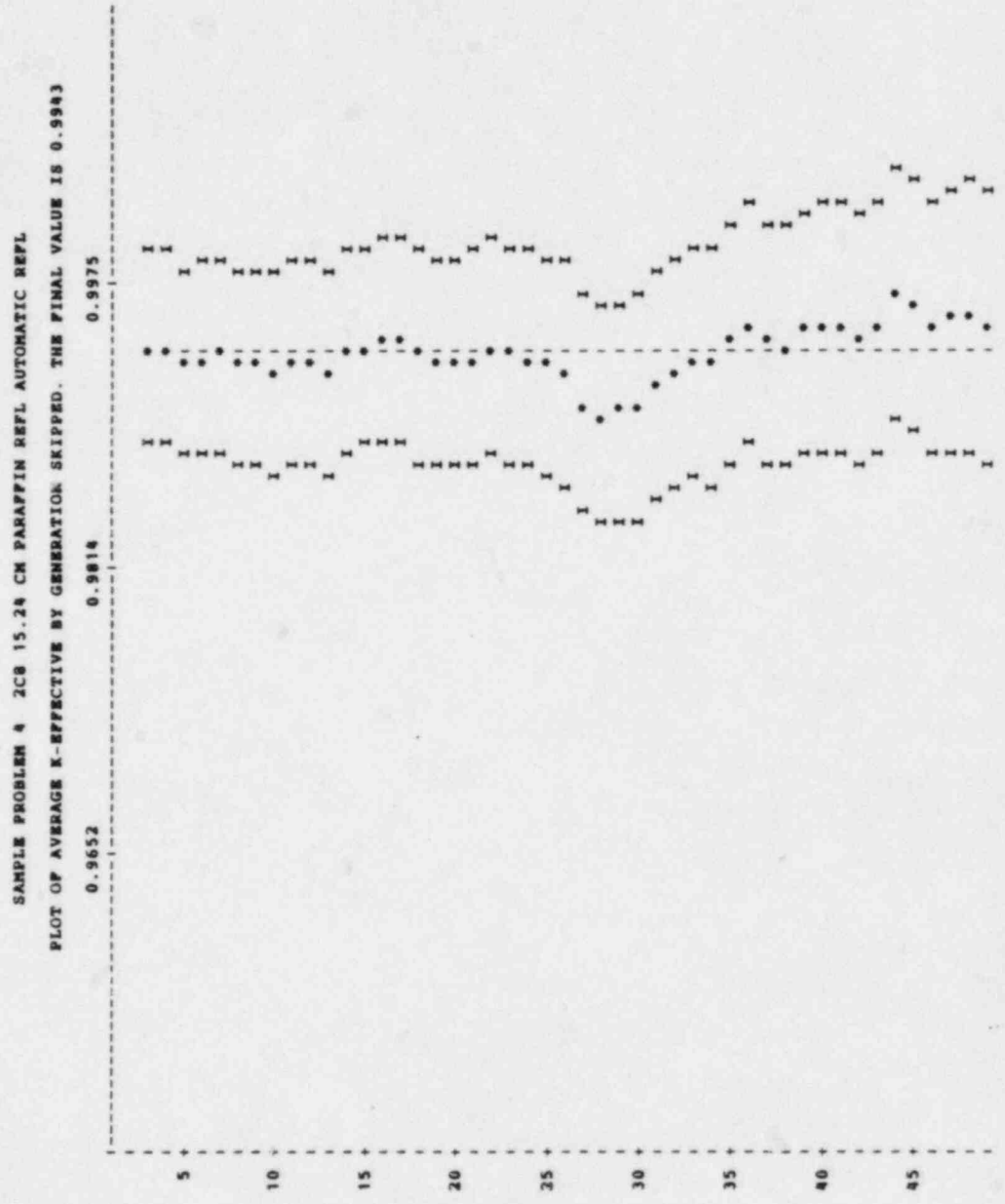
SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

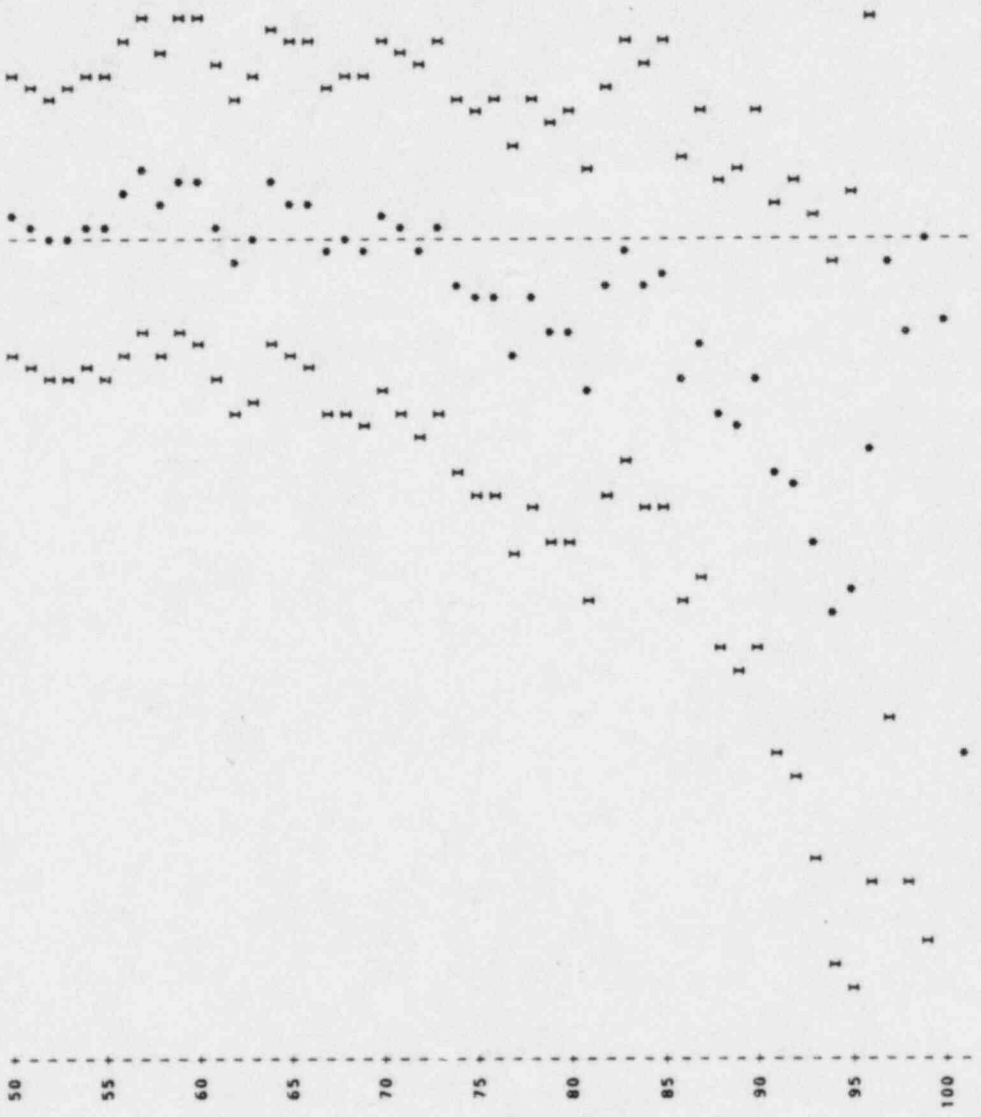
LIFETIME = 1.68757E-04 + OR - 1.46466E-05 GENERATION TIME = 8.58956E-05 + OR - 9.99888E-07

NO. OF INITIAL GENERATIONS SKIPPED	AVERAGE K-EFFECTIVE	DEVIATION	67 PER CENT CONFIDENCE INTERVAL	95 PER CENT CONFIDENCE INTERVAL	99 PER CENT CONFIDENCE INTERVAL	NUMBER OF HISTORIES
3	0.99427	+ OR - 0.00536	0.98891 TO 0.99963	0.98355 TO 1.00499	0.97818 TO 1.01035	30000
4	0.99428	+ OR - 0.00542	0.98886 TO 0.99969	0.98345 TO 1.00511	0.97803 TO 1.01052	29700
5	0.99333	+ OR - 0.00539	0.98794 TO 0.99871	0.98255 TO 1.00410	0.97717 TO 1.00949	29400
6	0.99342	+ OR - 0.00544	0.98798 TO 0.99886	0.98254 TO 1.00430	0.97709 TO 1.00974	29100
7	0.99382	+ OR - 0.00548	0.98833 TO 0.99930	0.98285 TO 1.00478	0.97737 TO 1.01027	28800
8	0.99318	+ OR - 0.00550	0.98768 TO 0.99869	0.98217 TO 1.00419	0.97667 TO 1.00970	28500
9	0.99316	+ OR - 0.00556	0.98760 TO 0.99872	0.98203 TO 1.00428	0.97647 TO 1.00985	28200
10	0.99267	+ OR - 0.00560	0.98707 TO 0.99827	0.98147 TO 1.00387	0.97587 TO 1.00947	27900
11	0.99324	+ OR - 0.00563	0.98761 TO 0.99888	0.98198 TO 1.00451	0.97635 TO 1.01014	27600
12	0.99343	+ OR - 0.00569	0.98774 TO 0.99912	0.98204 TO 1.00481	0.97635 TO 1.01050	27300
17	0.99439	+ OR - 0.00586	0.98854 TO 1.00025	0.98268 TO 1.00610	0.97682 TO 1.01196	25800
22	0.99398	+ OR - 0.00613	0.98785 TO 1.00011	0.98172 TO 1.00624	0.97559 TO 1.01237	24300
27	0.99098	+ OR - 0.00627	0.98472 TO 0.99725	0.97845 TO 1.00352	0.97218 TO 1.00979	22800
32	0.99271	+ OR - 0.00652	0.98619 TO 0.99922	0.97967 TO 1.00574	0.97315 TO 1.01228	21300
37	0.99434	+ OR - 0.00672	0.98762 TO 1.00106	0.98090 TO 1.00778	0.97418 TO 1.01450	19800
42	0.99445	+ OR - 0.00717	0.98728 TO 1.00162	0.98011 TO 1.00879	0.97294 TO 1.01596	18300
47	0.99564	+ OR - 0.00738	0.98826 TO 1.00302	0.98088 TO 1.01039	0.97351 TO 1.01777	16800
52	0.99395	+ OR - 0.00798	0.98597 TO 1.00193	0.97799 TO 1.00991	0.97001 TO 1.01789	15300
57	0.99752	+ OR - 0.00860	0.98892 TO 1.00611	0.98032 TO 1.01472	0.97172 TO 1.02331	13800
62	0.99294	+ OR - 0.00877	0.98417 TO 1.00171	0.97540 TO 1.01048	0.96663 TO 1.01925	12300
67	0.99330	+ OR - 0.00898	0.98432 TO 1.00228	0.97535 TO 1.01125	0.96637 TO 1.02023	10800
72	0.99337	+ OR - 0.01016	0.98321 TO 1.00354	0.97304 TO 1.01370	0.96288 TO 1.02386	9300
77	0.98781	+ OR - 0.01103	0.97678 TO 0.99885	0.96574 TO 1.00988	0.95471 TO 1.02091	7800
82	0.99122	+ OR - 0.01140	0.97982 TO 1.00263	0.96842 TO 1.01403	0.95701 TO 1.02543	6300
87	0.98802	+ OR - 0.01276	0.97526 TO 1.00078	0.96250 TO 1.01354	0.94974 TO 1.02630	4800
92	0.98066	+ OR - 0.01644	0.96422 TO 0.99710	0.94778 TO 1.01354	0.93134 TO 1.02998	3300
97	0.99256	+ OR - 0.02527	0.96729 TO 1.01783	0.94201 TO 1.04311	0.91674 TO 1.06838	1800


SAMPLE PROBLEM 6 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL
PLOT OF AVERAGE K-EFFECTIVE BY GENERATION RUN. THE FINAL VALUE IS 0.9951
0.9925
1.0232
1.0540
5
10
15
20
25
30
35
40
45


50
55
60
65
70
75
80
85
90
95
100


SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL
PLOT OF AVERAGE K-EFFECTIVE BY GENERATION SKIPPED. THE FINAL VALUE IS 0.9943
0.9652
0.9814
0.9975
5
10
15
20
25
30
35
40
45


50
55
60
65
70
75
80
85
90
95
100

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

SKIPPING 3 GENERATIONS

GROUP	FISSION FRACTION	UNIT	REGION	FISSIONS	PERCENT DEVIATION	ABSORPTIONS	PERCENT DEVIATION	LEAKAGE	PERCENT DEVIATION
1	0.1102			1.09529D-01	1.7513	3.86782D-02	1.7513	5.19813D-03	8.0358
2	0.1789			1.77840D-01	1.2886	7.24442D-02	1.2886	6.65555D-03	10.0246
3	0.1020			1.01394D-01	1.5245	4.38716D-02	1.5245	2.19342D-03	19.9744
4	0.1566			1.55696D-01	1.2057	7.08092D-02	1.2057	2.81881D-03	29.9144
5	0.1341			1.33356D-01	1.5553	6.33564D-02	1.5553	3.13666D-03	3[illegible].0134
6	0.0403			4.00253D-02	3.1547	2.07969D-02	3.1547	3.51818D-03	55.3551
7	0.0114			1.13795D-02	5.7035	6.13824D-03	5.7035	1.09337D-03	100.0000
8	0.0121			1.20770D-02	5.1660	7.07082D-03	5.1249	0.0	0.0
9	0.0122			1.21078D-02	6.1409	7.62082D-03	5.9748	3.15609D-03	73.6653
10	0.0078			7.75391D-03	7.6151	5.44002D-03	7.1971	0.0	0.0
11	0.0061			6.07253D-03	7.3250	5.21466D-03	6.5984	0.0	0.0
12	0.0049			4.92043D-03	7.3865	5.82071D-03	6.1135	3.31943D-03	100.0000
13	0.0075			7.47180D-03	8.8195	5.47517D-03	6.4575	2.06236D-03	100.0000
14	0.0075			7.43386D-03	9.6342	5.44577D-03	5.9981	0.0	0.0
15	0.0200			1.98818D-02	5.5792	2.07703D-02	3.1042	2.03713D-03	100.0000
16	0.1884			1.87335D-01	1.8031	6.01514D-01	5.1328	3.41204D-02	83.9561
SYSTEM TOTAL =				9.94275D-01	0.5392	9.80467D-01	3.1608	6.93055D-02	42.1817

ELAPSED TIME 3.11850 MINUTES

RANDUM NUMBER- 1F30B7469D89

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

**** FISSION DENSITIES ****

UNIT	REGION	FISSION DENSITY	PERCENT DEVIATION	TOTAL FISSIONS
1	1	1.112E-04	0.54	9.94 E-01
	2	0.0	0.0	0.0
GLOBAL UNIT				
2	1	0.0	0.0	0.0
	2	0.0	0.0	0.0
	3	0.0	0.0	0.0
	4	0.0	0.0	0.0
	5	0.0	0.0	0.0
	6	0.0	0.0	0.0
	7	0.0	0.0	0.0

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

FLUXES FOR UNIT 1

GROUP	REGION 1 FLUX	REGION 1 PERCENT DEVIATION	REGION 2 FLUX	REGION 2 PERCENT DEVIATION
1	7.446E-05	1.80	2.342E-05	2.04
2	1.338E-04	1.08	4.319E-05	1.39
3	8.277E-05	1.43	2.615E-05	1.65
4	1.281E-04	1.01	4.069E-05	1.29
5	9.434E-05	1.45	3.180E-05	1.41
6	1.639E-05	3.17	1.205E-05	3.27
7	2.450E-06	6.39	7.217E-06	3.92
8	1.541E-06	6.18	6.994E-06	3.53
9	6.930E-07	8.18	6.315E-06	3.60
10	2.074E-07	9.69	3.921E-06	5.42
11	1.506E-07	9.53	3.715E-06	5.14
12	1.596E-07	10.80	3.933E-06	5.20
13	2.656E-07	11.54	3.334E-06	5.97
14	1.025E-07	12.19	2.813E-06	5.07
15	1.054E-07	6.61	7.444E-06	4.35
16	3.663E-07	2.63	6.837E-05	2.18

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

FLUXES FOR GLOBAL UNIT

	REGION 1		REGION 2		REGION 3		REGION 4		REGION 5		REGION 6	
GROUP	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION	FLUX	PERCENT DEVIATION
1	0.0	0.0	8.682E-06	2.06	3.534E-06	2.78	1.493E-06	3.87	7.007E-07	5.38	3.128E-07	7.53
2	0.0	0.0	1.626E-05	1.39	5.955E-06	2.01	2.403E-06	3.09	1.038E-06	4.78	4.718E-07	7.62
3	0.0	0.0	9.804E-06	1.57	3.369E-06	2.14	1.218E-06	3.73	5.243E-07	7.52	2.278E-07	15.09
4	0.0	0.0	1.593E-05	1.20	5.432E-06	1.86	2.031E-06	2.86	8.396E-07	7.02	2.561E-07	13.34
5	0.0	0.0	1.815E-05	0.98	6.880E-06	1.80	2.588E-06	3.96	1.272E-06	8.90	4.962E-07	18.81
6	0.0	0.0	1.046E-05	1.08	4.428E-06	1.76	1.793E-06	4.58	8.347E-07	10.85	4.000E-07	25.59
7	0.0	0.0	8.089E-06	1.09	3.703E-06	1.79	1.474E-06	5.44	5.944E-07	12.66	2.558E-07	26.93
8	0.0	0.0	7.031E-06	1.40	3.629E-06	1.99	1.285E-06	5.61	6.016E-07	16.60	3.001E-07	28.62
9	0.0	0.0	6.571E-06	1.31	3.587E-06	2.08	1.469E-06	5.61	7.140E-07	16.28	2.968E-07	31.37
10	0.0	0.0	4.526E-06	1.68	2.524E-06	2.27	1.042E-06	5.71	3.751E-07	16.78	1.842E-07	40.47
11	0.0	0.0	3.961E-06	1.39	2.226E-06	2.90	9.294E-07	7.74	5.139E-07	17.78	1.747E-07	38.92
12	0.0	0.0	4.132E-06	1.62	2.531E-06	2.90	1.018E-06	7.20	5.009E-07	17.24	4.263E-07	39.89
13	0.0	0.0	3.750E-06	1.55	2.220E-06	2.54	9.260E-07	8.70	3.968E-07	23.03	1.528E-07	41.46
14	0.0	0.0	3.051E-06	1.82	1.884E-06	3.48	8.313E-07	8.98	4.364E-07	20.61	1.942E-07	45.92
15	0.0	0.0	8.823E-06	1.83	5.868E-06	3.38	2.398E-06	8.04	1.233E-06	20.02	2.543E-07	36.77
16	0.0	0.0	1.252E-04	1.75	1.206E-04	3.51	6.945E-05	9.92	3.610E-05	16.44	1.381E-05	28.66

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

FLUXES FOR GLOBAL UNIT

	REGION 7	
GROUP	FLUX	PERCENT DEVIATION
1	1.808E-07	8.06
2	2.605E-07	11.77
3	9.614E-08	21.10
4	1.521E-07	47.32
5	1.460E-07	36.43
6	1.137E-07	55.34
7	3.327E-08	100.00
8	0.0	0.0
9	1.178E-07	74.17
10	0.0	0.0
11	0.0	0.0
12	1.025E-07	100.00
13	7.604E-08	100.00
14	0.0	0.0
15	5.779E-08	100.00
16	1.269E-06	65.01

SAMPLE PROBLEM 4 2C8 15.24 CM PARAFFIN REFL AUTOMATIC REFL

FREQUENCY FOR GENERATIONS 4 TO 103

```
0.8613 TO 0.8844      ****
0.8844 TO 0.9075      **
0.9075 TO 0.9306      *******
0.9306 TO 0.9537      ********
0.9537 TO 0.9768      ************
0.9768 TO 0.9999      ***********************
0.9999 TO 1.0230      *************
1.0230 TO 1.0461      ******************
1.0461 TO 1.0692      **********
1.0692 TO 1.0922      *****
1.0922 TO 1.1153      **
1.1153 TO 1.1384      *
```

FREQUENCY FOR GENERATIONS 29 TO 103

```
0.8613 TO 0.8844      ****
0.8844 TO 0.9075      *
0.9075 TO 0.9306      *******
0.9306 TO 0.9537      *******
0.9537 TO 0.9768      ********
0.9768 TO 0.9999      ***************
0.9999 TO 1.0230      ************
1.0230 TO 1.0461      ************
1.0461 TO 1.0692      ******
1.0692 TO 1.0922      ****
1.0922 TO 1.1153      **
1.1153 TO 1.1384
```

FREQUENCY FOR GENERATIONS 54 TO 103

```
0.8613 TO 0.8844      ***
0.8844 TO 0.9075
0.9075 TO 0.9306      *****
0.9306 TO 0.9537      *****
0.9537 TO 0.9768      *****
0.9768 TO 0.9999      *******
0.9999 TO 1.0230      *********
1.0230 TO 1.0461      ******
1.0461 TO 1.0692      ****
1.0692 TO 1.0922      ****
1.0922 TO 1.1153      **
1.1153 TO 1.1384
```

FREQUENCY FOR GENERATIONS 79 TO 103

```
0.8613 TO 0.8844      **
0.8844 TO 0.9075
0.9075 TO 0.9306      **
0.9306 TO 0.9537      **
0.9537 TO 0.9768      ***
0.9768 TO 0.9999      ****
0.9999 TO 1.0230      ****
1.0230 TO 1.0461      ****
1.0461 TO 1.0692      ***
1.0692 TO 1.0922
1.0922 TO 1.1153      *
1.1153 TO 1.1384
```

****

CONGRATULATIONS! YOU HAVE SUCCESSFULLY TRAVERSED THE PERILOUS PATH THROUGH KENO V IN 3.11917 MINUTES

****

FILE 20

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

****** NUMERIC PARAMETERS ******

TME	MAXIMUM PROBLEM TIME (MIN)	0.75
TBA	TIME PER GENERATION (MIN)	0.50
GEN	NUMBER OF GENERATIONS	102
NPG	NUMBER PER GENERATION	300
NSK	NUMBER OF GENERATIONS TO BE SKIPPED	3
BEG	BEGINNING GENERATION NUMBER	1
RES	GENERATIONS BETWEEN CHECKPOINTS	0
X1D	NUMBER OF EXTRA 1-D CROSS SECTIONS	0
NBK	NEUTRON BANK SIZE	325
XNB	EXTRA POSITIONS IN NEUTRON BANK	0
NFB	FISSION BANK SIZE	300
XFB	EXTRA POSITIONS IN FISSION BANK	0
WTA	DEFAULT VALUE OF WEIGHT AVERAGE	0.5000
WTH	WEIGHT HIGH FOR SPLITTING	3.0000
WTL	WEIGHT LOW FOR RUSSIAN ROULETTE	0.3333
RND	STARTING RANDUM NUMBER	00071AF0498D
NB8	NUMBER OF D.A. BLOCKS ON UNIT 8	200
NL8	LENGTH OF D.A. BLOCKS ON UNIT 8	512
ADJ	MODE OF CALCULATION	FORWARD
	INPUT DATA WRITTEN ON RESTART UNIT	NO
	BINARY DATA INTERFACE	NO

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

****** LOGICAL PARAMETERS ******

RUN	EXECUTE PROBLEM AFTER CHECKING DATA	YES	PLT	PLOT PICTURE MAP(S)	YES
FLX	COMPUTE FLUX	NO	FDN	COMPUTE FISSION DENSITIES	NO
SMU	COMPUTE AVG UNIT SELF-MULTIPLICATION	NO	NUB	COMPUTE NU-BAR & AVG FISSION GROUP	NO
MKU	COMPUTE MATRIX K-EFF BY UNIT NUMBER	NO	MKP	COMPUTE MATRIX K-EFF BY UNIT LOCATION	NO
CKU	COMPUTE COFACTOR K-EFF BY UNIT NUMBER	NO	CKP	COMPUTE COFACTOR K-EFF BY UNIT LOCATION	NO
FMU	PRINT FISS PROD MATRIX BY UNIT NUMBER	NO	FMP	PRINT FISS PROD MATRIX BY UNIT LOCATION	NO
MKH	COMPUTE MATRIX K-EFF BY HOLE NUMBER	NO	MKA	COMPUTE MATRIX K-EFF BY ARRAY NUMBER	NO
CKH	COMPUTE COFACTOR K-EFF BY HOLE NUMBER	NO	CKA	COMPUTE COFACTOR K-EFF BY ARRAY NUMBER	NO
FMH	PRINT FISS PROD MATRIX BY HOLE NUMBER	NO	FMA	PRINT FISS PROD MATRIX BY ARRAY NUMBER	NO
HHL	COLLECT MATRIX BY HIGHEST HOLE LEVEL	NO	HAL	COLLECT MATRIX BY HIGHEST ARRAY LEVEL	NO
AMX	PRINT ALL MIXED CROSS SECTIONS	NO	FAR	PRINT FIS. AND ABS. BY REGION	NO
XS1	PRINT 1-D MIXTURE X-SECTIONS	NO	PAX	PRINT XSEC-ALBEDO CORRELATION TABLES	NO
XS2	PRINT 2-D MIXTURE X-SECTIONS	NO	PWT	PRINT WEIGHT AVERAGE ARRAY	NO
XAP	PRINT MIXTURE ANGLES & PROBABILITIES	NO	PGM	PRINT INPUT GEOMETRY	NO
PKI	PRINT FISSION SPECTRUM	NO	BUG	PRINT DEBUG INFORMATION	NO
P1D	PRINT EXTRA 1-D CROSS SECTIONS	NO	TRK	PRINT TRACKING INFORMATION	NO

PARAMETER INPUT COMPLETED

..... 208 IO'S WERE USED READING THE PARAMETER DATA .....

********** DATA READING COMPLETED **********

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

	UNIT NUMBER	DATA SET NAME	VOLUME NAME	UNIT FUNCTION
XSC	14	SYS83314.T133920.RA000.NFCF11D.R0000013	X3A1C0	MIXED CROSS SECTIONS
ALB	79	E.LMP00000.KENO5.ALBEDOS	NE3330	INPUT ALBEDOS
WTS	80	E.LMP00000.KENO4.WGTS	NE3330	INPUT WEIGHTS
SKT	16	SYS83314.T133920.RA000.NFCF11D.R0000015	X3A1C1	WRITE SCRATCH DATA
LIB	41	E.LMP00000.AMPX.HRXSCS	NE3330	INPUT AMPX WORKING LIBRARY
	8	SYS83314.T133920.RA000.NFCF11D.R0000007	MVSDA2	INPUT DATA DIRECT ACCESS
	9	SYS83314.T133920.RA000.NFCF11D.R0000008	X3A1C1	SUPER GROUPED DIRECT ACCESS
	10	SYS83314.T133920.RA000.NFCF11D.R0000009	X3A1C0	XSEC MIXING DIRECT ACCESS

..... 16 IO'S WERE USED PREPARING INPUT DATA .....

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

MIXING TABLE

NUMBER OF SCATTERING ANGLES = 1
CROSS SECTION MESSAGE THRESHOLD =3.0E-05

ENTRY	MIXTURE	NUCLIDE	DENSITY
1	1	92500	4.48006E-02
2	1	92800	2.65780E-03
3	1	92400	4.82700E-04
4	1	92600	9.57000E-05

CROSS SECTIONS READ FROM THE AMPX WORKING LIBRARY ON UNIT 41

92400	U-234	MIHALCEO MOD OF H-R U-238
92500	U-235 YR	HANSEN ROACH
92600	U-236	MIHALCEO MOD OF H-R U-238
92800	U-238 Y	HANSEN ROACH

..... 129 IO'S WERE USED MIXING CROSS-SECTIONS .....

..... 14 IO'S WERE USED PREPARING THE CROSS SECTIONS .....

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

****** ADDITIONAL INFORMATION ******

NUMBER OF ENERGY GROUPS	16	USE LATTICE GEOMETRY	NO
NO. OF FISSION SPECTRUM SOURCE GROUP	1	GLOBAL ARRAY NUMBER	0
NO. OF SCATTERING ANGLES IN XSECS	1	NUMBER OF UNITS IN THE GLOBAL X DIR.	0
ENTRIES/NEUTRON IN THE NEUTRON BANK	16	NUMBER OF UNITS IN THE GLOBAL Y DIR.	0
ENTRIES/NEUTRON IN THE FISSION BANK	9	NUMBER OF UNITS IN THE GLOBAL Z DIR.	0
NUMBER OF MIXTURES USED	1	USE A GLOBAL REFLECTOR	YES
NUMBER OF BIAS ID'S USED	1	USE NESTED HOLES	NO
NUMBER OF DIFFERENTIAL ALBEDOS USED	0	NUMBER OF HOLES	0
TOTAL INPUT GEOMETRY REGIONS	3	MAXIMUM HOLE NESTING LEVEL	0
NUMBER OF GEOMETRY REGIONS USED	3	USE NESTED ARRAYS	NO
NUMBER OF GEOMETRY UNITS	1	NUMBER OF ARRAYS USED	0
LARGEST ARRAY NUMBER	1	MAXIMUM ARRAY NESTING LEVEL	0
+X BOUNDARY CONDITION	VACUUM	-X BOUNDARY CONDITION	VACUUM
+Y BOUNDARY CONDITION	VACUUM	-Y BOUNDARY CONDITION	VACUUM
+Z BOUNDARY CONDITION	VACUUM	-Z BOUNDARY CONDITION	VACUUM

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

****** SPACE AND SUPERGROUP INFORMATION ******

116224 WORDS IS THE TOTAL SPACE AVAILABLE.

8671 WORDS WERE USED FOR NON-SUPERGROUP STORAGE.

107553 WORDS OF STORAGE ARE AVAILABLE FOR SUPERGROUPED DATA.

116162 WORDS OF STORAGE ARE AVAILABLE FOR CONSTRUCTING THE SUPERGROUPS.

107515 WORDS OF STORAGE ARE AVAILABLE TO EACH SUPERGROUP.

55 WORDS ARE NEEDED FOR THE LARGEST GROUP.

8854 WORDS OF STORAGE IS SUFFICIENT TO RUN THIS PROBLEM.

9239 WORDS OF STORAGE WILL ALLOW THE PROBLEM TO RUN WITH ONE SUPERGROUP.

SUPERGROUP	STARTING GROUP	ENDING GROUP	XSEC LENGTH	ALBEDO LENGTH	TOTAL LENGTH
1	1	16	41	0	530

..... 22 IO'S WERE USED IN SUPERGROUPING .....

..... 2 IO'S WERE USED LOADING THE DATA .....

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

GEOMETRY DESCRIPTION FOR THOSE UNITS UTILIZED IN THIS PROBLEM

GLOBAL UNIT 1

REGION	MEDIA NUM	BIAS ID										
1 CYLINDER	1	1	RADIUS =	8.8900	+Z =	10.109	-Z =	.0	CENTERLINE IS AT	X =	5.0800	Y = .0
2 CYLINDER	0	1	RADIUS =	13.970	+Z =	10.109	-Z =	.0	CENTERLINE IS AT	X =	.0	Y = .0
3 CYLINDER	1	1	RADIUS =	19.050	+Z =	10.109	-Z =	.0	CENTERLINE IS AT	X =	.0	Y = .0

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS
VOLUMES FOR THOSE UNITS UTILIZED IN THIS PROBLEM

UNIT	REGION	GEOMETRY REGION	VOLUME	CUMULATIVE VOLUME
1	1	1	2.50993E+03 CM**3	2.50993E+03 CM**3
	2	2	3.68806E+03 CM**3	6.19799E+03 CM**3
	3	3	5.32718E+03 CM**3	1.15252E+04 CM**3

UNIT	USES	REGION	TOTAL VOLUME
1	1	1	2.50993E+03 CM**3
		2	3.68806E+03 CM**3
		3	5.32718E+03 CM**3

BIASING INFORMATION

A DEFAULT WEIGHT OF 0.500 WILL BE USED FOR ALL BIAS ID'S.

..... 391 IO'S WERE USED IN KENO-V BEFORE TRACKING .....

..... 0.01183MINUTES WERE USED PROCESSING DATA. .....

VOLUME FRACTION OF FISSILE MATERIAL IN THE CORE= 0.68000E+00

START TYPE 0 WAS USED.

THE NEUTRONS WERE STARTED UNIFORMLY THROUGHOUT THE ENTIRE VOLUME DEFINED BY THE OUTERMOST GEOMETRY CARD.

0.00117 MINUTES WERE REQUIRED FOR STARTING. TOTAL ELAPSED TIME IS 0.01300 MINUTES.

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

GENERATION KENO MESSAGE NUMBER	GENERATION K-EFFECTIVE K5-132	ELAPSED TIME MINUTES WARNING....ONLY	AVERAGE K-EFFECTIVE 282 INDEPENDENT	AVG K-EFF DEVIATION FISSION POINTS WERE	MATRIX K-EFFECTIVE GENERATED	MATRIX K-EFF DEVIATION
1	7.93163E-01	1.55000E-02	1.00000E+00	0.0	0.0	0.0
2	9.23795E-01	1.81667E-02	1.00000E+00	0.0	0.0	0.0
3	8.46078E-01	2.06667E-02	8.46078E-01	0.0	0.0	0.0
4	8.68823E-01	2.33333E-02	8.57450E-01	1.13860E-02	0.0	0.0
5	9.48828E-01	2.63333E-02	8.87909E-01	3.11612E-02	0.0	0.0
6	1.00527E+00	2.93333E-02	9.17250E-01	3.66929E-02	0.0	0.0
7	1.00181E+00	3.23333E-02	9.34163E-01	3.30736E-02	0.0	0.0
8	1.03038E+00	3.53333E-02	9.50200E-01	3.14068E-02	0.0	0.0
9	9.88138E-01	3.83333E-02	9.55619E-01	2.70915E-02	0.0	0.0
10	9.24516E-01	4.10000E-02	9.51731E-01	2.37819E-02	0.0	0.0
11	1.05661E+00	4.43333E-02	9.63385E-01	2.39936E-02	0.0	0.0
12	1.02734E+00	4.73333E-02	9.69780E-01	2.23930E-02	0.0	0.0
13	9.98793E-01	5.01667E-02	9.72417E-01	2.04266E-02	0.0	0.0
14	1.06141E+00	5.33333E-02	9.79833E-01	2.00672E-02	0.0	0.0
15	9.15206E-01	5.60000E-02	9.74862E-01	1.91170E-02	0.0	0.0
16	1.01552E+00	5.90000E-02	9.77766E-01	1.79355E-02	0.0	0.0
17	9.73917E-01	6.16667E-02	9.77509E-01	1.66992E-02	0.0	0.0
18	1.05072E+00	6.48333E-02	9.82085E-01	1.62769E-02	0.0	0.0
19	1.00348E+00	6.78333E-02	9.83343E-01	1.53422E-02	0.0	0.0
20	9.30478E-01	7.05000E-02	9.80405E-01	1.47620E-02	0.0	0.0
21	9.82645E-01	7.33333E-02	9.80523E-01	1.39661E-02	0.0	0.0
22	9.31359E-01	7.61666E-02	9.78064E-01	1.34757E-02	0.0	0.0
23	9.57049E-01	7.90000E-02	9.77063E-01	1.28571E-02	0.0	0.0
24	9.62646E-01	8.18333E-02	9.76407E-01	1.22783E-02	0.0	0.0
25	1.06914E+00	8.48333E-02	9.80439E-01	1.24054E-02	0.0	0.0
26	1.03126E+00	8.80000E-02	9.82556E-01	1.20657E-02	0.0	0.0
27	9.40319E-01	9.08333E-02	9.80866E-01	1.16962E-02	0.0	0.0
28	9.71573E-01	9.35000E-02	9.80509E-01	1.12432E-02	0.0	0.0
29	1.03512E+00	9.66666E-02	9.82531E-01	1.10074E-02	0.0	0.0
30	9.66564E-01	9.94999E-02	9.81960E-01	1.06228E-02	0.0	0.0
31	1.05487E+00	1.02500E-01	9.84474E-01	1.05535E-02	0.0	0.0
32	1.04203E+00	1.05667E-01	9.86393E-01	1.03749E-02	0.0	0.0
33	1.07773E+00	1.08667E-01	9.89339E-01	1.04583E-02	0.0	0.0
34	9.50606E-01	1.11500E-01	9.88128E-01	1.01990E-02	0.0	0.0
35	9.30205E-01	1.14167E-01	9.86372E-01	1.00411E-02	0.0	0.0
36	1.00706E+00	1.17000E-01	9.84981E-01	9.76082E-03	0.0	0.0
37	1.04482E+00	1.20000E-01	9.88633E-01	9.62103E-03	0.0	0.0
38	1.04343E+00	1.23167E-01	9.90155E-01	9.47253E-03	0.0	0.0
39	9.53824E-01	1.25833E-01	9.89173E-01	9.26600E-03	0.0	0.0
40	1.05756E+00	1.29000E-01	9.90972E-01	9.19712E-03	0.0	0.0
41	9.84879E-01	1.31833E-01	9.90816E-01	8.95562E-03	0.0	0.0
42	1.06387E+00	1.35000E-01	9.92642E-01	8.92204E-03	0.0	0.0
43	1.01531E+00	1.37833E-01	9.93195E-01	8.71953E-03	0.0	0.0
44	9.49997E-01	1.40500E-01	9.92166E-01	8.57193E-03	0.0	0.0
45	9.28889E-01	1.43167E-01	9.90694E-01	8.49896E-03	0.0	0.0
46	9.49714E-01	1.45833E-01	9.89763E-01	8.35562E-03	0.0	0.0
47	9.96529E-01	1.48667E-01	9.89913E-01	8.16943E-03	0.0	0.0
48	1.03751E+00	1.51667E-01	9.90948E-01	8.05639E-03	0.0	0.0
49	9.79936E-01	1.54500E-01	9.90713E-01	7.88645E-03	0.0	0.0
50	1.00344E+00	1.57333E-01	9.90978E-01	7.72500E-03	0.0	0.0
51	9.53546E-01	1.60167E-01	9.90214E-01	7.60445E-03	0.0	0.0
52	1.03244E+00	1.63167E-01	9.91058E-01	7.49885E-03	0.0	0.0

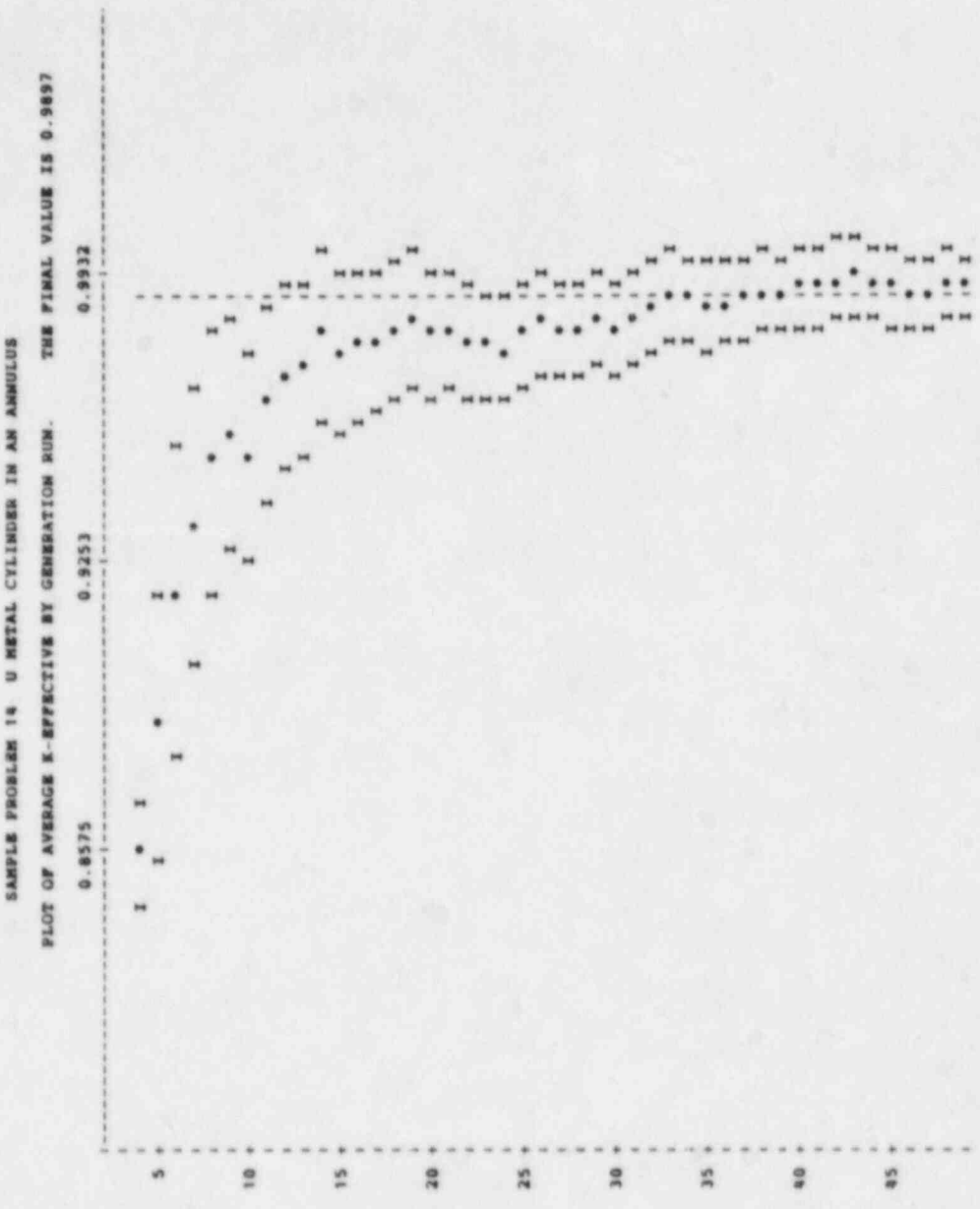
53	9.52890E-01	1.66000E-01	9.90310E-01	7.38871E-03	0.0	0.0
54	9.83362E-01	1.69000E-01	9.90176E-01	7.24674E-03	0.0	0.0
55	9.87221E-01	1.71833E-01	9.90120E-01	7.10908E-03	0.0	0.0
56	9.51756E-01	1.74500E-01	9.89410E-01	7.01249E-03	0.0	0.0
57	9.71109E-01	1.77333E-01	9.89077E-01	6.89190E-03	0.0	0.0
58	1.06287E+00	1.80500E-01	9.90394E-01	6.89513E-03	0.0	0.0
59	9.88891E-01	1.83333E-01	9.90368E-01	6.77353E-03	0.0	0.0
60	1.01286E+00	1.86333E-01	9.90755E-01	6.66696E-03	0.0	0.0
61	1.02195E+00	1.89333E-01	9.91284E-01	6.57478E-03	0.0	0.0
62	9.57094E-01	1.92167E-01	9.90714E-01	6.48924E-03	0.0	0.0
63	9.80636E-01	1.95167E-01	9.90549E-01	6.38441E-03	0.0	0.0
64	9.29799E-01	1.97833E-01	9.89569E-01	6.35673E-03	0.0	0.0
65	1.02171E+00	2.00833E-01	9.90079E-01	6.27599E-03	0.0	0.0
66	9.25756E-01	2.03333E-01	9.89074E-01	6.25835E-03	0.0	0.0
67	9.40234E-01	2.06167E-01	9.88322E-01	6.20593E-03	0.0	0.0
68	1.01976E+00	2.09000E-01	9.88798E-01	6.13087E-03	0.0	0.0
69	9.83964E-01	2.12000E-01	9.88726E-01	6.03903E-03	0.0	0.0
70	1.02224E+00	2.15000E-01	9.89219E-01	5.97015E-03	0.0	0.0
71	1.00379E+00	2.18000E-01	9.89430E-01	5.88676E-03	0.0	0.0
72	9.02219E-01	2.20500E-01	9.88184E-01	5.93474E-03	0.0	0.0
73	9.62485E-01	2.23167E-01	9.87822E-01	5.86192E-03	0.0	0.0
74	9.91216E-01	2.26000E-01	9.87869E-01	5.78011E-03	0.0	0.0
75	9.57117E-01	2.28667E-01	9.87448E-01	5.71606E-03	0.0	0.0
76	9.58339E-01	2.31667E-01	9.87054E-01	5.65233E-03	0.0	0.0
77	9.12621E-01	2.34333E-01	9.86062E-01	5.66415E-03	0.0	0.0
78	1.01407E+00	2.37167E-01	9.86430E-01	5.60128E-03	0.0	0.0
79	9.77682E-01	2.40167E-01	9.86316E-01	5.52938E-03	0.0	0.0
80	9.71371E-01	2.42833E-01	9.86125E-01	5.46163E-03	0.0	0.0
81	9.88163E-01	2.45667E-01	9.85644E-01	5.41361E-03	0.0	0.0
82	9.42932E-01	2.48333E-01	9.85110E-01	5.37208E-03	0.0	0.0
83	1.03578E+00	2.51333E-01	9.85736E-01	5.34202E-03	0.0	0.0
84	1.03370E+00	2.54500E-01	9.86320E-01	5.30908E-03	0.0	0.0
85	1.05271E+00	2.57500E-01	9.87120E-01	5.30560E-03	0.0	0.0
86	1.03900E+00	2.60667E-01	9.87738E-01	5.27864E-03	0.0	0.0
87	1.01564E+00	2.63667E-01	9.88066E-01	5.22653E-03	0.0	0.0
88	9.88397E-01	2.66500E-01	9.88070E-01	5.16565E-03	0.0	0.0
89	9.66922E-01	2.69333E-01	9.87827E-01	5.11179E-03	0.0	0.0
90	1.01021E+00	2.72167E-01	9.88081E-01	5.05963E-03	0.0	0.0
91	1.00037E+00	2.75000E-01	9.88219E-01	5.00448E-03	0.0	0.0
92	1.06000E+00	2.78167E-01	9.89017E-01	5.01242E-03	0.0	0.0
93	9.72230E-01	2.81167E-01	9.88832E-01	4.96032E-03	0.0	0.0
94	1.02006E+00	2.84167E-01	9.89171E-01	4.91796E-03	0.0	0.0
95	1.04449E+00	2.87167E-01	9.89766E-01	4.90089E-03	0.0	0.0
96	1.02859E+00	2.90167E-01	9.90179E-01	4.86605E-03	0.0	0.0
97	9.06424E-01	2.92833E-01	9.89297E-01	4.89474E-03	0.0	0.0
98	1.07447E+00	2.96000E-01	9.90185E-01	4.92414E-03	0.0	0.0
99	1.03094E+00	2.98833E-01	9.90605E-01	4.89145E-03	0.0	0.0
100	9.99459E-01	3.01833E-01	9.90695E-01	4.84218E-03	0.0	0.0
101	9.92134E-01	3.04667E-01	9.90709E-01	4.79305E-03	0.0	0.0
102	9.91279E-01	3.07500E-01	9.90715E-01	4.74486E-03	0.0	0.0
103	8.90621E-01	3.10000E-01	9.89724E-01	4.80126E-03	0.0	0.0

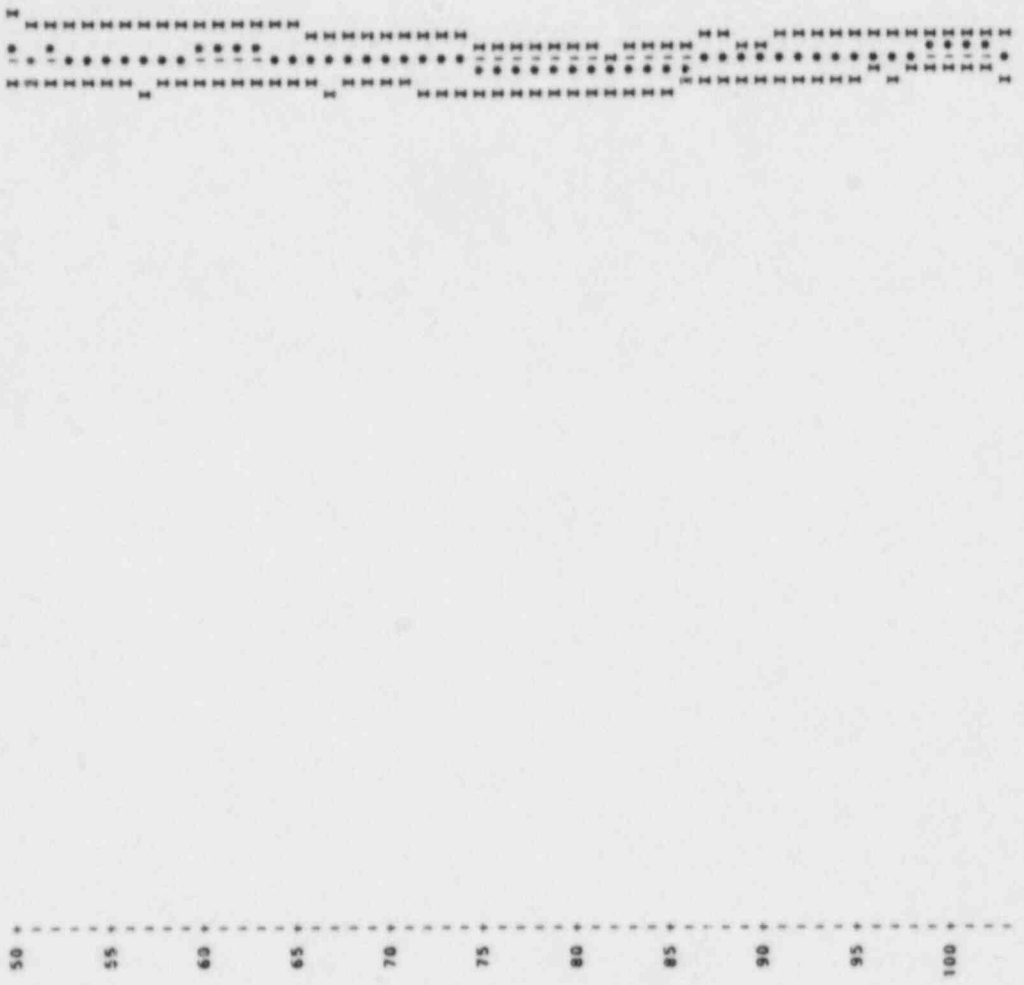
KENO MESSAGE NUMBER K5-123 EXECUTION TERMINATED DUE TO COMPLETION OF THE SPECIFIED NUMBER OF GENERATIONS.

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

LIFETIME = 7.838538E-09 + OR - 4.92332E-11 GENERATION TIME = 6.540948E-09 + OR - 4.93587E-11

NO. OF INITIAL GENERATIONS SKIPPED	AVERAGE K-EFFECTIVE	DEVIATION	67 PER CENT CONFIDENCE INTERVAL	95 PER CENT CONFIDENCE INTERVAL	99 PER CENT CONFIDENCE INTERVAL	NUMBER OF HISTORIES
3	0.99116	+ OR - 0.00462	0.98654 TO 0.99578	0.98192 TO 1.00040	0.97730 TO 1.00502	30000
4	0.99240	+ OR - 0.00450	0.98790 TO 0.99689	0.98340 TO 1.00139	0.97890 TO 1.00589	29700
5	0.99284	+ OR - 0.00452	0.98832 TO 0.99736	0.98380 TO 1.00188	0.97928 TO 1.00640	29400
6	0.99271	+ OR - 0.00457	0.98815 TO 0.99728	0.98358 TO 1.00184	0.97901 TO 1.00641	29100
7	0.99262	+ OR - 0.00461	0.98801 TO 0.99723	0.98339 TO 1.00184	0.97878 TO 1.00646	28860
8	0.99222	+ OR - 0.00464	0.98758 TO 0.99686	0.98293 TO 1.00151	0.97829 TO 1.00615	28500
9	0.99226	+ OR - 0.00469	0.98757 TO 0.99696	0.98288 TO 1.00165	0.97818 TO 1.00634	28200
10	0.99299	+ OR - 0.00469	0.98831 TO 0.99768	0.98362 TO 1.00237	0.97893 TO 1.00705	27900
11	0.99230	+ OR - 0.00469	0.98761 TO 0.99699	0.98293 TO 1.00167	0.97824 TO 1.00636	27600
12	0.99192	+ OR - 0.00472	0.98719 TO 0.99664	0.98247 TO 1.00136	0.97775 TO 1.00608	27300
17	0.99186	+ OR - 0.00484	0.98702 TO 0.99669	0.98218 TO 1.00153	0.97735 TO 1.00636	25800
22	0.99260	+ OR - 0.00497	0.98764 TO 0.99757	0.98267 TO 1.00254	0.97771 TO 1.00750	24300
27	0.99264	+ OR - 0.00509	0.98755 TO 0.99773	0.98246 TO 1.00282	0.97737 TO 1.00790	22800
32	0.99113	+ OR - 0.00527	0.98586 TO 0.99641	0.98059 TO 1.00168	0.97531 TO 1.00695	21300
37	0.99030	+ OR - 0.00534	0.98497 TO 0.99564	0.97963 TO 1.00098	0.97430 TO 1.00631	19800
42	0.98781	+ OR - 0.00542	0.98239 TO 0.99324	0.97697 TO 0.99866	0.97154 TO 1.00408	18300
47	0.98957	+ OR - 0.00571	0.98387 TO 0.99528	0.97816 TO 1.00099	0.97246 TO 1.00669	16800
52	0.98842	+ OR - 0.00609	0.98233 TO 0.99451	0.97624 TO 1.00059	0.97015 TO 1.00668	15300
57	0.99050	+ OR - 0.00664	0.98386 TO 0.99714	0.97721 TO 1.00379	0.97057 TO 1.01043	13800
62	0.98828	+ OR - 0.00713	0.98115 TO 0.99541	0.97402 TO 1.00253	0.96689 TO 1.00966	12300
67	0.99226	+ OR - 0.00756	0.98470 TO 0.99981	0.97715 TO 1.00737	0.96959 TO 1.01493	10800
72	0.99320	+ OR - 0.00816	0.98504 TO 1.00137	0.97688 TO 1.00953	0.96872 TO 1.01769	9300
77	1.00029	+ OR - 0.00883	0.99146 TO 1.00912	0.98263 TO 1.01795	0.97380 TO 1.02678	7800
82	1.00730	+ OR - 0.01001	0.99729 TO 1.01731	0.98728 TO 1.02733	0.97727 TO 1.03734	6300
87	0.99854	+ OR - 0.01229	0.98625 TO 1.01082	0.97396 TO 1.02311	0.96168 TO 1.03540	4800
92	0.99552	+ OR - 0.01682	0.97870 TO 1.01233	0.96188 TO 1.02915	0.94507 TO 1.04597	3300
97	0.99648	+ OR - 0.02485	0.97163 TO 1.02134	0.94678 TO 1.04619	0.92192 TO 1.07104	1800


SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS
PLOT OF AVERAGE K-EFFECTIVE BY GENERATION RUN. THE FINAL VALUE IS 0.9897
0.8575
0.9253
0.9932
5
10
15
20
25
30
35
40
45


50
55
60
65
70
75
80
85
90
95
100

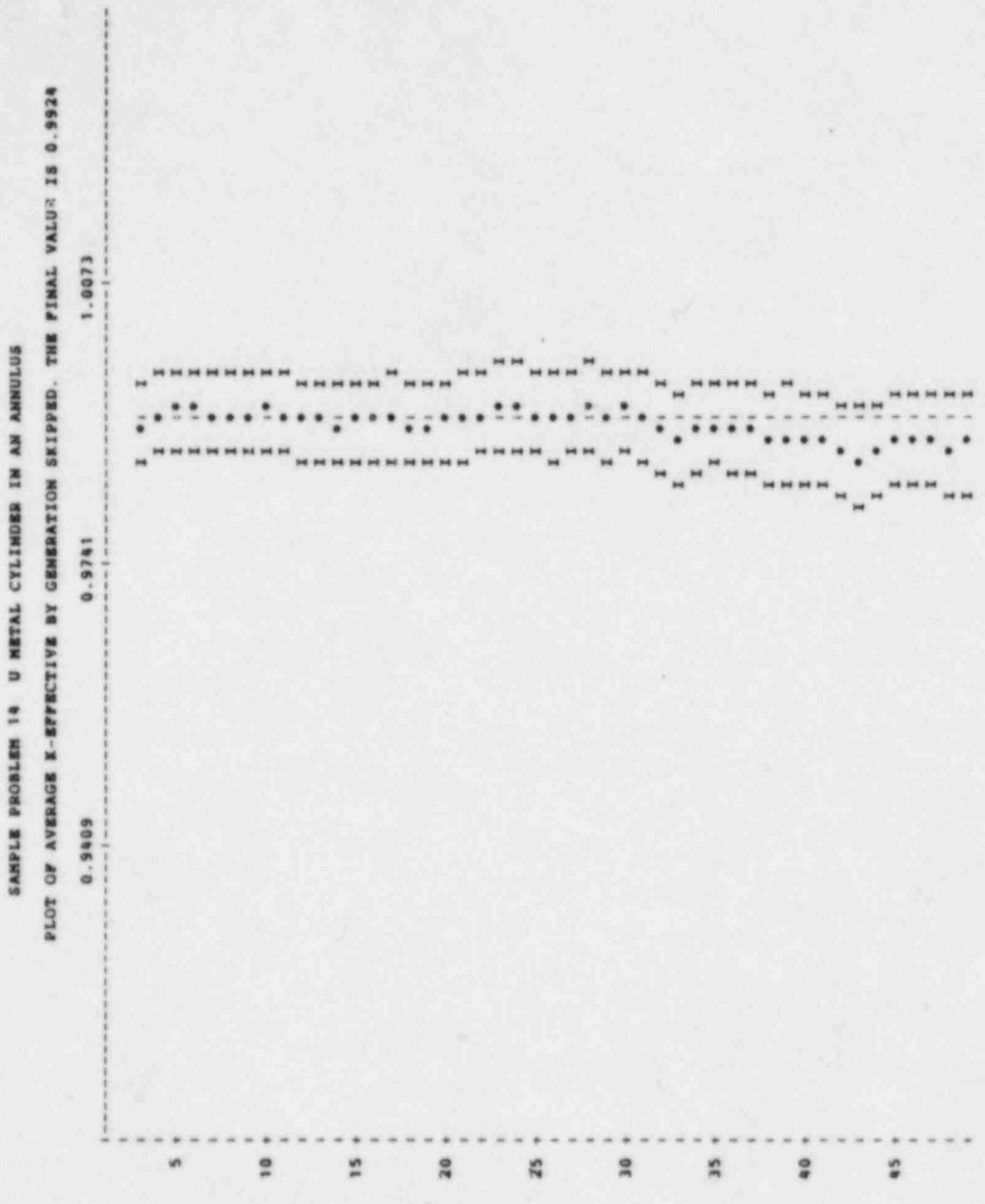

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS
PLOT OF AVERAGE K-EFFECTIVE BY GENERATION SKIPPED. THE FINAL VALUE IS 0.9924
0.9409
0.9741
1.0073
5
10
15
20
25
30
35
40
45

FIG.E.33

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

SKIPPING 7 GENERATIONS

GROUP	FISSION FRACTION	UNIT	REGION	FISSIONS	PERCENT DEVIATION	ABSORPTIONS	PERCENT DEVIATION	LEAKAGE	PERCENT DEVIATION
1	0.1424			1.41127D-01	1.5499	4.98362D-02	1.5459	8.12576D-02	1.7563
2	0.2351			2.32984D-01	1.2306	9.49076D-02	1.2306	1.44924D-01	1.3905
3	0.1393			1.38068D-01	1.4847	5.97396D-02	1.4847	9.04180D-02	1.3959
4	0.2343			2.32212D-01	1.0310	1.05608D-01	1.0310	1.45101D-01	1.0513
5	0.1987			1.96944D-01	1.3514	9.35670D-02	1.3514	9.81139D-02	1.2725
6	0.0497			4.92265D-02	3.1216	2.55777D-02	3.1216	1.19575D-02	4.3890
7	0.0006			6.05143D-04	25.5679	3.26422D-04	25.5679	8.07650D-05	42.4419
8	0.0			0.0	0.0	0.0	0.0	0.0	0.0
9	0.0			0.0	0.0	0.0	0.0	0.0	0.0
10	0.0			0.0	0.0	0.0	0.0	0.0	0.0
11	0.0			0.0	0.0	0.0	0.0	0.0	0.0
12	0.0			0.0	0.0	0.0	0.0	0.0	0.0
13	0.0			0.0	0.0	0.0	0.0	0.0	0.0
14	0.0			0.0	0.0	0.0	0.0	0.0	0.0
15	0.0			0.0	0.0	0.0	0.0	0.0	0.0
16	0.0			0.0	0.0	0.0	0.0	0.0	0.0
SYSTEM TOTAL =				9.91167D-01	0.4661	4.29562D-01	0.4785	5.71853D-01	0.3682

ELAPSED TIME 0.31183 MINUTES

RANDUM NUMBER= 075FAF8D47A1

SAMPLE PROBLEM 14 U METAL CYLINDER IN AN ANNULUS

FREQUENCY FOR GENERATIONS 4 TO 103

```
0.8601 TO 0.8832     *
0.8832 TO 0.9063     **
0.9063 TO 0.9294     ******
0.9294 TO 0.9525     **************
0.9525 TO 0.9756     ******************
0.9756 TO 0.9987     ****************
0.9987 TO 1.0218     ********************
1.0218 TO 1.0449     ******************
1.0449 TO 1.0680     *********
1.0680 TO 1.0911     ***
```

FREQUENCY FOR GENERATIONS 29 TO 103

```
0.8601 TO 0.8832
0.8832 TO 0.9063     **
0.9063 TO 0.9294     ****
0.9294 TO 0.9525     *********
0.9525 TO 0.9756     ************
0.9756 TO 0.9987     **************
0.9987 TO 1.0218     **************
1.0218 TO 1.0449     ****************
1.0449 TO 1.0680     ******
1.0680 TO 1.0911     **
```

FREQUENCY FOR GENERATIONS 54 TO 103

```
0.8601 TO 0.8832
0.8832 TO 0.9063     **
0.9063 TO 0.9294     ***
0.9294 TO 0.9525     *****
0.9525 TO 0.9756     ********
0.9756 TO 0.9987     **********
0.9987 TO 1.0218     **********
1.0218 TO 1.0449     ********
1.0449 TO 1.0680     ***
1.0680 TO 1.0911     *
```

FREQUENCY FOR GENERATIONS 79 TO 103

```
0.8601 TO 0.8832
0.8832 TO 0.9063     *
0.9063 TO 0.9294     *
0.9294 TO 0.9525     **
0.9525 TO 0.9756     ***
0.9756 TO 0.9987     ****
0.9987 TO 1.0218     *****
1.0218 TO 1.0449     ******
1.0449 TO 1.0680     **
1.0680 TO 1.0911     *
```

****

CONGRATULATIONS! YOU HAVE SUCCESSFULLY TRAVERSED THE PERILOUS PATH THROUGH KENO V IN 0.31183 MINUTES

****

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

****** NUMERIC PARAMETERS ******

TME	MAXIMUM PROBLEM TIME (MIN)	2.00
TBA	TIME PER GENERATION (MIN)	0.50
GEN	NUMBER OF GENERATIONS	103
NPG	NUMBER PER GENERATION	300
NSK	NUMBER OF GENERATIONS TO BE SKIPPED	3
BEG	BEGINNING GENERATION NUMBER	1
RES	GENERATIONS BETWEEN CHECKPOINTS	0
X1D	NUMBER OF EXTRA 1-D CROSS SECTIONS	1
NBK	NEUTRON BANK SIZE	325
XNB	EXTRA POSITIONS IN NEUTRON BANK	0
NFS	FISSION BANK SIZE	300
XFB	EXTRA POSITIONS IN FISSION BANK	0
WTA	DEFAULT VALUE OF WEIGHT AVERAGE	0.5000
WTH	WEIGHT HIGH FOR SPLITTING	3.0000
WTL	WEIGHT LOW FOR RUSSIAN ROULETTE	0.3333
RND	STARTING RANDUM NUMBER	00071AFD498D
NB8	NUMBER OF D.A. BLOCKS ON UNIT 8	200
NLS	LENGTH OF D.A. BLOCKS ON UNIT 8	512
ADJ	MODE OF CALCULATION	FORWARD
	INPUT DATA WRITTEN ON RESTART UNIT	NO
	BINARY DATA INTERFACE	NO

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

****** LOGICAL PARAMETERS ******

RUN	EXECUTE PROBLEM AFTER CHECKING DATA	YES		PLT	PLOT PICTURE MAP(S)	YES	
FLX	COMPUTE FLUX	YES		FDN	COMPUTE FISSION DENSITIES	YES	
SMU	COMPUTE AVG UNIT SELF-MULTIPLICATION	YES		NUB	COMPUTE NU-BAR & AVG FISSION GROUP	YES	
MKU	COMPUTE MATRIX K-EFF BY UNIT NUMBER	YES		MKP	COMPUTE MATRIX K-EFF BY UNIT LOCATION	YES	
CKU	COMPUTE COFACTOR K-EFF BY UNIT NUMBER	YES		CKP	COMPUTE COFACTOR K-EFF BY UNIT LOCATION	YES	
FMU	PRINT FISS PROD MATRIX BY UNIT NUMBER	YES		FMP	PRINT FISS PROD MATRIX BY UNIT LOCATION	YES	
MFH	COMPUTE MATRIX K-EFF BY HOLE NUMBER	NO		MKA	COMPUTE MATRIX K-EFF BY ARRAY NUMBER	NO	
CKH	COMPUTE COFACTOR K-EFF BY HOLE NUMBER	NO		CKA	COMPUTE COFACTOR K-EFF BY ARRAY NUMBER	NO	
FMH	PRINT FISS PROD MATRIX BY HOLE NUMBER	NO		FMA	PRINT FISS PROD MATRIX BY ARRAY NUMBER	NO	
MHL	COLLECT MATRIX BY HIGHEST HOLE LEVEL	NO		HAL	COLLECT MATRIX BY HIGHEST ARRAY LEVEL	NO	
AMX	PRINT ALL MIXED CROSS SECTIONS	NO		FAR	PRINT FIS. AND ABS. BY REGION	NO	
XS1	PRINT 1-D MIXTURE X-SECTIONS	NO		PAX	PRINT XSEC-ALBEDO CORRELATION TABLES	NO	
XS2	PRINT 2-D MIXTURE X-SECTIONS	NO		PWT	PRINT WEIGHT AVERAGE ARRAY	NO	
XAP	PRINT MIXTURE ANGLES & PROBABILITIES	NO		PGM	PRINT INPUT GEOMETRY	NO	
PKI	PRINT FISSION SPECTRUM	NO		BUG	PRINT DEBUG INFORMATION	NO	
P1D	PRINT EXTRA 1-D CROSS SECTIONS	NO		TRK	PRINT TRACKING INFORMATION	NO	

PARAMETER INPUT COMPLETED

..... 208 IO'S WERE USED READING THE PARAMETER DATA .....

********** DATA READING COMPLETED **********

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

	UNIT NUMBER	DATA SET NAME	VOLUME NAME	UNIT FUNCTION
XSC	14	SYS83314.T173403.RA000.NFCF11E.R0000013	X3B1C0	MIXED CROSS SECTIONS
ALB	79	E.LMP00000.KENO5.ALBEDOS	NE3330	INPUT ALBEDOS
WTS	80	E.LMP00000.KENO4.WGTS	NE3330	INPUT WEIGHTS
SCT	16	SYS83314.T173403.RA000.NFCF11E.R0000015	MVSDA1	WRITE SCRATCH DATA
LIB	41	E.LMP00000.AMPX.HRXSCS	NE3330	INPUT AMPX WORKING LIBRARY
	8	SYS83314.T173403.RA000.NFCF11E.R0000007	X3B1C1	INPUT DATA DIRECT ACCESS
	9	SYS83314.T173403.RA000.NFCF11E.R0000008	MVSDA1	SUPER GROUPED DIRECT ACCESS
	10	SYS83314.T173403.RA000.NFCF11E.R0000009	X3B1C1	XSEC MIXING DIRECT ACCESS

..... 31 IG'S WERE USED PREPARING INPUT DATA .....

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

MIXING TABLE

NUMBER OF SCATTERING ANGLES = 1
CROSS SECTION MESSAGE THRESHOLD =3.0E-05

ENTRY	MIXTURE	NUCLIDE	DENSITY
1	1	92660	3.22750E-03
2	1	92501	4.48020E-02
3	2	1102	5.81000E-02
4	2	7100	1.97530E-03
5	2	8100	3.69270E-02
6	2	92501	9.84710E-04
7	2	92860	7.76970E-05
8	3	6100	3.55520E-02
9	3	1102	5.68840E-02
10	3	8100	1.42210E-02

CROSS SECTIONS READ FROM THE AMPX WORKING LIBRARY ON UNIT 41

1102	HYDROGEN DE/E		HANSEN ROACH
6100	CARBON		HANSEN ROACH
7100	NITROGEN		HANSEN ROACH
8100	OXYGEN		HANSEN ROACH
92501	U-235-1R	SIG P = 20	HANSEN ROACH
92860	U-238-5R	SIG P = 200	HANSEN ROACH

..... 214 IO'S WERE USED MIXING CROSS-SECTIONS .....

1-D CROSS SECTION ARRAY ID NUMBERS
1 2002 1452 27 18 1018

..... 27 IO'S WERE USED PREPARING THE CROSS SECTIONS .....

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

****** ADDITIONAL INFORMATION ******

NUMBER OF ENERGY GROUPS	16	USE LATTICE GEOMETRY	YES
NO. OF FISSION SPECTRUM SOURCE GROUP	1	GLOBAL ARRAY NUMBER	3
NO. OF SCATTERING ANGLES IN XSECS	1	NUMBER OF UNITS IN THE GLOBAL X DIR.	2
ENTRIES/NEUTRON IN THE NEUTRON BANK	23	NUMBER OF UNITS IN THE GLOBAL Y DIR.	1
ENTRIES/NEUTRON IN THE FISSION BANK	14	NUMBER OF UNITS IN THE GLOBAL Z DIR.	1
NUMBER OF MIXTURES USED	3	USE A GLOBAL REFLECTOR	NO
NUMBER OF BIAS ID'S USED	1	USE NESTED HOLES	NO
NUMBER OF DIFFERENTIAL ALBEDOS USED	0	NUMBER OF HOLES	0
TOTAL INPUT GEOMETRY REGIONS	8	MAXIMUM HOLE NESTING LEVEL	0
NUMBER OF GEOMETRY REGIONS USED	8	USE NESTED ARRAYS	YES
NUMBER OF GEOMETRY UNITS	4	NUMBER OF ARRAYS USED	3
LARGEST ARRAY NUMBER	3	MAXIMUM ARRAY NESTING LEVEL	2
+X BOUNDARY CONDITION	VACUUM	-X BOUNDARY CONDITION	VACUUM
+Y BOUNDARY CONDITION	VACUUM	-Y BOUNDARY CONDITION	VACUUM
+Z BOUNDARY CONDITION	VACUUM	-Z BOUNDARY CONDITION	VACUUM

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

****** SPACE AND SUPERGROUP INFORMATION ******

116224 WORDS IS THE TOTAL SPACE AVAILABLE.

12871 WORDS WERE USED FOR NON-SUPERGROUP STORAGE.

103353 WORDS OF STORAGE ARE AVAILABLE FOR SUPERGROUPED DATA.

116128 WORDS OF STORAGE ARE AVAILABLE FOR CONSTRUCTING THE SUPERGROUPS.

103315 WORDS OF STORAGE ARE AVAILABLE TO EACH SUPERGROUP.

152 WORDS ARE NEEDED FOR THE LARGEST GROUP.

13151 WORDS OF STORAGE IS SUFFICIENT TO RUN THIS PROBLEM.

14451 WORDS OF STORAGE WILL ALLOW THE PROBLEM TO RUN WITH ONE SUPERGROUP.

SUPERGROUP	STARTING GROUP	ENDING GROUP	XSEC LENGTH	ALBEDO LENGTH	TOTAL LENGTH
1	1	16	203	0	1542

..... 47 IO'S WERE USED IN SUPERGROUPING .....

ARRAY NUMBER	UNITS IN X DIR.	UNITS IN Y DIR.	UNITS IN Z DIR.	NESTING LEVEL
1	1	2	2	2
2	1	2	2	2
3 GLOBAL	2	1	1	1

..... 6 IO'S WERE USED LOADING THE DATA .....

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

REGION	MEDIA NUM	BIAS ID	GEOMETRY DESCRIPTION FOR THOSE UNITS UTILIZED IN THIS PROBLEM ·											
			----- UNIT 1 -----											
URANYL NITRATE SOLUTION IN A PLEXIGLAS CONTAINER														
1 CYLINDER	2	1	RADIUS = 9.5250	+Z = 8.8900	-Z = -8.8900	CENTERLINE IS AT	X = .0	Y = .0						
2 CYLINDER	3	1	RADIUS = 10.160	+Z = 9.5250	-Z = -9.5250	CENTERLINE IS AT	X = .0	Y = .0						
3 CUBOID	0	1	+X = 10.875	-X = -10.875	+Y = 10.875	-Y = -10.875	+Z = 10.240	-Z = -10.240						
			----- UNIT 2 -----											
URANIUM METAL CYLINDER														
1 CYLINDER	1	1	RADIUS = 5.7480	+Z = 5.3825	-Z = -5.3825	CENTERLINE IS AT	X = .0	Y = .0						
2 CUBOID	0	1	+X = 6.5900	-X = -6.5900	+Y = 6.5900	-Y = -6.5900	+Z = 6.2250	-Z = -6.2250						
			----- UNIT 3 EXTERNAL TO LATTICE 1 -----											
1X2X2 ARRAY OF SOLUTION UNITS														
1 ARRAY NUMBER	1		+X = 21.750	-X = .0	+Y = 43.500	-Y = .0	+Z = 40.960	-Z = .0						
			----- UNIT 4 EXTERNAL TO LATTICE 2 -----											
1X2X2 ARRAY OF METAL UNITS PADDED TO MATCH SOLUTION ARRAY														
1 ARRAY NUMBER	2		+X = 13.180	-X = .0	+Y = 26.360	-Y = .0	+Z = 24.900	-Z = .0						
2 CUBOID	0	1	+X = 13.180	-X = .0	+Y = 34.930	-Y = -8.5700	+Z = 32.930	-Z = -8.0300						

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

------- UNIT ORIENTATION DESCRIPTION FOR ARRAY 1 -------

Z LAYER 1, X COLUMN 1 TO 1 LEFT TO RIGHT Y ROW 1 TO 2 BOTTOM TO TOP

1

1

Z LAYER 2, X COLUMN 1 TO 1 LEFT TO RIGHT Y ROW 1 TO 2 BOTTOM TO TOP

1

1

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

------- UNIT ORIENTATION DESCRIPTION FOR ARRAY 2 -------

Z LAYER 1, X COLUMN 1 TO 1 LEFT TO RIGHT Y ROW 1 TO 2 BOTTOM TO TOP

2

2

Z LAYER 2, X COLUMN 1 TO 1 LEFT TO RIGHT Y ROW 1 TO 2 BOTTOM TO TOP

2

2

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

------- UNIT ORIENTATION DESCRIPTION FOR ARRAY 3 -------

COMPOSITE ARRAY OF SOLUTION AND METAL UNITS

Z LAYER 1, X COLUMN 1 TO 2 LEFT TO RIGHT Y ROW 1 TO 1 BOTTOM TO TOP

4 3

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)
VOLUMES FOR THOSE UNITS UTILIZED IN THIS PROBLEM

UNIT	REGION	GEOMETRY REGION	VOLUME	CUMULATIVE VOLUME
1	1	1	5.06771E+03 CM**3	5.06771E+03 CM**3
	2	2	1.11006E+03 CM**3	6.17777E+03 CM**3
	3	3	3.51055E+03 CM**3	9.68832E+03 CM**3
2	1	4	1.11737E+03 CM**3	1.11737E+03 CM**3
	2	5	1.04535E+03 CM**3	2.16272E+03 CM**3
SURROUNDING GEOMETRY VOLUMES - GEOMETRY REGION 6 IS AN ARRAY PLACEMENT BOUNDARY REGION				
3	1	6	3.87533E+04 CM**3	3.87533E+04 CM**3
SURROUNDING GEOMETRY VOLUMES - GEOMETRY REGION 7 IS AN ARRAY PLACEMENT BOUNDARY REGION				
4	1	7	8.65087E+03 CM**3	8.65087E+03 CM**3
	2	8	1.48327E+04 CM**3	2.34836E+04 CM**3

UNIT	USES	REGION	TOTAL VOLUME
1	4	1	2.02708E+04 CM**3
		2	4.44025E+03 CM**3
		3	1.40422E+04 CM**3
2	4	1	4.46948E+03 CM**3
		2	4.18139E+03 CM**3
3	1	1	3.87533E+04 CM**3
4	1	1	8.65087E+03 CM**3
		2	1.48327E+04 CM**3

BIASING INFORMATION

A DEFAULT WEIGHT OF 0.500 WILL BE USED FOR ALL BIAS ID'S.

533 ID'S WERE USED IN KENO-V BEFORE TRACKING

0.01483MINUTES WERE USED PROCESSING DATA.

VOLUME FRACTION OF FISSILE MATERIAL IN THE CORE= 0.39752E+00

START TYPE 0 WAS USED.

THE NEUTRONS WERE STARTED WITH A FLAT DISTRIBUTION IN A CUBOID DEFINED BY:
+X= 3.49306E+01 -X= 0.0 +Y= 4.35000E+01 -Y= 0.0 +Z= 4.09600E+01 -Z= 0.0

0.00283 MINUTES WERE REQUIRED FOR STARTING. TOTAL ELAPSED TIME IS 0.01767 MINUTES.

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

GENERATION	GENERATION K-EFFECTIVE	ELAPSED TIME MINUTES	AVERAGE K-EFFECTIVE	AVG K-EFF DEVIATION	MATRIX K-EFFECTIVE	MATRIX K-EFF DEVIATION
KENO MESSAGE NUMBER KS-132		WARNING... ONLY	260 INDEPENDENT	FISSION POINTS WERE	GENERATED	
1	7.48506E-01	2.56667E-02	1.00000E+00	0.0	7.10490E-01	0.0
2	9.67344E-01	3.45000E-02	1.00000E+00	0.0	7.98807E-01	2.57685E-02
3	9.49693E-01	4.31667E-02	9.49693E-01	0.0	8.26246E-01	1.70463E-02
4	1.00185E+00	5.23333E-02	9.75774E-01	2.60694E-02	8.70596E-01	2.61679E-02
5	9.60620E-01	6.10000E-02	9.70722E-01	1.58795E-02	8.84585E-01	2.06513E-02
6	1.01547E+00	7.03333E-02	9.81909E-01	1.58478E-02	8.97788E-01	1.77512E-02
7	1.01776E+00	7.96666E-02	9.89078E-01	1.42158E-02	9.01571E-01	1.52573E-02
8	9.97154E-01	8.85000E-02	9.90424E-01	1.16851E-02	9.10874E-01	1.45150E-02
9	9.82732E-01	9.73333E-02	9.89325E-01	9.93755E-03	9.06992E-01	1.99374E-02
10	9.22461E-01	1.05500E-01	9.80967E-01	1.19970E-02	9.03364E-01	2.51499E-02
11	1.04861E+00	1.15000E-01	9.88483E-01	1.29783E-02	9.14596E-01	2.40443E-02
12	9.46243E-01	1.23500E-01	9.84259E-01	1.23529E-02	9.15321E-01	2.34605E-02
13	1.00663E+00	1.32167E-01	9.86293E-01	1.13571E-02	9.17519E-01	2.17669E-02
14	9.54608E-01	1.40167E-01	9.83652E-01	1.06985E-02	9.16439E-01	2.54137E-02
15	1.09190E+00	1.49500E-01	9.91979E-01	1.28912E-02	9.25459E-01	2.43149E-02
16	1.01992E+00	1.58500E-01	9.93975E-01	1.21006E-02	9.26683E-01	2.33973E-02
17	9.77112E-01	1.67500E-01	9.92850E-01	1.13211E-02	9.27447E-01	2.21536E-02
18	9.92960E-01	1.76500E-01	9.92857E-01	1.05902E-02	9.28567E-01	2.11978E-02
19	1.02483E+00	1.85333E-01	9.94738E-01	1.01243E-02	9.27698E-01	2.03106E-02
20	1.05331E+00	1.94167E-01	9.97991E-01	1.00867E-02	9.28916E-01	1.93058E-02
21	9.53435E-01	2.02500E-01	9.95646E-01	9.82479E-03	9.28190E-01	1.92364E-02
22	1.04023E+00	2.11500E-01	9.97874E-01	9.58535E-03	9.29166E-01	1.92612E-02
23	1.02995E+00	2.20167E-01	9.99401E-01	9.24452E-03	9.34984E-01	1.85224E-02
24	1.10953E+00	2.29833E-01	1.00441E+00	1.01398E-02	9.38777E-01	1.78044E-02
25	9.72884E-01	2.38833E-01	1.00304E+00	9.79001E-03	9.36725E-01	1.72313E-02
26	1.00122E+00	2.47833E-01	1.00296E+00	9.37128E-03	9.38213E-01	1.66449E-02
27	9.05264E-01	2.56000E-01	9.99052E-01	9.80002E-03	9.34962E-01	1.74313E-02
28	9.73335E-01	2.64833E-01	9.98063E-01	9.46951E-03	9.35007E-01	1.77133E-02
29	1.03461E+00	2.73500E-01	9.99416E-01	9.21100E-03	9.35515E-01	1.74942E-02
30	1.05339E+00	2.82167E-01	1.00134E+00	9.08353E-03	9.36846E-01	1.82547E-02
31	1.07911E+00	2.91167E-01	1.00402E+00	9.16840E-03	9.39607E-01	1.91021E-02
32	9.18313E-01	2.98667E-01	1.00117E+00	9.30698E-03	9.34900E-01	2.13486E-02
33	1.01663E+00	3.06667E-01	1.00167E+00	9.01581E-03	9.33261E-01	2.32877E-02
34	1.01560E+00	3.15500E-01	1.00210E+00	8.74168E-03	9.32849E-01	2.37877E-02
35	1.02189E+00	3.24167E-01	1.00270E+00	8.49244E-03	9.33352E-01	2.32786E-02
36	1.01796E+00	3.33000E-01	1.00315E+00	8.25321E-03	9.33147E-01	2.29948E-02
37	1.03878E+00	3.41833E-01	1.00417E+00	8.07902E-03	9.37236E-01	2.25200E-02
38	1.02357E+00	3.51500E-01	1.00471E+00	7.86638E-03	9.39356E-01	2.20626E-02
39	1.06509E+00	3.60833E-01	1.00634E+00	7.82436E-03	9.42175E-01	2.17480E-02
40	9.93878E-01	3.69500E-01	1.00601E+00	7.62720E-03	9.41906E-01	2.11913E-02
41	9.55863E-01	3.77667E-01	1.00472E+00	7.53783E-03	9.39797E-01	2.08525E-02
42	1.09357E+00	3.87167E-01	1.00694E+00	7.67673E-03	9.39821E-01	2.03825E-02
43	1.03060E+00	3.95833E-01	1.00752E+00	7.50747E-03	9.38642E-01	2.03221E-02
44	1.08327E+00	4.05000E-01	1.00933E+00	7.54441E-03	9.39109E-01	1.99695E-02
45	1.07735E+00	4.13833E-01	1.01051E+00	7.53585E-03	9.40434E-01	1.96925E-02
46	1.01447E+00	4.22667E-01	1.01099E+00	7.36536E-03	9.41240E-01	1.93928E-02
47	9.80039E-01	4.30833E-01	1.01030E+00	7.23190E-03	9.41546E-01	1.89906E-02
48	9.74246E-01	4.39333E-01	1.00952E+00	7.11693E-03	9.41060E-01	1.87180E-02
49	9.95819E-01	4.48167E-01	1.00922E+00	6.96959E-03	9.40592E-01	1.83540E-02
50	9.59846E-01	4.56500E-01	1.00819E+00	6.90093E-03	9.39269E-01	1.82450E-02
51	1.05040E+00	4.65333E-01	1.00906E+00	6.81410E-03	9.40749E-01	1.79555E-02
52	9.52597E-01	4.74000E-01	1.00793E+00	6.77157E-03	9.40873E-01	1.76074E-02

53	1.00783E+00	4.82667E-01	1.00793E+00	6.63775E-03	9.41339E-01	1.73163E-02
54	1.03606E+00	4.91000E-01	1.00847E+00	6.53097E-03	9.42947E-01	1.70774E-02
55	1.05756E+00	4.99500E-01	1.00939E+00	6.47070E-03	9.45062E-01	1.68111E-02
56	9.61397E-01	5.08000E-01	1.00850E+00	6.41217E-03	9.45084E-01	1.65933E-02
57	1.16868E+00	5.17833E-01	1.01142E+00	6.93724E-03	9.49793E-01	1.66978E-02
58	9.97893E-01	5.26833E-01	1.01117E+00	6.81689E-03	9.49526E-01	1.64424E-02
59	9.89907E-01	5.35500E-01	1.01080E+00	6.70641E-03	9.50130E-01	1.62125E-02
60	9.37740E-01	5.44167E-01	1.00954E+00	6.70995E-03	9.50317E-01	1.59737E-02
61	9.58961E-01	5.53000E-01	1.00868E+00	6.65062E-03	9.49187E-01	1.56900E-02
62	1.07586E+00	5.62000E-01	1.00980E+00	6.63434E-03	9.51055E-01	1.56811E-02
63	1.05048E+00	5.71167E-01	1.01047E+00	6.55947E-03	9.51997E-01	1.54558E-02
64	1.04321E+00	5.80000E-01	1.01100E+00	6.47295E-03	9.51297E-01	1.53349E-02
65	9.59632E-01	5.88333E-01	1.01018E+00	6.42174E-03	9.52061E-01	1.51096E-02
66	9.94363E-01	5.97000E-01	1.00993E+00	6.32646E-03	9.52233E-01	1.49079E-02
67	1.01438E+00	6.06333E-01	1.01000E+00	6.22918E-03	9.51889E-01	1.47077E-02
68	9.18178E-01	6.14500E-01	1.00861E+00	6.28931E-03	9.50662E-01	1.44856E-02
69	9.66086E-01	6.22833E-01	1.00798E+00	6.22874E-03	9.49142E-01	1.43812E-02
70	1.05761E+00	6.31833E-01	1.00871E+00	6.17978E-03	9.49407E-01	1.43069E-02
71	9.68329E-01	6.40333E-01	1.00812E+00	6.11585E-03	9.49108E-01	1.41704E-02
72	1.05291E+00	6.49500E-01	1.00876E+00	6.06340E-03	9.49270E-01	1.40107E-02
73	1.12490E+00	6.59333E-01	1.01040E+00	6.19504E-03	9.50188E-01	1.38356E-02
74	1.07543E+00	6.68167E-01	1.01130E+00	6.17632E-03	9.50089E-01	1.36649E-02
75	1.01520E+00	6.77333E-01	1.01135E+00	6.09211E-03	9.50483E-01	1.34942E-02
76	1.02542E+00	6.86500E-01	1.01154E+00	6.01125E-03	9.50425E-01	1.33229E-02
77	1.02755E+00	6.95333E-01	1.01176E+00	5.93694E-03	9.50232E-01	1.32070E-02
78	1.11066E+00	7.04667E-01	1.01306E+00	5.99983E-03	9.51102E-01	1.30922E-02
79	9.42550E-01	7.12667E-01	1.01214E+00	5.99068E-03	9.50660E-01	1.29296E-02
80	9.64588E-01	7.20833E-01	1.01153E+00	5.94645E-03	9.49610E-01	1.28328E-02
81	9.97242E-01	7.30000E-01	1.01135E+00	5.87292E-03	9.49107E-01	1.26689E-02
82	1.07101E+00	7.38833E-01	1.01210E+00	5.84700E-03	9.49863E-01	1.25221E-02
83	9.78012E-01	7.47000E-01	1.01168E+00	5.78979E-03	9.50118E-01	1.23869E-02
84	1.01656E+00	7.55333E-01	1.01173E+00	5.71905E-03	9.49087E-01	1.23405E-02
85	1.06210E+00	7.64333E-01	1.01234E+00	5.68304E-03	9.49962E-01	1.22166E-02
86	1.03038E+00	7.73167E-01	1.01256E+00	5.61709E-03	9.50472E-01	1.21006E-02
87	9.84869E-01	7.81500E-01	1.01223E+00	5.56216E-03	9.50148E-01	1.19849E-02
88	1.00739E+00	7.90333E-01	1.01217E+00	5.49679E-03	9.50040E-01	1.18607E-02
89	1.02863E+00	7.98833E-01	1.01236E+00	5.43625E-03	9.50828E-01	1.17468E-02
90	1.02880E+00	8.08000E-01	1.01255E+00	5.37747E-03	9.51322E-01	1.16282E-02
91	9.42541E-01	8.16667E-01	1.01176E+00	5.37412E-03	9.51197E-01	1.14991E-02
92	1.00301E+00	8.25500E-01	1.01167E+00	5.31569E-03	9.51028E-01	1.13801E-02
93	8.86025E-01	8.33667E-01	1.01029E+00	5.43629E-03	9.49032E-01	1.14230E-02
94	9.94878E-01	8.42167E-01	1.01012E+00	5.37914E-03	9.48533E-01	1.13523E-02
95	1.06066E+00	8.50667E-01	1.01066E+00	5.34788E-03	9.49063E-01	1.12660E-02
96	1.02296E+00	8.59333E-01	1.01079E+00	5.29296E-03	9.48772E-01	1.12659E-02
97	1.02992E+00	8.68000E-01	1.01099E+00	5.24156E-03	9.49175E-01	1.11923E-02
98	1.01782E+00	8.75833E-01	1.01106E+00	5.18591E-03	9.48595E-01	1.12861E-02
99	1.03606E+00	8.84000E-01	1.01132E+00	5.13856E-03	9.48522E-01	1.16470E-02
100	1.07497E+00	8.92167E-01	1.01197E+00	5.12737E-03	9.48820E-01	1.19174E-02
101	1.01804E+00	9.00167E-01	1.01203E+00	5.07629E-03	9.48897E-01	1.22514E-02
102	1.07172E+00	9.08167E-01	1.01263E+00	5.06011E-03	9.49267E-01	1.24365E-02
103	1.13332E+00	9.17000E-01	1.01382E+00	5.14992E-03	9.49641E-01	1.23898E-02

KENO MESSAGE NUMBER K5-123 EXECUTION TERMINATED DUE TO COMPLETION OF THE SPECIFIED NUMBER OF GENERATIONS.

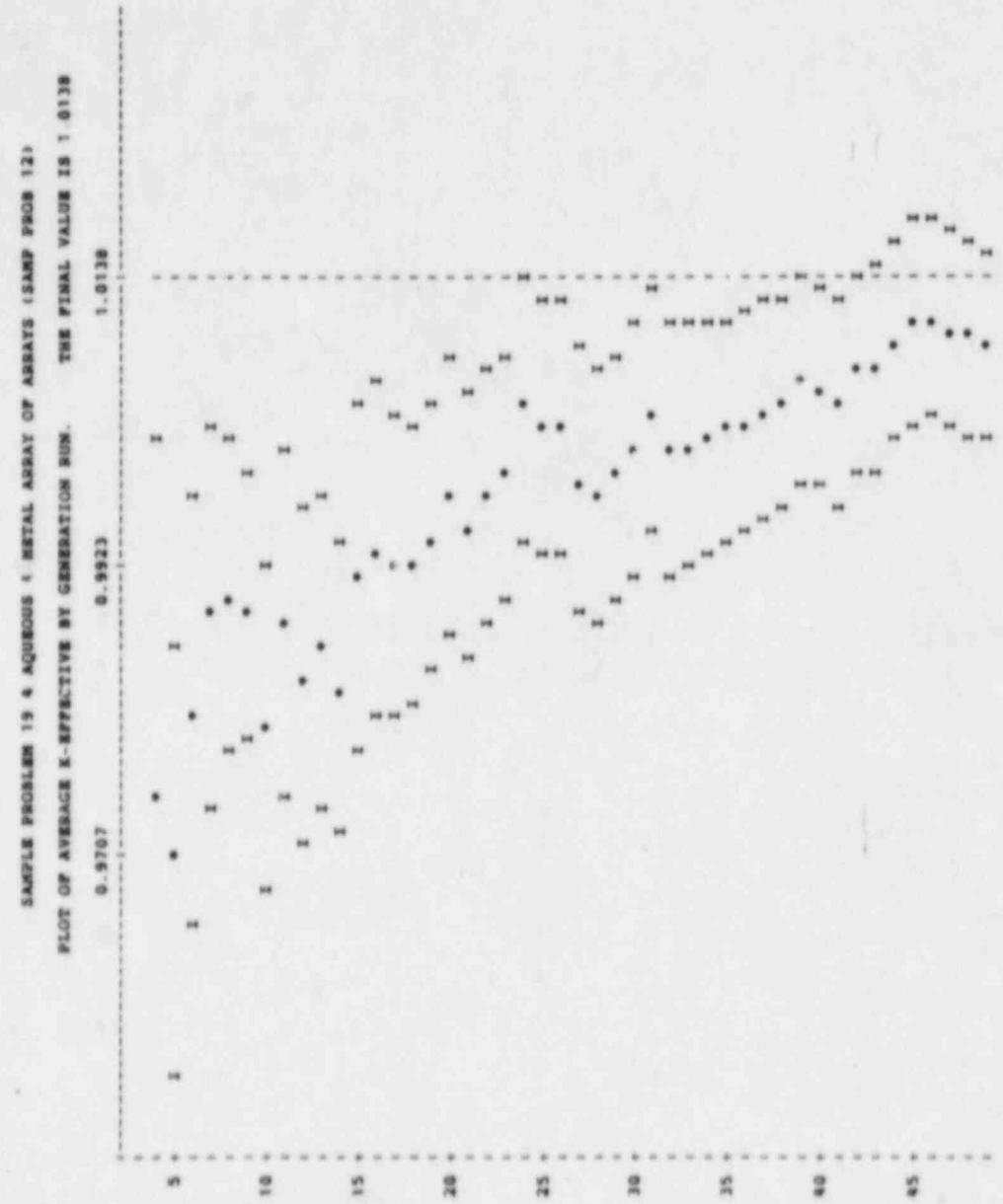
THE MATRIX K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY POSITION INDEX MATRIX.
THERE ARE NBIMAX * NBJMAX * NBIMAX POSITIONS IN AN ARRAY.

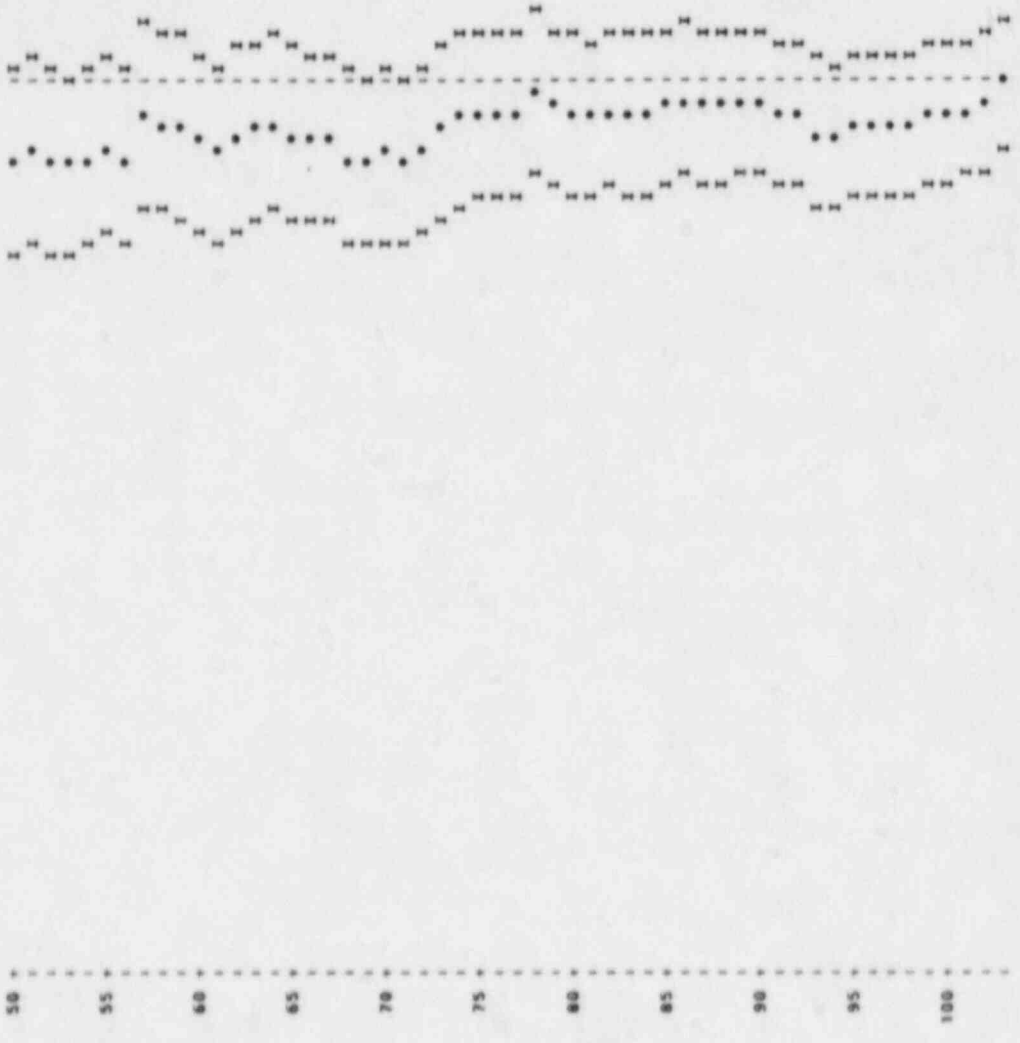
SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

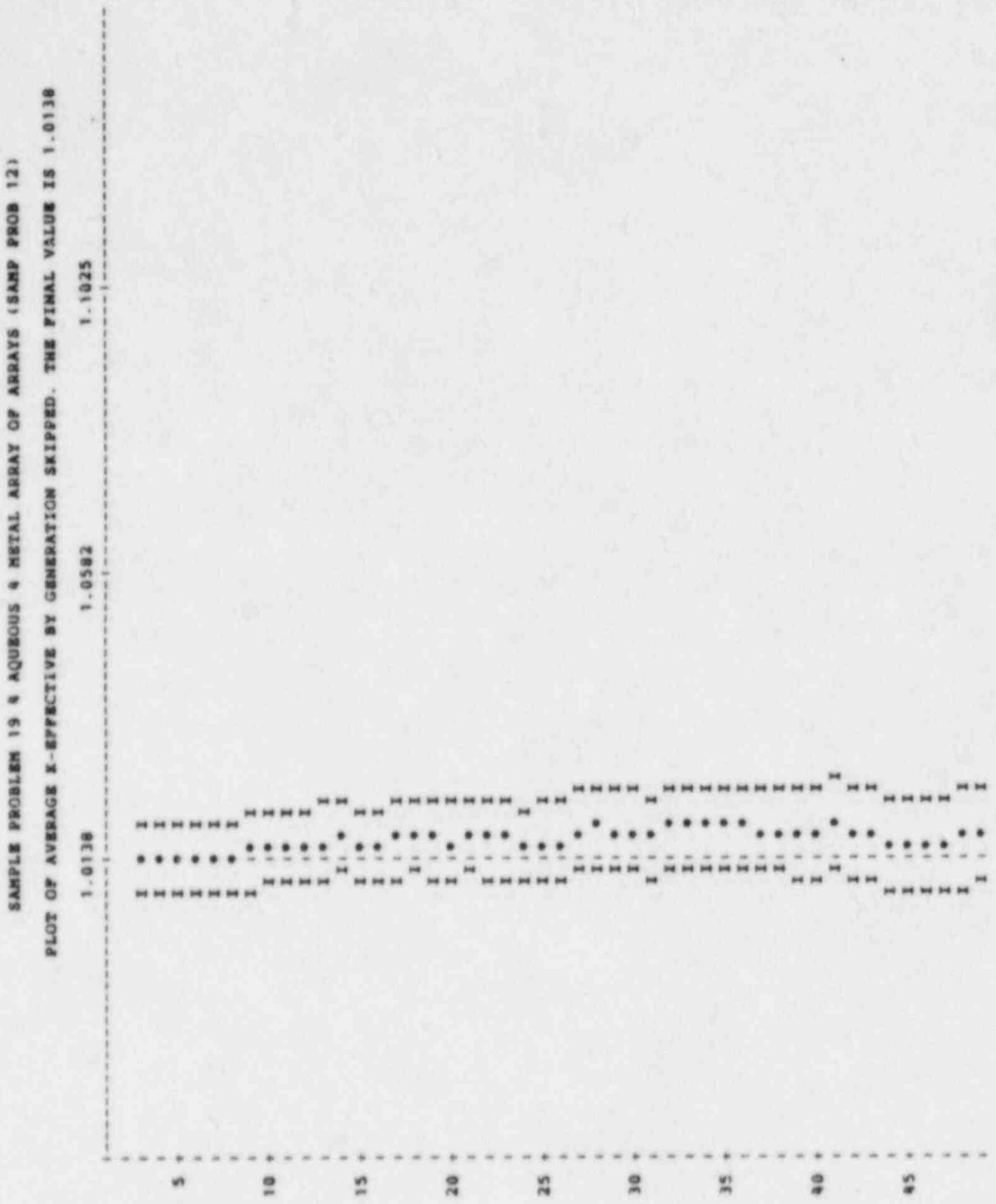
LIFETIME = 3.26469E-06 + OR - 4.28304E-08
NU BAR = 2.50549D+00 + OR - 7.94380D-04

GENERATION TIME = 3.95491E-06 + OR - 5.96115E-08
AVERAGE FISSION GROUP = 9.48951E+00 + OR - 6.27760E-02
SELF MULTIPLICATION = 6.52203E-01 + OR - 4.97591E-03

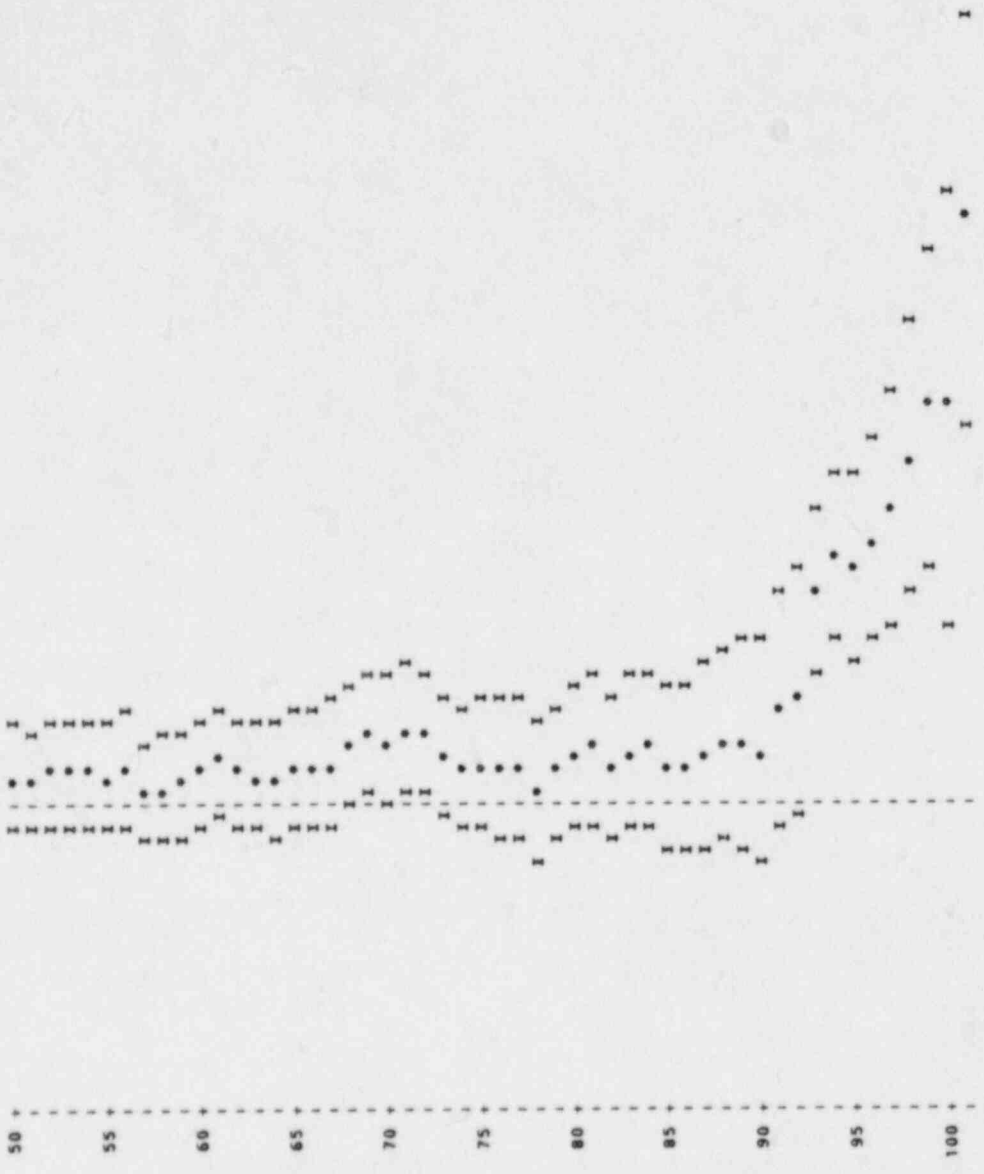
NO. OF INITIAL GENERATIONS SKIPPED	AVERAGE K-EFFECTIVE	DEVIATION	67 PER CENT CONFIDENCE INTERVAL	95 PER CENT CONFIDENCE INTERVAL	99 PER CENT CONFIDENCE INTERVAL	NUMBER OF HISTORIES
3	1.01447	+ OR - 0.00516	1.00931 TO 1.01962	1.00415 TO 1.02478	0.99900 TO 1.02993	30000
4	1.01459	+ OR - 0.00521	1.00939 TO 1.01980	1.00418 TO 1.02501	0.99897 TO 1.03021	29700
5	1.01514	+ OR - 0.00523	1.00991 TO 1.02037	1.00468 TO 1.02560	0.99945 TO 1.03083	29400
6	1.01514	+ OR - 0.00528	1.00986 TO 1.02042	1.00457 TO 1.02571	0.99929 TO 1.03099	29100
7	1.01511	+ OR - 0.00534	1.00977 TO 1.02045	1.00443 TO 1.02579	0.99909 TO 1.03113	28800
8	1.01530	+ OR - 0.00539	1.00991 TO 1.02069	1.00452 TO 1.02609	0.99912 TO 1.03148	28500
9	1.01565	+ OR - 0.00544	1.01021 TO 1.02109	1.00477 TO 1.02653	0.99933 TO 1.03197	28200
10	1.01665	+ OR - 0.00540	1.01125 TO 1.02205	1.00584 TO 1.02746	1.00044 TO 1.03286	27900
11	1.01630	+ OR - 0.00545	1.01085 TO 1.02176	1.00540 TO 1.02721	0.99995 TO 1.03266	27600
12	1.01707	+ OR - 0.00546	1.01162 TO 1.02253	1.00616 TO 1.02799	1.00070 TO 1.03344	27300
17	1.01748	+ OR - 0.00564	1.01184 TO 1.02312	1.00620 TO 1.02877	1.00056 TO 1.03441	25800
22	1.01776	+ OR - 0.00591	1.01186 TO 1.02367	1.00595 TO 1.02958	1.00004 TO 1.03548	24300
27	1.01869	+ OR - 0.00596	1.01272 TO 1.02464	1.00676 TO 1.03061	1.00080 TO 1.03657	22800
32	1.01917	+ OR - 0.00610	1.01307 TO 1.02528	1.00696 TO 1.03138	1.00086 TO 1.03749	21300
37	1.01895	+ OR - 0.00656	1.01238 TO 1.02551	1.00558 TO 1.03207	0.99926 TO 1.03864	19800
42	1.01834	+ OR - 0.00686	1.01147 TO 1.02520	1.00461 TO 1.03207	0.99774 TO 1.03893	18300
47	1.01666	+ OR - 0.00727	1.00938 TO 1.02393	1.00211 TO 1.03121	0.99484 TO 1.03848	16800
52	1.01961	+ OR - 0.00772	1.01189 TO 1.02732	1.00418 TO 1.03504	0.99646 TO 1.04276	15300
57	1.01671	+ OR - 0.00774	1.00897 TO 1.02444	1.00123 TO 1.03218	0.99349 TO 1.03992	13800
62	1.01971	+ OR - 0.00817	1.01154 TO 1.02788	1.00337 TO 1.03605	0.99520 TO 1.04422	12300
67	1.02073	+ OR - 0.00907	1.01166 TO 1.02980	1.00259 TO 1.03887	0.99352 TO 1.04794	10800
72	1.02526	+ OR - 0.00952	1.01574 TO 1.03479	1.00621 TO 1.04431	0.99669 TO 1.05383	9300
77	1.01979	+ OR - 0.01045	1.00934 TO 1.03025	0.99888 TO 1.04070	0.98843 TO 1.05116	7800
82	1.02041	+ OR - 0.01098	1.00943 TO 1.03139	0.99845 TO 1.04237	0.98747 TO 1.05335	6300
87	1.02229	+ OR - 0.01379	1.00850 TO 1.03609	0.99471 TO 1.04988	0.98092 TO 1.06367	4800
92	1.03149	+ OR - 0.01851	1.01298 TO 1.05000	0.99447 TO 1.06851	0.97595 TO 1.08702	3300
97	1.05865	+ OR - 0.01811	1.04054 TO 1.07676	1.02243 TO 1.09487	1.00432 TO 1.11298	1800


SAMPLE PROBLEM 19 4 AQUEOUS + METAL ASSAY OF ASSAYS (SAMP PROB 12)
PLOT OF AVERAGE K-EFFECTIVE BY GENERATION RUN.
THE FINAL VALUE IS 1.0138
0.9707
0.9923
1.0138
5
10
15
20
25
30
35
40
45


50
55
60
65
70
75
80
85
90
95
100


SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)
PLOT OF AVERAGE K-EFFECTIVE BY GENERATION SKIPPED. THE FINAL VALUE IS 1.0138
1.0138
1.0582
1.1025
5
10
15
20
25
30
35
40
45

F11.E.51



SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

SKIPPING 3 GENERATIONS

GROUP	FISSION FRACTION	UNIT	REGION	FISSIONS	PERCENT DEVIATION	ABSORPTIONS	PERCENT DEVIATION	LEAKAGE	PERCENT DEVIATION
1	0.0676			6.86151D-02	2.2981	2.56719D-02	2.1663	6.36150D-02	1.9596
2	0.1124			1.14000D-01	1.8606	4.67593D-02	1.8447	1.15176D-01	1.5918
3	0.0633			6.42585D-02	2.3234	2.78801D-02	2.3188	5.80234D-02	2.3102
4	0.1017			1.03125D-01	2.2047	4.70127D-02	2.2004	8.85165D-02	1.6003
5	0.0911			9.23924D-02	2.1633	4.39033D-02	2.1630	6.74845D-02	1.5323
6	0.0304			3.08045D-02	3.0207	1.60097D-02	3.0198	2.62214D-02	3.7399
7	0.0113			1.14537D-02	5.0328	6.18407D-03	5.0279	1.75605D-02	3.8829
8	0.0123			1.24335D-02	3.8519	7.23549D-03	3.8343	1.33875D-02	4.6877
9	0.0211			2.14378D-02	2.2977	1.29839D-02	2.2827	1.08164D-02	6.0877
10	0.0224			2.27142D-02	1.9902	1.44424D-02	1.9780	7.44896D-03	6.2838
11	0.0115			1.16566D-02	3.2940	7.88209D-03	3.2106	5.42980D-03	7.6158
12	0.0083			8.40679D-03	2.8830	8.09368D-03	2.7656	6.65488D-03	6.4675
13	0.0120			1.21424D-02	3.0315	6.97803D-03	2.7780	5.44152D-03	7.2349
14	0.0266			2.70122D-02	1.9841	1.33959D-02	1.9416	4.37580D-03	8.1648
15	0.1270			1.28793D-01	1.5610	6.64127D-02	1.5564	8.23836D-03	5.3454
16	0.2812			2.85227D-01	1.3132	1.43405D-01	1.3100	8.00329D-03	4.3058
SYSTEM TOTAL -				1.01447D+00	0.5082	4.94250D-01	0.5014	5.06393D-01	0.4344

ELAPSED TIME 0.91883 MINUTES

RANDUM NUMBER- 83D3766DBC01

F11.E.52

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

****

POSITION K-EFFECTIVE= 9.49641E-01 + OR - 1.23898E-02
THE POSITION K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY POSITION INDEX MATRIX.

****

ELAPSED TIME 0.91883 MINUTES

FISSION PRODUCTION BY POSITION INDEX MATRIX
(I, J) P IS THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED AT POSITION INDEX J BY A NEUTRON BORN AT POSITION INDEX I.

(1, 1) 0.0 (1, 2) 1.08E+00

(2, 1) 0.0 (2, 2) 9.50E-01

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

SOURCE VECTOR BY POSITION INDEX

INDEX	VECTOR
1	0.0
2	1.00000E+00

AVERAGE SELF MULTIPLICATION BY ARRAY POSITION

THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN A UNIT LOCATED AT A GIVEN POSITION IN THE ARRAY BY
A NEUTRON BORN IN THAT SAME UNIT IS 9.49641E-01 + OR - 1.23898E-02

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

POSITION INDEX	POSITION X	Y	Z	UNIT	COFACTOR K-EFFECTIVE	DEVIATION
1	1	1	1	4	9.49641E-01	1.23898E-02
2	2	1	1	3	0.0	1.23898E-02

ELAPSED TIME 0.91883 MINUTES

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

****

UNIT K-EFFECTIVE= 1.01224E+00 + OR - 9.86082E-03
THE UNIT K-EFFECTIVE IS THE LARGEST EIGENVALUE OF THE FISSION PRODUCTION BY UNIT NUMBER MATRIX.

****

ELAPSED TIME 0.91883 MINUTES

FISSION PRODUCTION BY UNIT NUMBER MATRIX
(I, J) P IS THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN UNIT J BY A NEUTRON BORN IN UNIT I.

(1, 1) 8.43E-01	(1, 2) 1.07E-01	(1, 3) 0.0	(1, 4) 0.0
(2, 1) 1.77E-01	(2, 2) 9.01E-01	(2, 3) 0.0	(2, 4) 0.0
(3, 1) 0.0	(3, 2) 0.0	(3, 3) 0.0	(3, 4) 0.0
(4, 1) 0.0	(4, 2) 0.0	(4, 3) 0.0	(4, 4) 0.0

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

SOURCE VECTOR BY UNIT

UNIT	VECTOR
1	5.11256E-01
2	4.88743E-01
3	0.0
4	0.0

AVERAGE SELF MULTIPLICATION BY UNIT

THE NUMBER OF NEXT GENERATION NEUTRONS PRODUCED IN A UNIT BY
A NEUTRON BORN IN THAT SAME UNIT IS 8.71285E-01 + OR - 5.39189E-03

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

UNIT	COFACTOR K-EFFECTIVE	DEVIATION
1	4.40266E-01	1.49430E-02
2	4.31019E-01	1.16566E-02
3	1.01246E+00	9.86556E-03
4	1.01241E+00	9.86435E-03

ELAPSED TIME 0.91883 MINUTES

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

**** FISSION DENSITIES ****

UNIT	REGION	FISSION DENSITY	PERCENT DEVIATION	TOTAL FISSIONS
1	1	2.584E-05	1.10	5.237E-01
	2	0.0	0.0	0.0
	3	0.0	0.0	0.0
2	1	1.098E-04	1.30	4.907E-01
	2	0.0	0.0	0.0
3	1	0.0	0.0	0.0
4	1	0.0	0.0	0.0
	2	0.0	0.0	0.0

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

FLUXES FOR UNIT 1

GROUP	REGION 1 FLUX	REGION 1 PERCENT DEVIATION	REGION 2 FLUX	REGION 2 PERCENT DEVIATION	REGION 3 FLUX	REGION 3 PERCENT DEVIATION
1	3.378E-05	2.10	2.285E-05	2.68	2.085E-05	2.97
2	6.565E-05	1.57	4.300E-05	1.96	3.862E-05	2.12
3	3.045E-05	1.90	2.017E-05	2.71	1.870E-05	3.29
4	4.431E-05	1.62	3.166E-05	2.11	2.967E-05	2.47
5	4.356E-05	1.55	2.982E-05	1.93	2.510E-05	2.65
6	3.108E-05	1.39	1.896E-05	2.72	1.285E-05	3.83
7	2.305E-05	1.58	1.446E-05	2.79	9.598E-06	4.53
8	2.018E-05	1.53	1.166E-05	2.63	7.014E-06	4.57
9	1.930E-05	1.62	1.037E-05	3.14	6.652E-06	4.92
10	1.199E-05	1.67	7.388E-06	4.22	4.528E-06	5.27
11	1.051E-05	1.96	5.829E-06	3.67	3.605E-06	7.93
12	1.114E-05	1.72	6.272E-06	3.49	3.754E-06	6.83
13	9.508E-06	1.79	5.570E-06	3.74	3.207E-06	6.20
14	7.307E-06	2.02	4.532E-06	4.39	2.701E-06	7.70
15	1.379E-05	1.55	8.332E-06	4.41	5.195E-06	5.38
16	1.119E-05	1.39	1.293E-05	3.33	6.839E-06	4.22

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

FLUXES FOR UNIT 2

GROUP	REGION 1 FLUX	REGION 1 PERCENT DEVIATION	REGION 2 FLUX	REGION 2 PERCENT DEVIATION
1	9.0388E-05	2.31	5.0778E-05	2.97
2	1.6668E-04	1.97	9.0918E-05	2.43
3	9.9338E-05	2.24	5.5248E-05	2.65
4	1.6588E-04	2.07	8.6758E-05	2.22
5	1.2578E-04	2.20	6.2278E-05	2.87
6	2.2198E-05	3.50	1.5238E-05	5.18
7	2.0878E-06	10.41	6.7768E-06	10.23
8	1.0548E-06	11.40	4.6018E-06	10.46
9	4.0188E-07	14.97	3.4958E-06	11.08
10	1.4458E-07	17.38	1.9328E-06	15.76
11	3.6338E-07	15.22	2.1888E-06	12.88
12	2.9108E-07	17.50	2.0298E-06	14.61
13	2.6018E-07	17.55	1.9398E-06	14.16
14	6.0478E-08	30.29	1.0608E-06	17.11
15	3.3998E-08	14.57	2.6118E-06	10.70
16	2.0228E-08	12.70	2.7598E-06	11.30

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

FLUXES FOR UNIT 3

GROUP	REGION 1 FLUX	REGION 1 PERCENT DEVIATION
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0
10	0.0	0.0
11	0.0	0.0
12	0.0	0.0
13	0.0	0.0
14	0.0	0.0
15	0.0	0.0
16	0.0	0.0

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

FLUXES FOR UNIT 4

GROUP	REGION 1 FLUX	REGION 1 PERCENT DEVIATION	REGION 2 FLUX	REGION 2 PERCENT DEVIATION
1	0.0	0.0	1.5898E-05	3.75
2	0.0	0.0	3.0048E-05	2.76
3	0.0	0.0	1.6678E-05	3.90
4	0.0	0.0	2.5328E-05	2.53
5	0.0	0.0	1.8338E-05	3.14
6	0.0	0.0	6.4938E-06	5.94
7	0.0	0.0	3.8418E-06	8.00
8	0.0	0.0	2.5478E-06	8.86
9	0.0	0.0	2.3748E-06	11.50
10	0.0	0.0	1.6708E-06	12.33
11	0.0	0.0	1.2788E-06	13.67
12	0.0	0.0	1.4618E-06	13.02
13	0.0	0.0	1.0568E-06	14.20
14	0.0	0.0	9.4478E-07	16.16
15	0.0	0.0	1.7198E-06	11.60
16	0.0	0.0	1.5778E-06	10.18

SAMPLE PROBLEM 19 4 AQUEOUS 4 METAL ARRAY OF ARRAYS (SAMP PROB 12)

FREQUENCY FOR GENERATIONS 4 TO 103

```
0.8831 TO 0.9062     **
0.9062 TO 0.9293     ***
0.9293 TO 0.9524     ****
0.9524 TO 0.9755     ****************
0.9755 TO 0.9986     ***************
0.9986 TO 1.0217     ******************
1.0217 TO 1.0448     ******************
1.0448 TO 1.0679     ***********
1.0679 TO 1.0910     ********
1.0910 TO 1.1141     ****
1.1141 TO 1.1372     **
1.1372 TO 1.1602
1.1602 TO 1.1833     *
```

FREQUENCY FOR GENERATIONS 29 TO 103

```
0.8831 TO 0.9062     *
0.9062 TO 0.9293     **
0.9293 TO 0.9524     ***
0.9524 TO 0.9755     **********
0.9755 TO 0.9986     **********
0.9986 TO 1.0217     ************
1.0217 TO 1.0448     ****************
1.0448 TO 1.0679     *********
1.0679 TO 1.0910     ********
1.0910 TO 1.1141     **
1.1141 TO 1.1372     **
1.1372 TO 1.1602
1.1602 TO 1.1833     *
```

FREQUENCY FOR GENERATIONS 54 TO 103

```
0.8831 TO 0.9062     *
0.9062 TO 0.9293     *
0.9293 TO 0.9524     ***
0.9524 TO 0.9755     ******
0.9755 TO 0.9986     *******
0.9986 TO 1.0217     *******
1.0217 TO 1.0448     **********
1.0448 TO 1.0679     ******
1.0679 TO 1.0910     *****
1.0910 TO 1.1141     *
1.1141 TO 1.1372     **
1.1372 TO 1.1602
1.1602 TO 1.1833     *
```

FREQUENCY FOR GENERATIONS 79 TO 103

```
0.8831 TO 0.9062     *
0.9062 TO 0.9293
0.9293 TO 0.9524     **
0.9524 TO 0.9755     *
0.9755 TO 0.9986     ****
0.9986 TO 1.0217     *****
1.0217 TO 1.0448     ******
1.0448 TO 1.0679     **
1.0679 TO 1.0910     ***
1.0910 TO 1.1141
1.1141 TO 1.1372     *
1.1372 TO 1.1602
1.1602 TO 1.1833
```

****

CONGRATULATIONS! YOU HAVE SUCCESSFULLY TRAVERSED THE PERILOUS PATH THROUGH KENO V IN 0.91950 MINUTES

****

F11.E.2 MICROFICHE OF COMPLETE SAMPLE PROBLEM OUTPUT

Microfiche of the computer output from all the sample problems is located at the end of the Appendices. The microfiche is labeled Sect. F11.E.2 and has the corresponding title.

APPENDIX F

F11.F MICROFICHE CROSS REFERENCE OF VARIABLES USED IN KENO V

The cross-reference listing of the variables used in KENO V is given on microfiche located on the following page. The microfiche is labeled F11.F and has the corresponding title.

NUREG/CR-0200
Volume 2, Section F12
ORNL/NUREG/CSD-2/V2/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

JUNEBUG-II — A THREE-DIMENSIONAL GEOMETRY PLOTTING CODE

M. B. Emmett
L. M. Petrie
J. T. West

Manuscript Completed: August 1984

Date Published: December 1984

Prepared for the
Office of Nuclear Regulatory Research
and the
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
Under Interagency Agreement
DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

CONTENTS

LIST OF FIGURES F12.v

LIST OF TABLES F12.vii

ACKNOWLEDGMENTS F12.ix

ABSTRACT F12.xi

F12.1 INTRODUCTION F12.1.1

F12.2 GENERAL DESCRIPTION OF GEOMETRY PLOTTING CONCEPTS F12.2.1

F12.3 INPUT DESCRIPTION F12.3.1

F12.4 PROGRAM FLOW F12.4.1

F12.5 SUBROUTINE DESCRIPTION F12.5.1

F12.6 SAMPLE PROBLEMS F12.6.1

LIST OF FIGURES

F12.4.1 JUNEBUG-II Program Flow F12.4.2

F12.6.1 JUNEBUG-II Drawing of "4 Aqueous 4 Metal" Sample Problem F12.6.2

F12.6.2 Sample Problem Computer Input and Output F12.6.3

LIST OF TABLES

F12.1.1 Definitions of Variables in COMMON TMBE F12.1.2

F12.1.2 Definitions of Variables in COMMON VEWDIM F12.1.2

F12.1.3 Definitions of Variables in COMMON RESOL F12.1.3

F12.1.4 Definitions of Variables in COMMON LINE F12.1.3

F12.3.1 Debug Print Messages From JUNEBUG-II F12.3.4

ACKNOWLEDGMENTS

During the development of JUNEBUG, the developers of the DISSPLA Graphics Software at Integrated Software Systems Corporation in San Diego, California, were very cooperative in discussing their 3-D graphics system. Their assistance is gratefully acknowledged and appreciated.

ABSTRACT

The JUNEBUG-II code is an exceptionally valuable tool which is used to verify and validate three-dimensional (3-D) combinatorial geometry and KENO geometry models. It uses the MARS geometry system for modeling complicated multiple array lattices as well as simple combinatorial geometry models. Input for JUNEBUG-II is very flexible and will accept standard KENO-IV input data, as well as combinatorial MARS geometry input.

JUNEBUG-II is an improved version of JUNEBUG which has been updated to provide faster running times and more efficient storage allocation. Variable I/O units are now used, a new and more efficient COMMON structure has been developed, and program errors detected by users have been eliminated.

F12.1 INTRODUCTION

JUNEBUG-II is a tool used to verify and validate three-dimensional combinatorial geometry and KENO geometry models. It uses the MARS geometry system for modeling complicated multiple array lattices as well as simple combinatorial geometry models. Input for JUNEBUG-II is very flexible. It will accept standard KENO-IV input data (see Sect. F5), as well as combinatorial MARS geometry input (see Sect. M9). JUNEBUG-II is based on JUNEBUG, but it has a new COMMON structure, variable I/O units, and some error corrections which improve running time and storage allocation.

The main features in JUNEBUG-II are:

1. the code accepts geometry input for MORSE-SGC/S (MARS) and KENO-IV codes;
2. the user may specify or default the bodies to draw and the bodies to check for intersections;
3. the user determines by geometry zone or material media regions of space which are transparent, nontransparent, or invisible;
4. the user specifies the subspace to plot;
5. the user determines the viewpoint;
6. the user determines the arrays to plot or skip;
7. in geometry models containing smaller arrays nested inside of larger arrays, the user specifies the depth of nesting to be plotted;
8. multiple plots from different viewpoints with different transparency specifications are available in a single run;
9. plot resolution is adjustable for "quick" plots; therefore, computer time requirements are user-controlled;
10. JUNEBUG-II uses the DISSPLA Graphics System for device independent plot capabilities. This allows the plot output to be diverted to any device the user selects.

This document is designed to introduce users to the concepts involved in 3-D geometry plotting. Section F12.2 introduces and discusses the following topics:

1. the criteria for drawing a line;
2. the concept of plot resolution, the amount of detail drawn, and the resolution involved;
3. the concept of hidden lines and how it is applied in JUNEBUG-II;
4. the procedure JUNEBUG-II utilizes to determine and draw combinatorial body intersections;
5. the combinatorial bodies available in JUNEBUG-II.

Section F12.2 is intended as a broad overview of the code to aid the user in obtaining good plots with a reasonable amount of computer time. Section F12.3 contains the JUNEBUG-II Input Description. Sample problems for JUNEBUG-II are given in Sect. M9.A on the MARS Geometry System. Tables F12.1.1 through F 12.1.4 define variables in labelled commons.

Table F12.1.1 Definitions of Variables in COMMON TMBE

Variable Name	Definition
KI	Pointer to body flag input array
KV	Pointer to zone transparency array
IERG	Error flag - If =0, exit on error. If =1, continue.
ILG	Signal to indicate whether this is a new view. If =0, initiate new view; =1, otherwise.
ISPR	Debug print level - controls amount of output printed for various errors. If = -1, ZEXITS will be called to print the combinatorial geometry zone transfer table.
ISRP	Boundary test - If =0, universe; if=1, array.
KMLU	Pointer to Master LookUp array.
KMDM	Pointer to Master Data Management array.
NMLU	Size of MLU array.
NXR	Block number at which data transfer is to start on direct access read or write
NAS	Associated variable for direct access I/O Number of record physically following the one just processed.

Table F12.1.2 Definitions of Variables in COMMON VEWDIM

Variable Name	Definition
SID(6)	Plot subspace in absolute universe or level 0 space - global. SID(1) is XMIN; SID(2) is XMAX: SID(3) is YMIN; SID(4) is YMAX: SID(5) is ZMIN; SID(6) is ZMAX.
SED(6)	Plot subspace in other universes - local coordinate.
VX(3)	Viewpoint coordinates - global.

Table F12.1.3 Definitions of Variables in COMMON RESOL

Variable Name	Definition
MI	Minimum increment for straight line resolution.
MX	Maximum increment for straight line resolution.
MLI	Minimum increment for curved line resolution.
MLX	Maximum increment for curved line resolution.
RES	Segment size factor for straight lines
ROS	Segment size factor for curved lines
IPL	Indicates how coordinates of points on a circle are stored. If = 0, x, y, z is stored in 3 separate arrays. If = 1, x, y, z is stored in one array of dimension 3*number of points.

Table F12.1.4 Definitions of Variables in COMMON LINE

Variable Name	Definition
U	X-direction cosine of vector from one point to another in a view
V	Y-direction cosine of vector from one point to another in a view
W	Z-direction cosine of vector from one point to another in a view
DST	Distance between two points on plotting line.

F12.2 GENERAL DESCRIPTION OF GEOMETRY PLOTTING CONCEPTS

All lines drawn by JUNEBUG-II must meet the following conditions:

1. the points on the line must be on an input zone boundary;
2. the points on the line are associated with a given body; the body must be referenced in at least one adjacent zone description;
3. the points must be visible from the viewpoint; all material zones between the viewpoint and the points drawn must be transparent or invisible;
4. the points must be inside the subspace specified to be plotted;
5. space around the points must be well defined or the job will, depending on user option, either abort or leave the plot space vacant.

If a plot is missing a line the user believes should be present, the user should make certain the line meets all the above requirements. Code zone boundaries inside the same input zone are not plotted, since they do not constitute a possible media boundary.

Before determining if a line is to be drawn, a method must be defined to describe the line. This introduces the problem of plot resolution.

In more simple terms, plot resolution is the number of points used to define a given curve or line in space. The line between two points determines an interval on the curve. The size of the interval determines the number of intervals defining the line and the detail of the line plotted. The more intervals in a line, the more points to process in the program, hence requiring more computer time.

The plot resolution determines the amount of detail plotted on body intersections. If an intersection is not completely drawn, the plot resolution should be increased. The user must decide if the additional detail desired is worth the additional computer time required.

The resolution used in a plot may be different between straight lines and curved lines. This feature gives the user some flexibility in optimizing his run time. The user inputs a minimum number of intervals for a line, a maximum number of intervals for a line, and a recommended interval length. The minimum and maximum number of intervals for a line take priority over the recommended interval length.

The total length of the line is divided by the recommended interval length to determine the number of intervals required to draw the line. If this number is less than the minimum number of intervals, the minimum number of intervals is drawn. If this number is greater than the maximum number of intervals, then the maximum number of intervals is drawn. This criteria is input for both straight lines and for curved lines on the first card of the program input. The resolution size is used to determine the resolution for the lines defining all body boundaries. If the user is plotting a lattice array, the increased resolution may be very costly. Computer time for JUNEBUG-II is directly proportional to the line resolution required.

After lines for a model have been defined, a selection of the lines to draw must take place. A plot with too many lines will look cluttered. A plot with too few lines will lack detail and possibly depth. To add depth to a figure and reduce the number of lines drawn on a plot, geometry zones are defined as being transparent, nontransparent, or invisible. The user should understand the hidden line logic in JUNEBUG-II so as to successfully utilize this feature with his models.

A line is made up of points defining intervals on a line. Rays are tracked from the viewpoint to each point on a line. Both points defining an interval must be visible from the viewpoint in order for the interval to be drawn. If a line disappears on a plot before it is actually hidden, then the line resolution is too coarse. The user must decide if the extra detail is worth the additional computational cost.

Lines are hidden when they are not visible from the viewpoint due to blockage by a nontransparent media zone. Lines may be selectively removed from plots in the following ways:

1. the body containing the line is not drawn;
2. if the line is an intersection of two bodies and the intersection flag is turned off for one of the two bodies, then the intersection is not computed;
3. if the line borders a zone which is flagged INVISIBLE, then the line is not drawn; lines adjacent to or inside an invisible zone are not drawn;
4. the points on the line are outside the plot subspace.

Many geometry models contain body intersections. The intersections are necessary to describe the particular shape being modeled. As such, they should appear in the geometry plot since they contain useful information on the model shapes.

JUNEBUG-II does not attempt to analytically compute the intersections of the various combinatorial bodies. This process would be tedious and complicated. The approach taken is a very simple technique which lacks mathematical complexity. After points have been generated for each of the defining borders (lines) of the bodies, then arbitrary lines are drawn connecting points at opposite sides of the bodies. These lines are parallel, in the case of cylinders, but not for planar surfaces. These lines are individually scanned and checked for intersections with each of the possible bodies. This may at first appear to be inefficient, but this technique is quick and the logic is straightforward.

The time required to generate body intersections may be controlled by use of the plot resolution parameters and by selectively bypassing the body intersection scan for specific bodies. The plot resolution parameters determine the resolution of the scan performed on each body. The body intersect array in the JUNEBUG-II input allows the user to bypass the scanning of selective bodies. In most combinatorial geometry models, the number of intersecting bodies is limited. If the geometry is a simple universe* or a KENO geometry model, then intersections may not exist, and it is not necessary to check for them. Subroutine SCAN performs the body scans and passes the intersecting points to SIGHT. An intersection table is composed giving the sides of each intersecting body. For example, the table may determine side 2 of body 4 intersects side 3 of body 5. Subroutine XCHEK generates the table and is called to verify that an intersection has not been previously generated. Some sides of some bodies may intersect twice, most only once. It is not possible for a side of one body to intersect with the same side of another body more than twice. The actual sides intersecting and the intersecting points generated may be printed out if the user desires the information. This is usually a very large amount of output.

JUNEBUG-II draws geometry from simple geometric enclosed volumes, called combinatorial bodies. The body descriptions are given in Sect. M9.2. JUNEBUG-II does not have all of the combinatorial bodies implemented. The bodies JUNEBUG-II presently draws are RPP, BOX, WED, SPH, ARB, and RCC. Models containing body types not implemented may be drawn but will be missing the nonimplemented bodies. The missing body types are still available for geometry modeling and ray tracing.

---

*See Sect. M9.3.1 for a description of the term universe. KENO users may consider a universe as combinatorial geometry's equivalent to KENO BOX TYPES.

F12.3 INPUT DESCRIPTION

JUNEBUG may be run "stand-alone" or as part of SCALE. When run in SCALE, the user's first input card should be **=JUNEBUG** beginning in column 1 and his last input card should be **END** beginning in column 1.

JUNEBUG-II CARD INPUT (All input is free form)

CARD A (11 Entries Required)

MI - Minimum number of intervals for straight lines ≥ 4

MX - Maximum number of intervals for straight lines ≥ MI

MLI - Minimum number of intervals for curved lines ≥ 4

MLX - Maximum number of intervals for curved lines ≥ MLI

IP(20) - Transparency input will be entered as:

= 0 by combinatorial or KENO input zone
= 1 by combinatorial code zone (not applicable for KENO INPUT)
= 2 by combinatorial or KENO media

ISRP - Boundary test will be performed as:

= 0 across input zone boundaries in the same universe
= 1 across input zone boundaries in different universes; cell and array boundaries will be tested.
NOTE: ISRP=0 is normally sufficient; ISRP=1 is a more rigorous test of the geometry.

IERG - Error Flag

= 0 Exit on error (recommended)
= 1 Continue plotting leaving undefined space vacant
NOTE: IERG=1 may abort if user's plot is poorly defined.

JMK - Geometry Type

= 0 COMJOM MARS INPUT
= 1 KENO-IV INPUT

ISPR - Debug Print Level†

= -1 Prints zone transfer table which can be very large.
= 0 to 5 (HIGHLY recommended ISPR = 0)! The larger ISPR, the more debug print the program gives the user. WARNING: ISPR=5 gives much output.

---

†See Table F12.3.1 for additional information.

RES - Straight Line Recommended Interval Size (cm)

ROS - Curved Line Recommended Interval Size (cm)

GEOMETRY INPUT (See Sects. M9.A and M9.B)

CARDS B-1 - If geometry is combinatorial MARS, input CARDS A.1 through A.7 should be entered (Appendix F12.A)

CARDS B-2 - If the geometry is KENO-IV, the KENO input should be as follows:

---Title Card
---KENO Region Cards (Sect. M9.B)
---END the KENO Input with DONE

CARDS B-3 - The array input, CARDS A.8 through A.10 should be entered (Appendix F12.A). A.10 is not needed for KENO-IV input.

PLOTTING INPUT

CARD C (two arrays of input)

Body Draw/Skip Array (one entry for each combinatorial body or KENO Region)

Body Intersect Array (one entry for each combinatorial body or KENO Region)

ENTER -1 and both arrays will be defaulted. The default draws all bodies with the exception of universe reference bodies (KENO CUBOID BOX TYPES - last card) and array reference bodies (KENO CORE BOUNDARY REGION).

If the arrays are entered the input is:

For the Body Draw/Skip Array -
0 - SKIP the body
1 - DRAW the body

For the Body Intersect Array -
0 - NO SCAN for intersections
1 - SCAN for intersections

CARD D (1 entry)

NVEW - Number of plots to draw

CARD E - TITLE FOR PLOT: terminate the title with "$"; less than 40 characters.

CARD F - PLOT PAGE SIZE (2 entries) XP, YP (in.) Recommend 10.0 by 7.5 in. for Tektronix users and 13 by 10.49 in. for CALCOMP and GOULD plot users.

CARD G ...PLOT SUBSPACE... (6 entries)
...XMIN, XMAX, YMIN, YMAX, ZMIN, and ZMAX

...The coordinates of these entries should be in absolute universe or level 0 space for combinatorial MARS geometry.

...For KENO-IV geometry the coordinates should be relative to the dimensions external to the CORE BDY card. If the problem has an array but no CORE BDY card, then assume a coordinate system whose origin is 0,0,0 at the lower left-hand corner of the lattice array.

CARD H ...PLOT VIEWPOINT... (3 entries)
...VX(1), VX(2), VX(3)

...Same coordinates as PLOT SUBSPACE BOUNDARY (see CARD G above)

...Must be a large distance from the plot volume subspace to prevent plot distortion. It cannot be inside of the plot volume.

...The viewpoint must be inside of defined space, unless the geometry input is strictly KENO-IV.

EXAMPLE: 8500.0 6500.0 5000.0

CARD I ...ZONE TRANSPARENCY ARRAY INPUT...

IVS array

(number of entries depends on IP(20) entry on CARD A)

ENTER 0 - TRANSPARENT ZONE VISIBLE BOUNDARIES

ENTER 1 - NONTRANSPARENT ZONE VISIBLE BOUNDARIES

ENTER 2 - INVISIBLE BOUNDARIES TRANSPARENT ZONE
(Any lines bordering an INVISIBLE ZONE will not be drawn)

CARD J ...(Entries required is number of arrays + 1)
IAR array

0-1 ...ARRAY CONTROL (1 entry per array)
...0 - DRAW the array
...1 - SKIP the array

0-2 ...Depth of geometric nesting to draw...
...(i.e., 0, 1, 2, or 3 level of geometry to draw)

CARD K (2 entries) (Required if NVEW ≠ 1)

IPS - For next plot do you wish to reread CARD E, F, and G
0 - NO
1 - YES
II - For next plot do you wish to reread CARD I and J
0 - NO
1 - YES

For each additional plot enter CARD K, optionally CARDS E, F, and G depending on IPS, CARD H, and optionally CARDS I and J depending on II. This allows the user to generate consecutive plots changing transparency data, viewpoint, plot space, title card, arrays plotted, and depth of nesting plotted. Data not reread remains as set by the previous plot parameters.

Table F12.3.1. Debug Print Messages From JUNEBUG-II

Value of ISPR	Subroutine Printing Message	Message
>= 2	ART	DATA GENERATED FOR BODY NO. ___IN UNIVERSE ___,POSITION _____ OF ARRAY NO. ___ IN LEVEL NO. ___
>= 2	ART	DATA GENERATED FOR BODY NO. ___ IN LEVEL ___
>= 1	DRAW	Calls XREF to edit MLU and MDM arrays.
>= 2	PLANE	DATA GENERATED FOR BODY ___ SIDE ___ ___ ___ FROM ___ TO ___ .
= -1	DRIVER	Calls ZEXITS to print the zone transfer summary tables.
>= 2	RCS	DATA GENERATED FOR BODY # ___ SURFACE # ___
>= 2	SCAN	BODY ___ SIDE ___ INTERSECTS WITH BODY ___ SIDE ___
>= 2	SIGHT	Calls TABE to print table of adjacent zone pairs.
>= 2	SIGHT (only if IST=3)	Prints variables L,XQ(L), YP(L),ZP(L).
>= 5	ZRAY	IN ZRAY * MM,IST,IZBOD,IXBOD, LL,KLOOP,LOOP,ISRP,ISPR ___ ___ ___ ___ ___ ___ ___ ___ ___
>= 5	ZRAY	IN ZRAY THE 3-D PLOTTING ARRAY COORDINATES ARE ___ ___ ___ ___
>= 2	ZRAY	Calls TABE to print table of adjacent zone pairs.

F12.4 PROGRAM FLOW

JUNEBUG-II's logic flow is shown in Fig. F12.4.1. This section is intended to convey to the user the broad logic flow of JUNEBUG-II without a detailed description of the array structure utilized in data storage or an in-depth discussion of individual subroutine flow. The following discussion attempts to explain the program flow given in Fig. F12.4.1.

MAIN is the entry point for program execution. It calls MESAGE to print out the banner page on the output. JUNEBUG-II reads the first card of data, and then calls ALOCAT, which determines the amount of computer memory available for data storage. It passes the first word of this space and the length of the space to DRIVER. This subroutine acts as the controlling driver of the program for the remainder of the execution. DRIVER first calls the combinatorial input routine JOMIN which calls JOMIN1, JOMIN2, and AZIP to set up the geometry data storage. If the problem input is KENO-IV format, JOMICK is called before JOMIN. The JOMICK routine translates the KENO input into combinatorial geometry and writes a binary data file which is read by JOMIN2 when JOMIN is called. DSET is called from DRIVER to read the body data arrays. These arrays determine whether a body is to be drawn or not, and whether intersections with the body are to be checked. These arrays may be defaulted, in which case DSET calls ODBOD. This completes the input processing and array setup.

The following discussion is performed for each universe until all universes have been processed.

DRIVER calls SETUP which calls BOOK. This constructs the direct access arrays for checking for body intersections. A table is composed for all possible bodies which may intersect a given body in a given universe. This list is used later in SCAN for checking for body intersections. DRIVER then calls FLOW which calls DRAW. DRAW calls the body subroutines which generate the points for the bodies in the universe being drawn. Each of the body routines for the planar bodies, ARB, BOX, WEDGE, RPP call Subroutine PLANE. Spheres and RCC's are generated by RCS. RCS and PLANE generate the points for the body lines and call SCAN for intersection checks. SCAN, RCS, and PLANE call SIGHT to check and store the points for the individual bodies in each universe for final processing by ART and ZRAY. If boundaries are not tested across universe borders (variable ISRP is zero) then SIGHT calls G1 and verifies each point passed is on an input zone border and is part of a body referenced in an adjacent zone description. DRIVER repeats the calls to SETUP and FLOW until points for all bodies in all universes have been generated. SIGHT saves the processed points on a sequential access file. This completes the universe processing. Next comes the wrap-up of the universe data and the actual plot generation.

DRIVER calls MOW which assimilates all the body data generated for each universe and places this data on a direct access scratch file for use by ART. It constructs directory tables for later use by ART. These tables describe the contents of each direct access record grouped by universe and body in the universe.

DISSPLA graph initialization subroutines are then called in DRIVER. GRAFIT, DSET, and ART are then called for each viewpoint. GRAFIT reads the graph title card and the viewpoint. DSET reads the transparency array for each plot. ART calls ZRAY and actually constructs the plot from the geometry input and the lattice array descriptions. ZRAY determines if each point on a line is visible from the viewpoint and if the point is located on a zone boundary. This is similar to the process used in SIGHT. In ZRAY if ISRP is 1, then array boundaries are checked between universes. In SIGHT, when ISRP is zero, input boundaries are checked inside a single universe. ZRAY calls CURVND which calls CURV3D to process the points before passing them to DISSPLA to be drawn. CURV3D takes the 3-D points and generates a 2-D plot. This is the "guts" of the plotting operation. The criteria in ZRAY for plotting a point is the same as in SIGHT. The points must meet the criteria for drawing a line as described earlier in Sect. F12.2. They must be on an input zone boundary and must be part of a body referenced in an adjacent code zone description. ZRAY checks across universe boundaries, as contrasted to SIGHT which checks across input zone boundaries inside the same universe.

ORNL Dwg. 81-5657R

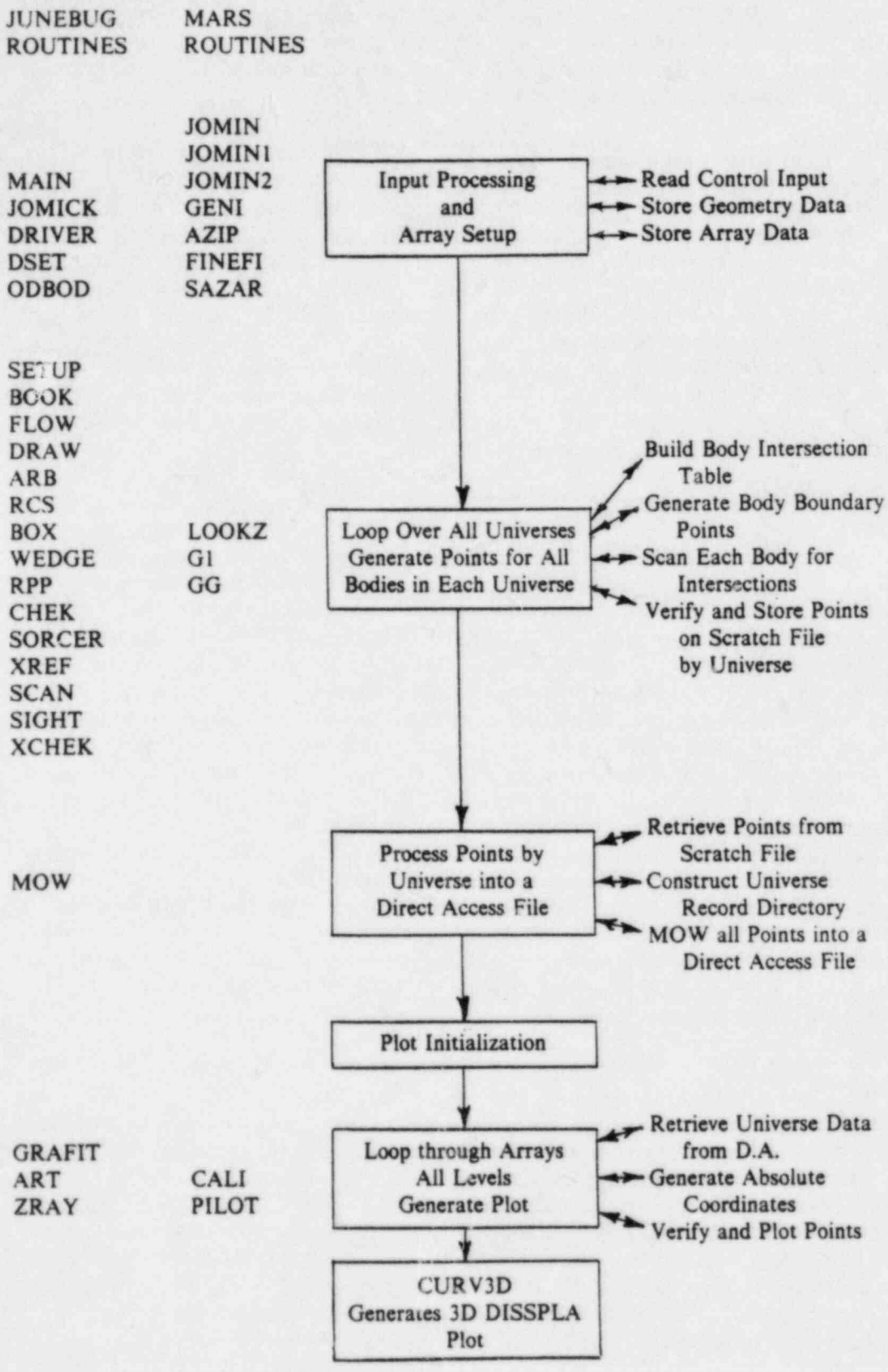


Fig. F12.4.1. JUNEBUG-II Program Flow

F12.5 SUBROUTINE DESCRIPTION

The following is a brief description of each of the JUNEBUG-II subroutines. The MARS subroutines are described in Sect. M9.D.

ARB

ARB is called from DRAW. It generates the points for the arbitrary polyhedron by calling PLANE for each side of the body.

ART

ART loops over the geometry description and generates the 3-D plot by calling ZRAY. It is called from DRIVER and obtains the plot points from direct access files sorted by universe. The points passed to ZRAY are absolute coordinates. ART performs coordinate translation utilizing a nesting table by calling CLEV.

BOOK

BOOK is called from SETUP. It generates a table of possible bodies which may intersect a given body in a particular universe. The table generated is the MLU array or Master Look-Up array used in SCAN for checking for body intersections. The MLU array points to the MDM array, the Master Data Management array containing the actual side and body table of intersections.

BOUND

BOUND is called from SIGHT and ZRAY to determine if the points to be drawn are part of a body used in an adjacent zone description. All points generated and passed throughout JUNEBUG-II are associated with a given body number. Points for body outlines have only one body associated with them, while points generated by body intersections have two bodies associated with them.

BOX

BOX is called from DRAW to generate points for BOX bodies. It calls PLANE for each side of a BOX. A BOX is a body type which contains six orthogonal sides arbitrarily positioned in space.

CHEK

CHEK is called from DRAW. It sets flags in the MDM array not to check for intersections on bodies already scanned and determined not to intersect.

CIRCLE

CIRCLE is called from RCS and generates points on a circle parallel to the X-Y coordinate plane. The points are later rotated to their true position in space. Circular points are needed for cylinders and spheres.

CROSS

CROSS is used extensively for calculating the cross product of two vectors. It has several entry points to compute vector addition (VADD), subtraction (VDIF), and the dot product of two vectors (DOT).

CURVND

CURVND sets up arrays to be plotted and calls the DISSPLA routine CURV3D for plotting these three-dimensional data.

DRAW

DRAW is called from FLOW. It controls the looping over all bodies referenced in a given universe as prescribed in the MLU array. It calls ARB, RCS, BOX, WEDGE, RPP, CHEK, SORCER, and XREF.

DRIVER

DRIVER is the controlling routine in JUNEBUG-II. It controls the entire program flow logic. It calls JOMIN which calls JOMIN1, JOMIN2, and AZIP or JOMICK to read in the geometry data. It calls SETUP and FLOW to generate points for each universe. It calls MOW to process the points by universe for later retrieval by ART. It calls COMPRS, and BGNPL for DISSPLA plot initialization. It calls GRAFIT and ART for each plot to be generated. The entire program flow is organized in this subroutine.

DSET

DSET is called in two places by DRIVER. The subroutine is in two parts. The first part reads in the body plot and body intersection input array or calls ODBOD to default the input. The second part reads in the ARRAY plot or skip array and the depth of nesting array.

ELLIPS

ELLIPS generates points for an elliptical circle. At present the ellipse body type is not implemented; if it were, ELLIPS would be called by RCS. This subroutine is not called by any routine in this version of JUNEBUG-II.

FLOW

FLOW is called from DRIVER. It calls DRAW. This routine mainly sets array pointers for calling DRAW.

GRAFIT

GRAFIT is called by DRIVER. It reads the dimensions for the plot subspace, XMIN, XMAX, YMIN, YMAX, ZMIN, and ZMAX and it reads the viewpoint coordinates, VX(1), VX(2), and VX(3). It calls DISSPLA routines PAGE, TITL3D, AXES3D, VUABS, and GRAF3D.

INLIST

INLIST may be called to give a printout of the card image input. Not currently called. If JUNEBUG is run under SCALE, the data are already printed out.

JOMICK

This routine translates KENO-IV data to combinatorial geometry data and writes the resulting data on a scratch file in the format required by JOMIN2.

LINBRK

LINBRK is a utility routine called by many subroutines in JUNEBUG-II. Given the coordinates for two points, it generates a set of coordinates for points at equal intervals between the two end points. Furthermore, it computes the distance between the two points and the direction cosines of the vector from the first point to the second point. It calls PLACE to verify both points are within the plot subspace.

MAIN

This is the entry point for execution. The first card of input with the control parameters is read and printed out by this routine. The routine calls ALOCAT, which in turn calls DRIVER.

MOW

MOW is called from DRIVER. It reads the sequential file containing point data for each body in each universe and writes this data onto a direct access file for use by ART. It constructs a directory for ART showing which records contain the data for a given universe.

MPACK

MPACK is a utility routine which packs an array of integers into a single integer word.

ODBOD

ODBOD is called by DSET to default the body draw/skip array and the body intersect array. Body not drawn by default are reference bodies for universes and reference bodies for arrays.

PLACE

PLACE is called from SIGHT. It determines if a point is inside the reference body of a universe. If the point is external to an array, it is discarded. This is important for properly drawing universes modeled with bodies which extend beyond the universe reference body.

PLANE

PLANE controls the drawing and scanning of planar surfaces. It is called by RPP, WEDGE, ARB, and BOX. It calls LINBRK, PSET, SCAN, SIGHT, and UNPACK.

PSET

PSET is a utility routine called by many routines in JUNEBUG-II. It takes coordinates in three arrays and constructs a single array of coordinates or it can perform the reverse operation.

RCS

RCS is called by DRAW for body types RCC, SPH, REC, TRC, and ELL. Only the RCC and SPH are implemented in the present version of JUNEBUG-II.

RISK

RISK is called from ART to translate coordinates defined on geometry level NL1 to geometry level NL2 using the nesting table stored in array LP. This translation is from a global coordinate to a local coordinate system. NL2 must be greater than or equal to NL1. Because RISK is also used in other codes that use the MARS geometry, it is packaged with MARS.

ROTATE

ROTATE is called from RCS to perform a coordinate rotation and translation given a rotational matrix and a translation coordinate. The routine can translate a coordinate from a nonrotated coordinate system into a rotated coordinate system or perform the reverse operation.

RPP

RPP is called from DRAW to generate the points for the RPP body type. It calls PLANE for each side of the RPP.

SCAN

SCAN is called from PLANE and RCS to compute intersections between bodies. The intersection points are passed to SIGHT for ultimate disposal. SCAN calls the combinatorial routine GG to compute body intersections.

SETUP

SETUP is called by DRIVER. It calls BOOK to set up the possible body intersection table for each body referenced in a given universe. It constructs the MLU table and computes the maximum length of the MDM array.

SIGHT

SIGHT is called from PLANE, RCS, and SCAN. It processes points generated from body data and from intersection data. It will optionally test points for belonging to a valid line. It calls LOOKZ and G1. The points are written out on a sequential file for retrieval by MOW.

TABE

TABE is called from SIGHT and ZRAY. It determines if points related to a given body are valid zone boundary points. It compares the body descriptions of the adjacent zones to determine if the body is referenced. If the body is not referenced, the points are eliminated.

TANT

TANT computes the tangents on a cylinder from an arbitrary viewpoint. It makes extensive use of subroutine CROSS and its entries.

UNPACK

UNPACK is a utility routine, which unpacks a packed integer.

WEDGE

WEDGE is called from DRAW to generate points for WEDGE body types. It calls PLANE for each side of the wedge.

XCHEK

XCHEK is called from SCAN to record body side intersection correspondence and to retrieve the information to prevent recomputing intersections previously computed.

XREF

XREF is called from DRAW to optionally print out the MLU and MDM arrays. These arrays contain the final intersection table generated by XCHEK during body scanning.

ZRAY

ZRAY is called from ART. It processes data points similar to subroutine SIGHT. ZRAY works in absolute global coordinants and calls CURVND which makes the only call to the DISSPLA subroutine CURV3D for plotting the three-dimensional combinatorial MARS geometry.

F12.6 SAMPLE PROBLEMS

Six examples of JUNEBUG-II input using Combinatorial MARS Input are presented in Sect. M9 of the SCALE document. These examples illustrate isometric drawings of simple combinatorial geometry and more complicated MARS array geometry.

This section contains an illustration of a KENO-IV Model examined with JUNEBUG-II. Because of the relative simplicity and limitations of KENO-IV geometry, it requires the least amount of additional input for a good drawing. The model presented in Fig. F12.6.1 is the "4 Aqueous and 4 Metal" Problem given in Sample Problem 19 of the KENO-IV Manual (ORNL-4938 or Sect. F5). The sample problem input and output are given in Fig. F12.6.2. The JUNEBUG-II input is described below:

Card No.	Description
1	JUNEBUG-II Control Parameters
2	KENO-IV Title Card
3-19	KENO-IV Mixed Box Region Cards
20	Array Size Description
21-22	KENO Mixed Box Orientation Cards
23	Default Body/Draw/Scan Input
24	Generate 1 Drawing
25	Plot Title Card
26	Plot Size
27	Dimensions Plotted
28	Plot Viewpoint
29	Transparency Input
30	Array Draw Control

The geometry input for the array size specification and array content is given in the MARS Input Description, Sects. M9.A.8 and M9.A.9. This sample problem ran approximately 33% faster with JUNEBUG-II than it did with JUNEBUG. The core storage required was also slightly less. In complicated geometries with many bodies and zones the number of I/O requests may be up to 20% less with JUNEBUG-II; and since the size of one of the large arrays has been reduced by approximately a factor of 7, the core storage reduction is significant.

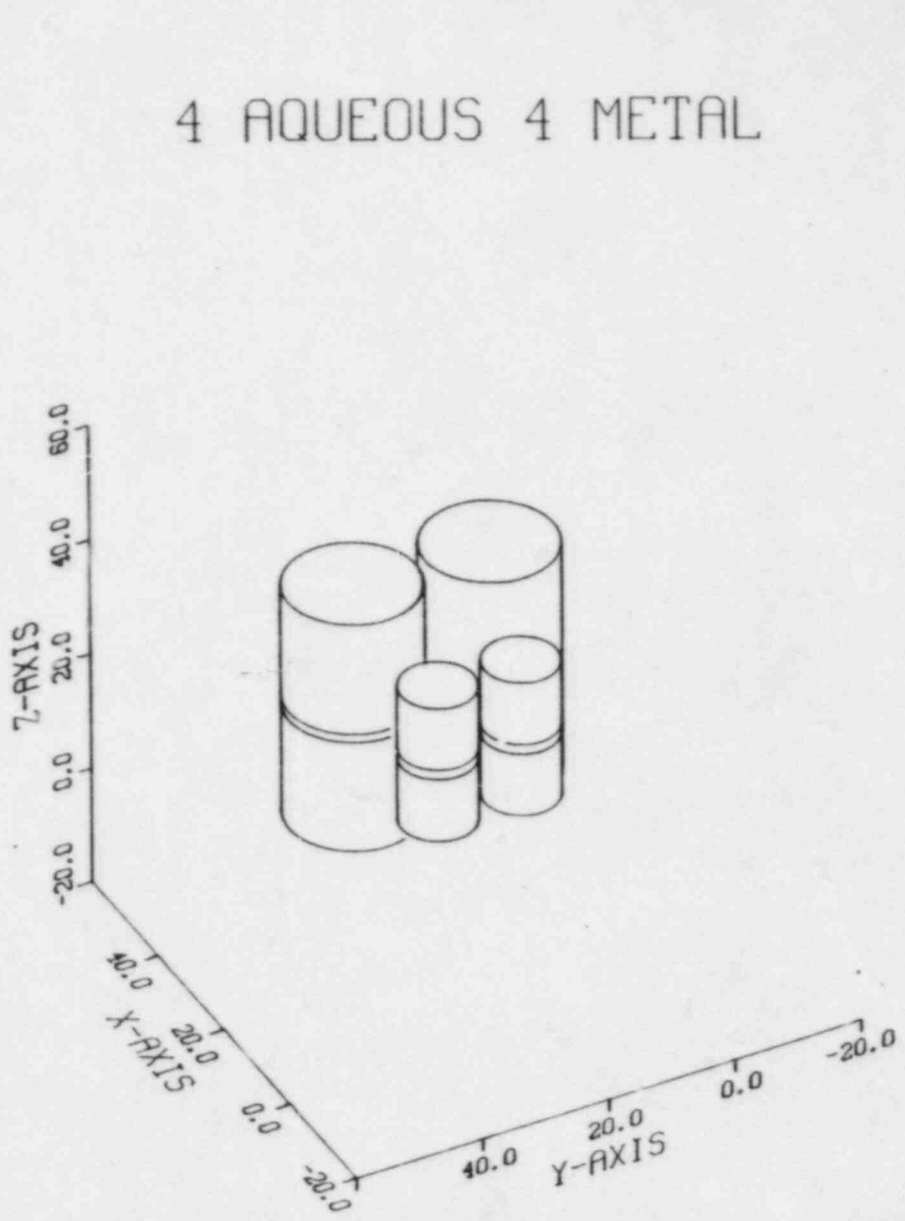
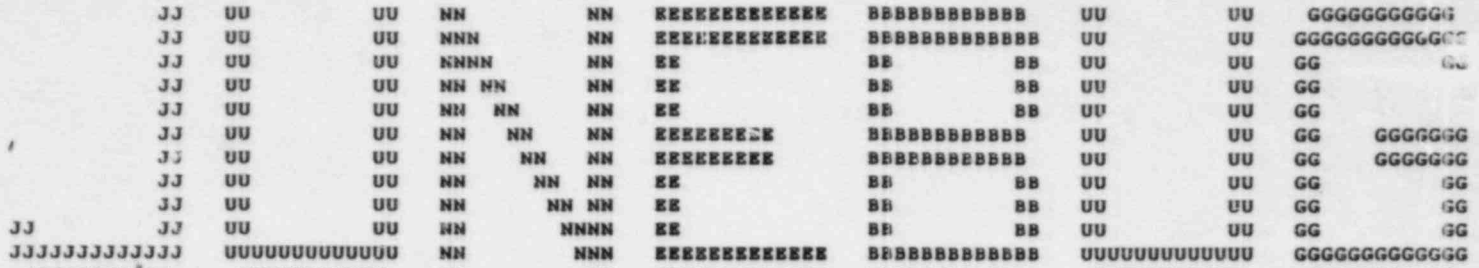
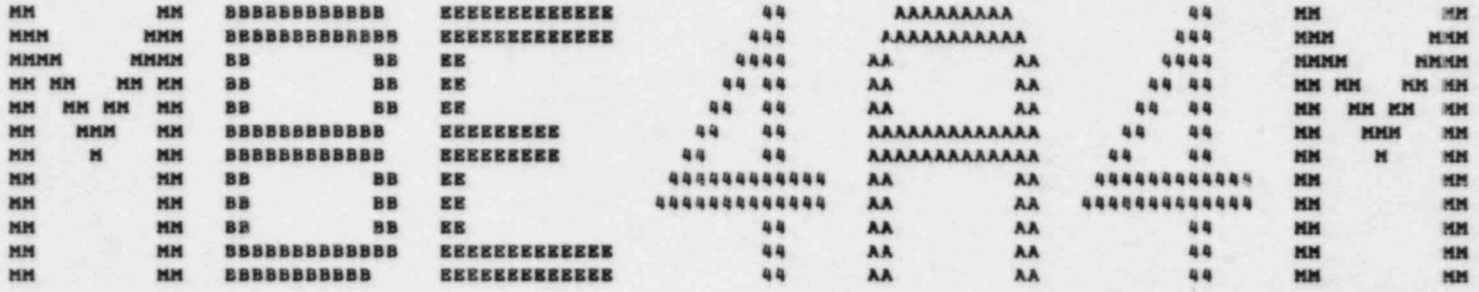


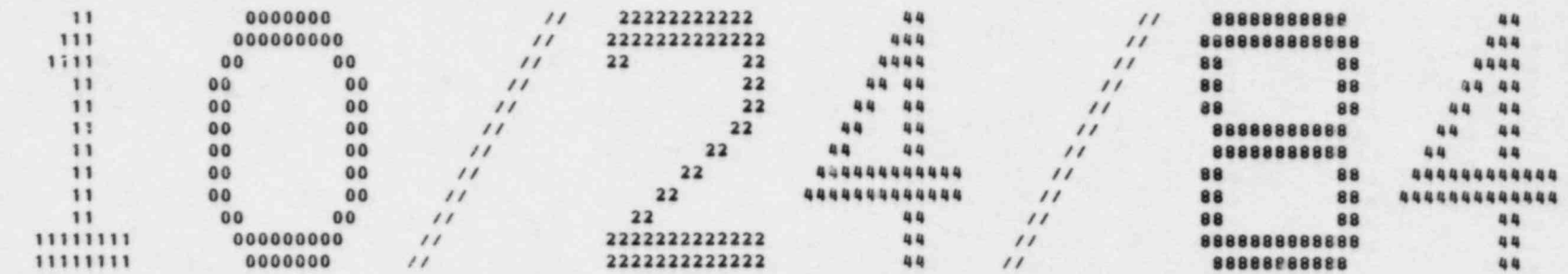
Fig. F12.6.1 JUNEBUG Drawing of "4 Aqueous 4 Metal" Sample Problem.

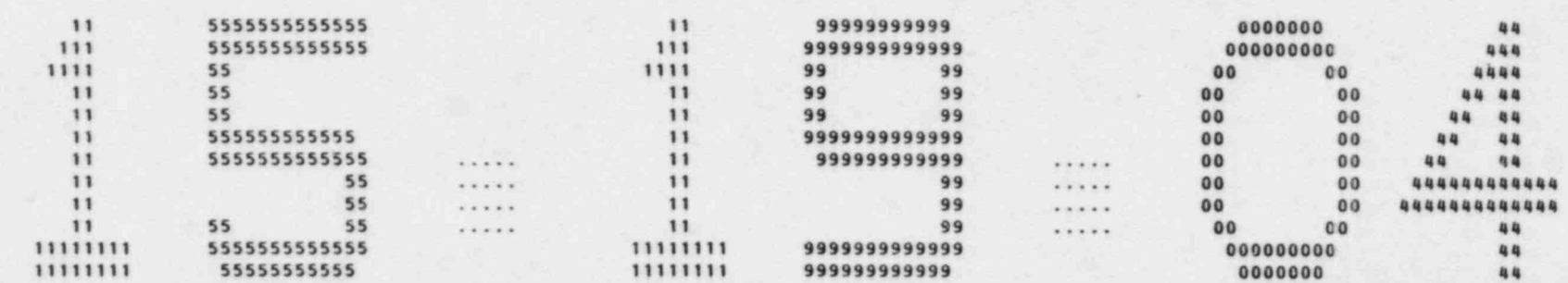
```
=JUNEBUG
 5  50  5  50 2 0 0 1 -1 0.5 0.2
4 AQUEOUS 4 METAL
BOX TYPE  1
CYLINDER  2 9.525 8.89 -8.89 16*0.5
CYLINDER  3 10.16 9.525 -9.525 16*0.5
CUBOID   0 10.875 -10.875 10.875 -10.875 10.24 -10.24 16*0.5
BOX TYPE  2
CYLINDER  1 5.748 5.3825 -5.3825 16*0.5
CUBOID   0 6.59 -15.16 6.59 -15.16 6.225 -14.255 16*0.5
BOX TYPE  3
CYLINDER  1 5.748 5.3825 -5.3825 16*0.5
CUBOID   0 6.59 -15.16 15.16 -6.59 6.225 -14.255 16*0.5
BOX TYPE  4
CYLINDER  1 5.748 5.3825 -5.3825 16*0.5
CUBOID   0 6.59 -15.16 6.59 -15.16 14.255 -6.225 16*0.5
BOX TYPE  5
CYLINDER  1 5.748 5.3825 -5.3825 16*0.5
CUBOID   0 6.59 -15.16 15.16 -6.59 14.255 -6.225 16*0.5
DONE
 2 2 2  0  1
1 3R2 1 2 1 1 2 1 0 2 9R1 0 3 3R1 2 2 1 3R1 0 4 6R1 2 2 1 0
5 3R1 2 2 1 2 2 1 1
 -1
 1
4 AQUEOUS 4 METAL$
 10.0 7.5
-10.0 60.0 -10.0 60.0 -10.0 60.0
 -8500.0 4500.0 7000.0
 1 1 1
 0 2
END
```

F12.6.2 Sample Problem Computer Input and Output


JJ UU UU NN NN EEEEEEEEEEEEE BBBBBBBBBBBB UU UU GGGGGGGGGG
JJ UU UU NNN NN EEEEEEEEEEEEE BBBBBBBBBBBBB UU UU GGGGGGGGGGGG
JJ UU UU NNNN NN EE BB BB UU UU GG GG
JJ UU UU NN NN NN EE BB BB UU UU GG
JJ UU UU NN NN NN EE BB BB UU UU GG
JJ UU UU NN NN NN EEEEEEEEE BBBBBBBBBBBB UU UU GG GGGGGG
JJ UU UU NN NN NN EEEEEEEEE BBBBBBBBBBBB UU UU GG GGGGGG
JJ UU UU NN NN NN EE BB BB UU UU GG GG
JJ UU UU NN NN NN EE BB BB UU UU GG GG
JJ JJ UU UU NN NNNN EE BB BB UU UU GG GG
JJJJJJJJJJJJ UUUUUUUUUUUUU NN NNN EEEEEEEEEEEEE BBBBBBBBBBBBB UUUUUUUUUUUUU GGGGGGGGGGGGG
JJJJJJJJJJ UUUUUUUUUU NN NN EEEEEEEEEEEEE BBBBBBBBBBB UUUUUUUUUU GGGGGGGGGG


MM MM BBBBBBBBBBBB EEEEEEEEEEEEE 44 AAAAAAAA 44 MM MM
MMM MMM BBBBBBBBBBBBB EEEEEEEEEEEEE 444 AAAAAAAAAA 444 MMM MMM
MMMM MMMM BB BB EE 4444 AA AA 4444 MMMM MMMM
MM MM MM MM BB BB EE 44 44 AA AA 44 44 MM MM MM MM
MM MM MM MM BB BB EE 44 44 AA AA 44 44 MM MM MM MM
MM MMM MM BBBBBBBBBBBB EEEEEEEEE 44 44 AAAAAAAAAAAA 44 44 MM MMM MM
MM M MM BBBBBBBBBBBB EEEEEEEEE 44 44 AAAAAAAAAAAA 44 44 MM M MM
MM MM BB BB EE 4444444444444 AA AA 4444444444444 MM MM
MM MM BB BB EE 4444444444444 AA AA 4444444444444 MM MM
MM MM BB BB EE 44 AA AA 44 MM MM
MM MM BBBBBBBBBBBB EEEEEEEEEEEEE 44 AA AA 44 MM MM
MM MM BBBBBBBBBBB EEEEEEEEEEEEE 44 AA AA 44 MM MM


11 0000000 // 22222222222 44 // 88888888888 44
111 000000000 // 2222222222222 444 // 8888888888888 444
1111 00 00 // 22 22 4444 // 88 88 4444
11 00 00 // 22 44 44 // 88 88 44 44
11 00 00 // 22 44 44 // 88 88 44 44
11 00 00 // 22 44 44 // 88888888888 44 44
11 00 00 // 22 44 44 // 88888888888 44 44
11 00 00 // 22 4444444444444 // 88 88 4444444444444
11 00 00 // 22 4444444444444 // 88 88 4444444444444
11 00 00 // 22 44 // 88 88 44
11111111 000000000 // 2222222222222 44 // 88888888888888 44
11111111 0000000 // 2222222222222 44 // 88888888888 44


11 5555555555555 11 99999999999 0000000 44
111 5555555555555 111 9999999999999 000000000 444
1111 55 1111 99 99 00 00 4444
11 55 11 99 99 00 00 44 44
11 55 11 99 99 00 00 44 44
11 555555555555 11 9999999999999 00 00 44 44
11 5555555555555 ..... 11 999999999999 ..... 00 00 44 44
11 55 ..... 11 99 ..... 00 00 4444444444444
11 55 ..... 11 99 ..... 00 00 4444444444444
11 55 55 ..... 11 99 ..... 00 00 44
11111111 5555555555555 11111111 9999999999999 000000000 44
11111111 55555555555 11111111 999999999999 0000000 44

F12.6.4

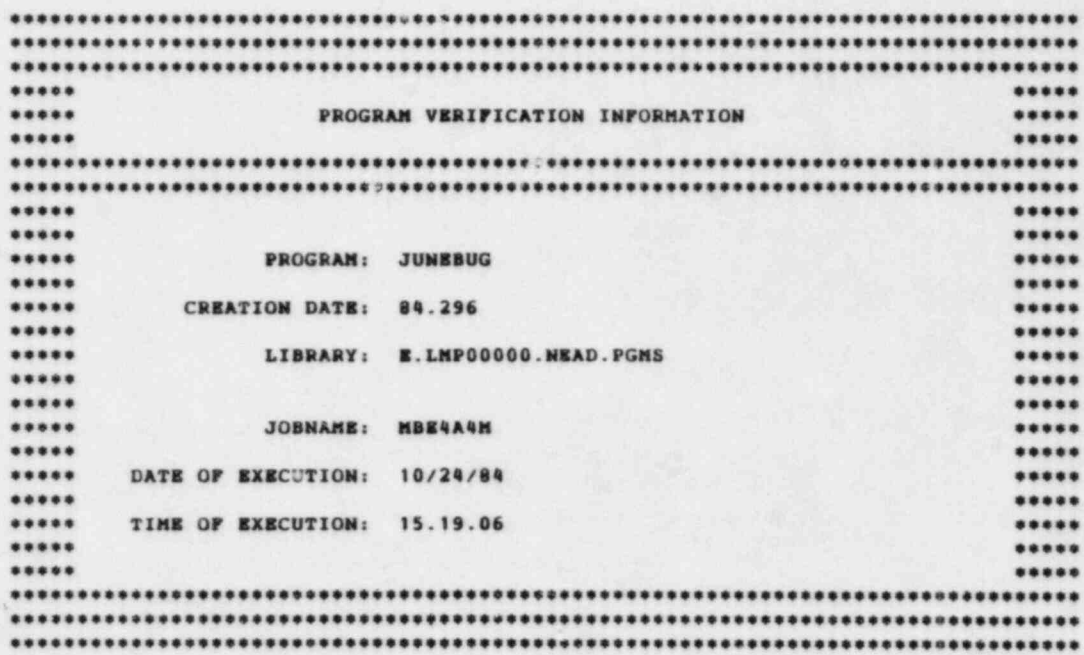


Fig. F12.6.2 (cont)

```
                JUNEBUG ISOMETRIC GEOMETRY DISPLAY

STRAIGHT LINE RESOLUTION FACTORS
     MINIMUM INCREMENTS                          5
     MAXIMUM INCREMENTS                         50
     SEGMENT SIZE FACTOR                    0.5000

CIRCULAR RESOLUTION FACTORS
     MINIMUM INCREMENTS                          5
     MAXIMUM INCREMENTS                         50
     SEGMENT SIZE FACTOR                    0.2000

TRANSPARENCY DATA (0-INPUT/1-CODE/2-MEDIA)       2

BOUNDARY TEST (0-UNIVERSE/1-ARRAY)               0

ERROR FLAG (0-EXIT ON ERROR/1-CONTINUE)          0

GEOMETRY TYPE (0-COMJOM/1-KENO IV)               1

DEBUG PRINT LEVEL                               -1

DRIVER CALLING JOMIN
```

Fig. F12.6.2 (cont)

```
 4 AQUEOUS 4 METAL

          IVOPT = 0          IDBG = 0


                                                    BODY DATA
 RCC     1  0.0              0.0             -0.9890000D+01  0.0             0.0             0.1778000D+02
            0.9525000D+01
 RCC     2  0.0              0.0             -0.9525000D+01  0.0             0.0             0.1905000D+02
            0.1016000D+02
 RPP     3 -0.1087500D+02    0.1087500D+02   -0.1087500D+02  0.1087500D+02  -0.1024000D+02   0.1024000D+02

 RCC     4  0.0              0.0             -0.5382500D+01  0.0             0.0             0.1076500D+02
            0.5748000D+01
 RPP     5 -0.1516000D+02    0.6590000D+01   -0.1516000D+02  0.6590000D+01  -0.1425500D+02   0.6225000D+01

 RCC     6  0.0              0.0             -0.5382500D+01  0.0             0.0             0.1076500D+02
            0.5748000D+01
 RPP     7 -0.1516000D+02    0.6590000D+01   -0.6590000D+01  0.1516000D+02  -0.1425500D+02   0.6225000D+01

 RCC     8  0.0              0.0             -0.5382500D+01  0.0             0.0             0.1076500D+02
            0.5748000D+01
 RPP     9 -0.1516000D+02    0.6590000D+01   -0.1516000D+02  0.6590000D+01  -0.6225000D+01   0.1425500D+02

 RCC    10  0.0              0.0             -0.5382500D+01  0.0             0.0             0.1076500D+02
            0.5748000D+01
 RPP    11 -0.1516000D+02    0.6590000D+01   -0.6590000D+01  0.1516000D+02  -0.6225000D+01   0.1425500D+02

 RPP    12  0.0              0.0              0.0            0.0             0.0             0.0

 RPP    13 -0.9999999D+15    0.9999999D+15   -0.9999999D+15  0.9999999D+15  -0.9999999D+15   0.9999999D+15

NUMBER OF BODIES      13
LENGTH OF FPD-ARRAY  110
```

Fig. F12.6.2 (cont)

INPUT ZONE DATA

INPUT ZONE	CODE ZONE	BODY NUMBERS
1	1	1
2	2	2 -1
3	3	3 -2
4	4	4
5	5	5 -4
6	6	6
7	7	7 -6
8	8	8
9	9	9 -8
10	10	10
11	11	11 -10
12	12	12
13	13	13 -3
14	14	13 -5
15	15	13 -7
16	16	13 -9
17	17	13 -11
18	18	13 -12

NUMBER OF INPUT ZONES 18
NUMBER OF CODE ZONES 18
LENGTH OF INTEGER ARRAY 439

Fig. F12.6.2 (cont)

CODE ZONE	INPUT ZONE	ZONE DATA LOC.	NO. OF BODIES	REGION NO.	MEDIA NO.	BOX INPUT ZONE	BOX CODE ZONE
1	1	92	1	1	2	1	1
2	2	98	2	1	3	1	1
3	3	109	2	1	1000	1	1
4	4	120	1	1	1	2	2
5	5	126	2	1	1000	2	2
6	6	137	1	1	1	3	3
7	7	143	2	1	1000	3	3
8	8	154	1	1	1	4	4
9	9	160	2	1	1000	4	4
10	10	171	1	1	1	5	5
11	11	177	2	1	1000	5	5
12	12	188	1	1	-1	0	0
13	13	194	2	1	-1000	1	1
14	14	205	2	1	-1000	2	2
15	15	216	2	1	-1000	3	3
16	16	227	2	1	-1000	4	4
17	17	238	2	1	-1000	5	5
18	18	249	2	1	1000	0	0

I	KR1(I)	KR2(I)
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18

```
MORSE REGION IN INPUT ZONE(I) ARRAY  (MRIZ(I),I=1,18)

    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

MORSE MEDIA IN INPUT ZONE(I) ARRAY (MMIZ(I),I=1,18)

    2    3 1000    1 1000    1 1000    1 1000    1 1000   -1-1000-1000-1000-1000-1000 1000

MORSE UNIVERSE IN INPUT ZONE(I) ARRAY (NBIZ(I),I=1,18)

    1    1    1    2    2    3    3    4    4    5    5    0    1    2    3    4    5    0
```

Fig. F12.6.2 (cont)

```
 OPTION 0 WAS USED IN CALCULATING VOLUMES, FOR  1 REGIONS
0-SET VOLUMES = 1, 1-CONCENTRIC SPHERES, 2-SLABS, 3-INPUTVOLUMES.

          VOLUMES ( CC ) USED IN COLLISIONS DENSITY AND TRACK LENGTH ESTIMATORS.
   REG        1
VOLUME  1.000D+00


                ARRAY NO.     1      ARRAY SIZE IS     2 BY     2 BY     2

                LEVEL      1 OF ARRAY NO.      1

    X =    1  2
 Y

 2         3  1

 1         2  1

                LEVEL      2 OF ARRAY NO.      1

    X =    1  2
 Y

 2         5  1

 1         4  1

                     UNIVERSE SPECIFICATIONS

                  UNIVERSE          1-SIMPLE/0-COMJOM


                       1                    1

                       2                    1

                       3                    1

                       4                    1

                       5                    1

 ARRAY DATA REQUIRES        107 LOCATIONS, LEAVING      41780 LOCATIONS
```

Fig. F12.6.2 (cont)

CODE ZONE	INPUT ZONE	ZONE DATA LOC.	NO. OF BODIES	REGION NO.	MEDIA NO.	BOX INPUT ZONE	BOX CODE ZONE
1	1	92	1	1	2	1	1
2	2	98	2	1	3	1	1
3	3	109	2	1	1000	1	1
4	4	120	1	1	1	2	2
5	5	126	2	1	1000	2	2
6	6	137	1	1	1	3	3
7	7	143	2	1	1000	3	3
8	8	154	1	1	1	4	4
9	9	160	2	1	1000	4	4
10	10	171	1	1	1	5	5
11	11	177	2	1	1000	5	5
12	12	188	1	1	-1	0	0
13	13	194	2	1	-1000	1	1
14	14	205	2	1	-1000	2	2
15	15	216	2	1	-1000	3	3
16	16	227	2	1	-1000	4	4
17	17	238	2	1	-1000	5	5
18	18	249	2	1	1000	0	0

I	KR1(I)	KR2(I)
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18

```
MORSE REGION IN INPUT ZONE(I) ARRAY  (MRIZ(I),I=1,18)

    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

MORSE MEDIA IN INPUT ZONE(I) ARRAY (MMIZ(I),I=1,18)

    2    3 1000    1 1000    1 1000    1 1000    1 1000   -1-1000-1000-1000-1000-1000 1000

MORSE UNIVERSE IN INPUT ZONE(I) ARRAY (NBIZ(I),I=1,18)

    1    1    1    2    2    3    3    4    4    5    5    0    1    2    3    4    5    0
```

Fig. F12.6.2 (cont)

```
 OPTION 0 WAS USED IN CALCULATING VOLUMES, FOR  1 REGIONS
0-SET VOLUMES = 1, 1-CONCENTRIC SPHERES, 2-SLABS, 3-INPUTVOLUMES.

          VOLUMES ( CC ) USED IN COLLISIONS DENSITY AND TRACK LENGTH ESTIMATORS.
   REG        1
VOLUME  1.000D+00

                ARRAY NO.    1     ARRAY SIZE IS    2 BY    2 BY    2

                LEVEL     1 OF ARRAY NO.     1

   X =   1  2
 Y

 2       3  1

 1       2  1

                LEVEL     2 OF ARRAY NO.     1

   X =   1  2
 Y

 2       5  1

 1       4  1

                    UNIVERSE SPECIFICATIONS

                 UNIVERSE          1-SIMPLE/0-COMJOM


                      1                   1

                      2                   1

                      3                   1

                      4                   1

                      5                   1

ARRAY DATA REQUIRES       107 LOCATIONS, LEAVING     41780 LOCATIONS
```

Fig. F12.6.2 (cont)

BODY FLAG INPUT

BODY NO.	0 - SKIP / 1 - DRAW	0 - NO SCAN / 1 - SCAN
1	1	0
2	1	0
3	0	0
4	1	0
5	0	0
6	1	0
7	0	0
8	1	0
9	0	0
10	1	0
11	0	0
12	0	0
13	0	0

BEGIN UNIVERSE = 0
BEGIN UNIVERSE = 1
BEGIN UNIVERSE = 2
BEGIN UNIVERSE = 3
BEGIN UNIVERSE = 4
BEGIN UNIVERSE = 5

Fig. F12.6.2 (cont)

```
                    PLOTTING COMMENCING
                    ...................

..... DISSPLA VERSION 8.2 .....
NO. OF FIRST PLOT  1


PLOT VIEWPOINT WILL BE   -8.500D+03   4.500D+03   7.000D+03
```

INPUT ZONE	ZONE TRANSPARENCY CODE ZONE	INPUT TRANSPARENCY 0 - TRANSPARENT / 1 - NON TRANSPARENT 2 - INVISIBLE
1	1	1
2	2	1
3	3	0
4	4	1
5	5	0
6	6	1
7	7	0
8	8	1
9	9	0
10	10	1
11	11	0
12	12	0
13	13	0
14	14	0
15	15	0
16	16	0
17	17	0
18	18	0

Fig. F12.6.2 (cont)

```
                    ARRAY  INPUT

          ARRAY NO.      0 - DRAW / 1 - SKIP

              1              0


          PLOT WILL BE FOR FIRST  2 LEVELS ONLY



PLOT NO.    1 WITH THE TITLE
     4 AQUEOUS 4 METAL$
HAS BEEN COMPLETED.

PLOT ID. READS
PLOT 1      15.20.00   WED 24 OCT, 1984
 JOB=MBE4A4M ,  ISSCO   DISSPLA VER  8.2
```

Fig. F12.6.2 (cont)

```
.................................
.                               .
.   WORKBOX-DIMENSIONS          .
.   ------------------          .
.                               .
.      X3DAXIS=  80.00          .
.      Y3DAXIS=  80.00          .
.      Z3DAXIS=  80.00          .
.           IN ABS. 3-D UNITS   .
.                               .
.................................
.                               .
.   VIEWPOINT                   .
.   ---------                   .
.                               .
.      XVU=-8.500E+03           .
.      YVU= 4.500E+03           .
.      ZVU= 7.000E+03           .
.           IN ABS. 3-D UNITS   .
.                               .
.................................
.                               .
.   GRAPH SET-UP ( GRAF3D )     .
.   ------------                .
.                               .
.     ORIGIN                    .
.     ------                    .
.      X3DORIGIN=-2.000E+01     .
.      Y3DORIGIN=-2.000E+01     .
.      Z3DORIGIN=-2.000E+01     .
.                               .
.     STEP SIZE                 .
.     ---------                 .
.      X3DSTP= 2.000E+01        .
.      Y3DSTP= 2.000E+01        .
.      Z3DSTP= 2.000E+01        .
.                               .
.     MAXIMUM                   .
.     -------                   .
.      X3DMAX= 6.000E+01        .
.      Y3DMAX= 6.000E+01        .
.      Z3DMAX= 6.000E+01        .
.                               .
.................................

  .....................................
  . LOCATION OF CURRENT PHYSICAL ORIGIN .
  . X=   1.25      Y=   0.81     INCHES .
  .    FROM LOWER LEFT CORNER OF PAGE   .
  .....................................

END OF DISSPLA 8.2 --      1549 VECTORS GENERATED IN      1 PLOT FRAMES.
-ISSCO-  4186 SORRENTO VALLEY BLVD.,SAN DIEGO CALIF. 92121

DISSPLA IS A CONFIDENTIAL PROPRIETARY PRODUCT OF ISSCO AND ITS USE
   IS SUBJECT TO A NONDISSEMINATION AND NONDISCLOSURE AGREEMENT.
```

Fig. F12.6.2 (cont)

COMBINATORIAL GEOMETRY ZONE TRANSFER SUMMARY TABLES

EXIT ZONE	EXIT BODY	LOC. IN MA ARRAY	ENTERING ZONE	NO. OF ENTRIES	LOC. FOR NEXT ZONE
1	1	95	2	202	0
2	2	101	3	141	0
2	-1	106	1	213	0
3	3	112	13	80	0
3	-2	117	2	876	0
4	4	123	5	67	0
5	5	129	14	32	0
5	-4	134	4	199	0
6	6	140	7	46	0
7	7	146	15	102	0
7	-6	151	6	252	0
8	8	157	0	0	0
9	9	163	16	259	0
9	-8	168	8	338	0
10	10	174	11	75	0
11	11	180	17	571	0
11	-10	185	10	385	0
12	12	191	0	0	0
13	13	197	0	0	0
13	-3	202	0	0	0
14	13	208	0	0	0
14	-5	213	0	0	0
15	13	219	0	0	0

Fig. F12.6.2 (cont)

COMBINATORIAL GEOMETRY ZONE TRANSFER SUMMARY TABLES

EXIT ZONE	EXIT BODY	LOC. IN MA ARRAY	ENTERING ZONE	NO. OF ENTRIES	LOC. FOR NEXT ZONE
15	-7	224	0	0	0
16	13	230	0	0	0
16	-9	235	0	0	0
17	13	241	0	0	0
17	-11	246	0	0	0
18	13	252	0	0	0
18	-12	257	12	1951	0

Fig. F12.6.2 (cont)

NUREG/CR-0200
VOLUME 3
ORNL/NUREG/CSD-2/V3/R2

SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

REVISIONS AND ERRATA SHEETS FOR VOLUME 3: MISCELLANEOUS

Print on Green Cover Stock

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
Under Interagency Agreement 40-549-75

Corrected 12/84

M2.2.69 YREAD

CALLING SEQUENCE:

CALL YREAD (DD,LD,LIM,IT,LERR)

FUNCTION:

YREAD is used to read an array using the free-form reading package while allowing interpolation, filling, skipping, multiple repeats, and arbitrary indexing.
(See Section M3 for a more complete description.)

ARGUMENTS:

DD,LD is the array to be read. DD is used for floating point numbers and LD for integers.
LIM is the number of entries in the array.
IT selects what kind of data are being read.
IT=0 selects integers.
IT=1 selects single precision floating point numbers.
IT=2 selects double precision floating point numbers.
LERR logical flag which is set true if an attempt is made to store data outside the DD or LD array.

SUBROUTINES:

DREAD,AREAD

COMMONS:

UNIT,QRDBUF

LANGUAGE:

FORTRAN

M2.2.70 YOPACK

CALLING SEQUENCE:

WORD=YOPACK(WORD,CHAR,I)

FUNCTION:

YOPACK is used to insert a character into a word at an arbitrary character position.

ARGUMENTS:

WORD is the word into which the character will be inserted.
CHAR is the character to be inserted.
I is the character position in WORD at which CHAR will be inserted.

LANGUAGE:

FORTRAN

NUREG/CR-0200
Volume 3, Section M6
ORNL/NUREG/CSD-2/V3/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

ORIGEN-S DATA LIBRARIES

J. C. Ryman

Manuscript Completed: February 1983
Date Published: June 1983
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

LIST OF TABLES

		Page
Table M6.2.1	Units of Half-Life Indicated by the Variable IU	M6.2.2
Table M6.2.2	Definitions of Branching Fraction Variables	M6.2.3
Table M6.2.3	Definitions of Nuclear Data Variables in Light Element Libraries for a Thermal Reactor	M6.2.5
Table M6.2.4	Reactor Type as a Function of Variable IT	M6.2.6
Table M6.2.5	Definitions of Spectral Indices for Thermal Reactors	M6.2.7
Table M6.2.6	Definitions of Nuclear Data Variables in the Actinide Library for a Thermal Reactor	M6.2.8
Table M6.2.7	Fissionable Nuclides Having Yield Data	M6.2.10
Table M6.2.8	Sources of Half-Lives, Recoverable Energies, Gamma Energies, and Branching Fractions	M6.2.16
Table M6.2.9	Explanation of Key Numbers Referring to Sources of Decay Data	M6.2.45
Table M6.3.1	Group Structures for Original Photon Yield Libraries	M6.3.1
Table M6.3.2	Group Structures for ORIGEN-S Photon Yield Libraries	M6.3.2
Table M6.3.3	FORMATS for Original Photon Yield Libraries	M6.3.4
Table M6.3.4	FORMATS for Current ORIGEN-S Photon Yield Libraries	M6.3.4
Table M6.7.1	Data in First Record of Binary Data Library	M6.7.2
Table M6.7.2	Structure of Binary Data Library	M6.7.3
Table M6.7.3	Initial Uranium Composition of Typical PWR Assembly	M6.7.10
Table M6.7.4	Nuclides with Updated Cross Sections in Typical PWR Libraries	M6.7.10
Table M6.8.1	DCB Parameters for ORIGEN-S Library Data Sets at ORNL	M6.8.1
Table M6.8.2	Card-image ORIGEN-S Libraries at ORNL	M6.8.2
Table M6.8.3	Some Binary ORIGEN-S Libraries at ORNL	M6.8.3

ACKNOWLEDGMENTS

The updating and description of the ORIGEN-S libraries could not have been accomplished without the advice, assistance, and encouragement of O. W. Hermann, who deserves much of the credit for this work.

Special thanks are due to A. G. Croff and C. W. Alexander of the ORNL Chemical Technology Division, who provided the ORNL Master Photon Data Base and much of the decay data. K. F. Eckerman and S. B. Watson also provided a portion of the decay data. The author is grateful to M. A. Bjerke and R. W. Roussin for their help in accessing the ENDF/B-V decay data.

The author would like to acknowledge M. J. Bell, who developed the original ORIGEN libraries, and C. W. Kee, G. W. Morrison, and R. E. Schenter, who developed the big light element and/or fission product libraries for ORIGEN.

Many colleagues at ORNL provided helpful suggestions and advice. In particular, the author would like to thank L. M. Petrie, J. P. Renier, and N. M. Greene. He would like to acknowledge the encouragement and supervision provided by G. E. Whitesides, R. M. Westfall, and especially C. V. Parks. He is further indebted to O. W. Hermann and C. V. Parks for their review of the manuscript.

At the U. S. Nuclear Regulatory Commission, the author would like to thank R. H. Odegaarden and other staff members for sponsoring and directing the development of ORIGEN-S. Special thanks are also due to P. Worthington and others of the USNRC Division of Engineering Technology for sponsoring the updating of the card-image nuclear data libraries.

The author gratefully thanks R. J. Patton and L. Norris for carefully preparing the drafts, final manuscript, and revisions.

M6.4 DATA PROGRAMMED INTO ORIGEN-S AND COUPLE

Several items of nuclear data are programmed directly into ORIGEN-S and COUPLE. Descriptions of these data are presented here. Although these data are not part of the ORIGEN-S libraries, the user may on occasion wish to change these data or to understand explicitly how they are used.

The arrays EACTGP and EGROUP are defined by DATA statements in subroutine FLXDIM in ORIGEN-S and in subroutine CARDIM in COUPLE. The array EACTGP contains the average energies of the 18 groups in the original card-image actinide photon yield library. The array EGROUP contains the average energies of the 12 groups in the original card-image light element and fission product photon yield libraries. These are the average energies of Table M6.3.1. It should be noted that the arrays EACTGP, EGROUP, and EFPGRP can be read as input data in the 36*, 37* and 39* arrays of ORIGEN-S (Sect. F7.6.10) or COUPLE (Sect. F6.5.8). When read as input data, these arrays contain the energy group boundaries (not the average group energies) for the card-image light element, actinide, and fission product photon yield libraries. These data should be read in only when the user has made his own card-image photon yield libraries to be read using the original ORIGEN library FORMATS of Table M6.3.3, and then only if different energy group boundaries have been used. These data should not be read when photon yield libraries with the new ORIGEN-S variable FORMATS are used.

A number of data in the ORIGEN-S COMMON/SPECDT/ are defined by DATA statements in a Block Data subroutine of ORIGEN-S. These data, used in the calculation of spontaneous fission and (α,n) neutron sources (Sect. F7.2.8), are described below in their order of appearance in the COMMON block. These same data are also defined for the SAS2 COMMON/SPECDT/ in a Block Data subroutine of the SAS2 control module (Sect. S2.2.7).

(1) The variable NGAL is the number of energy groups in which neutron spectra data are provided for (α,n) neutron sources due to ^{242}Cm, ^{244}Cm, and ^{238}Pu.

(2) The variable NGSP is the number of energy groups in which neutron spectra data are provided for spontaneous fission neutron sources from ^{242}Cm and ^{244}Cm.

(3) The array GRPAL contains the energy group boundaries (MeV) for neutron spectra from (α,n) reactions.

(4) The array GRPSP contains the energy group boundaries (MeV) for neutron spectra from spontaneous fission.

(5) The array SA38 contains the neutron spectrum from (α,n) reactions due to alpha decays in ^{238}Pu.

(6) The array SA42 contains the neutron spectrum from (α,n) reactions due to alpha decays in ^{242}Cm.

(7) The array SA44 contains the neutron spectrum from (α,n) reactions due to alpha decays in ^{244}Cm.

(8) The array SS42 contains the neutron spectrum from spontaneous fission of ^{242}Cm.

(9) The array SS44 contains the neutron spectrum from spontaneous fission of ^{244}Cm.

(10) The array EALPH contains the mean alpha energies (MeV) for those actinides which decay by alpha emission.

(11) The array FA contains the alpha decay branching fractions for the actinides.

(12) The array SPONU contains the values of the spontaneous fission yield ν_{SF}, for those actinides which have measured data.

(13) The array HALFSP contains spontaneous fission half-lives (years) for a number of actinides.

(14) The array AEK contains data for an empirical formula used to estimate the spontaneous fission yield for those actinides having no measured value of ν_{SF}.

The array MSRS in subroutine FLXDI1 in ORIGEN-S, and in subroutine CARDIM in COUPLE, contains the identification numbers of the fissionable nuclides which have fission product yield data in the card-image nuclear data libraries (Table M6.2.7).

There are a number of arrays defined by DATA statements in subroutine POWERF in ORIGEN-S which contain data used to compute the recoverable energy (MeV per fission) for converting power to flux. The array IDL contains the identification numbers of ten light element nuclides having values of recoverable energy per neutron capture in the array QCLIT. The array QNFISS contains values of recoverable energy per (n,fission) reaction for 30 actinides whose identification numbers are given in array IDA. The array QCACT contains values of recoverable energy per neutron capture for the same 30 actinides. The array IDF has identification numbers for 30 fission products whose values of recoverable energy per neutron capture are given in the array QCFPR. The recoverable energy values in arrays QCLIT, QCACT, QNFISS, and QCFPR were taken primarily from ENDF.[5,23] The recoverable energy per neutron capture is assumed to be 5.0 MeV for all nuclides for which data are not given in the DATA statements.

Table M6.7.1 (Cont.)

Variable Name	Description
THERM	Thermal group flux weighting factor.
RES	Resonance group flux weighting factor.
FAST	Fast group flux weighting factor.
ERR	Truncation error limit below which values computed by ORIGEN-S are considered to be zero.
IDREFS	ID number of nuclide in AMPX weighted cross section library containing the weighting spectrum used to obtain the flux weighting factors THERM, RES, and FAST.
NEWID	ID number of binary data library.

A summary description of the data in all the records of a binary data library is given in Table M6.7.2. Most of these data have been described in earlier sections of this report.

Table M6.7.2 Structure of Binary Data Library

Record	Length[a]	Array	Description
1	20	---	Header information of Table M6.7.1.
2	20*NTIT[b]	TITLE	Title cards containing descriptive information.
3	NFISO	NUCFIS	ID numbers of fissionable nuclides having fission product yields.
4	3*NFISO	SIGFIS	Three-group fission cross sections for nuclides in NUCFIS array in the order ((SIGFIS(N,I),N=1,3),I=1,NFISO).
5	ITOT	NUCL	ID numbers for all nuclides in library.
6	ITOT	KD[c]	KD(I) is the number of parents for production of nuclide I by decay transitions.
7	ITOT	NONO[c]	NONO(I) is the total number of parents for nuclide I.
8	NON	LOC[c]	LOC(J) is the position in array NUCL of the parent of the transition stored in position J of the compressed transition matrix A.
9	NON	A_1^c	Compressed transition matrix for thermal group neutron capture transitions.
10	ITOT	$TOCAP_1^c$	Thermal group total capture cross sections.

Table M6.7.2 (Cont.)

Record	Length[a]	Array	Description
11	IACT	$FISS_1$	Thermal group fission cross sections.
12	ILITE+IACT	$GENNEU_1$	Thermal group neutron yields, expressed as the sum of the product of neutrons per capture reaction times the capture cross section for that reaction, summed over all neutron-producing reactions.
13	NON	A_2^c	Compressed transition matrix for resonance group neutron capture transitions.
14	ITOT	$TOCAP_2^c$	Resonance group total capture cross sections.
15	IACT	$FISS_2$	Resonance group fission cross sections.
16	ILITE+IACT	$GENNEU_2$	Resonance group neutron yields.
17	NON	A_3^c	Compressed transition matrix for fast group neutron capture transitions.
18	ITOT	$TOCAP_3^c$	Fast group total capture cross sections.
19	IACT	$FISS_3$	Fast group fission cross sections.
20	ILITE+IACT	$GENNEU_3$	Fast group neutron yields.
21	NON	A_λ^c	Compressed transition matrix for decay transitions.
22	ITOT	DIS^c	Radioactive decay constants.
23[e]	NON	$A^{c,d}$	Compressed transition matrix for effective one-group neutron capture transitions and for decay transitions.
24[e]	ITOT	$TOCAP^{c,d}$	Effective one-group capture cross sections.
25[e]	IACT	$FISS^d$	Effective one-group fission cross sections.
26[e]	ILITE+IACT	$GENNEU^d$	Effective one-group neutron yields.
27	ITOT	Q^f	Recoverable energy values.
28	ITOT	FG^f	Fractions of Q-values due to photons.
29	ITOT	$AMPC^f$	RCG values for air.
30	ITOT	$WMPC^f$	RCG values for water.
31	ILITE	$ABUND^f$	Natural abundances of light element nuclides.
32	IACT	SPONF	Values of the product of the decay constant times the spontaneous fission branching fraction times Avagadro's number (no longer used).
33	IACT	ALPHAN	Values of the product of the decay constant times the alpha decay branching fraction times Avagadro's number times the number of neutrons per alpha decay (no longer used).
34	NENLE+1	EGROUP	Mean photon group energies (NENLE values) for light element photon yield data, ordered from low energy to high. The last value is the lower energy bound of group 1.

Table M6.7.2 (Cont.)

Record	Length[a]	Array	Description
35	NENAC+1	EACTGP	Mean photon group energies (NENAC values) for actinide photon yield data, ordered from low energy to high. The last value is the lower energy bound of group 1.
36	NENFP+1	EFPGRP	Mean photon group energies (NENFP values) for fission product photon yield data, ordered from low energy to high. The last value is the lower energy bound of group 1.
37	NENAC	SFGAMA	Multigroup photon yields for gamma rays accompanying spontaneous fission, in the group structure specified by EACTGP. See Sect. M6.4 for details.
38	NENLE*ILITE	GAMGRP	Multigroup light element photon yields, in the order ((GAMGRP(N,I),N=1,NENLE),I=1, ILITE).
39	NENFP*IFP	GFPGRP	Multigroup fission product photon yields, in the order ((GFPGRP(N,I),N=1,NENFP), I=1,IFP).
40	NENAC*IACT	ACTGRP	Multigroup actinide photon yields, in the order ((ACTGRP(N,I),N=1,NENAC),I=1,IACT).

[a] Length in single-precision words.
[b] NTIT is the number of title cards.
[c] These arrays are discussed in Sect. M6.2.5.
[d] Effective one-group values have been weighted with THERM, RES, and FAST.
[e] Records 23 through 26 are present only in a binary working library.
[f] These variables are defined in Sect. M6.2.1.

M6.7.3 GENERATION OF SOME USEFUL LIBRARIES

As an aid to the beginning user, this section presents the input data for COUPLE, ORIGEN-S, and SAS2 cases to generate some binary data libraries which have been useful at ORNL.

The first example contains back-to-back runs of COUPLE and ORIGEN-S. The first COUPLE case converts the data in the large light element, actinide, and large fission product card-image nuclear data libraries to a binary working library for an LWR. The flux-weighting factors used are THERM=0.632, RES=0.333, and FAST=2.0. The card-image nuclear data and photon yield libraries are read from logical unit 28. The binary working library generated by this case is written on logical unit 32.

The second COUPLE case reads the binary working library from logical unit 32, adds beta-delayed neutron (β,n) decay transitions for 101 fission product nuclides, and writes an updated binary working library on logical unit 33. It also writes an updated binary master library on logical unit 31 (for which a DD card defining a scratch unit must be included in the JCL), but this library is not saved.

The beta (negatron) decay transitions from the ORIGEN-S card-image nuclear data libraries include the (β,n) decay transitions. Therefore, 202 pairs of instructions in the 81$, 82$, and 83* arrays are necessary to add the 101 (β,n) decay transitions to the library. The 81$ array contains 101 pairs of repeated parent nuclide identification numbers. The 82$ array contains 101 corresponding pairs of daughter nuclide identification numbers, one of which gives the negatron decay daughter, while the other gives the (β,n) decay daughter. The 83* array has 101 pairs of corresponding decay constants (i.e., total decay constant times the corresponding branching fractions) derived from ENDF/B-V data.[23] The total decay constant for each parent nuclide is automatically corrected by COUPLE to reflect the changes specified by the 81$, 82$, and 83* arrays. Since the new negatron decay constant is less than the original by an amount equal to the (β,n) decay constant, the total decay constant remains unchanged.

Addition of the (β,n) decay transitions to a binary library will give better predictions of the concentrations of certain fission product nuclides and associated quantities such as afterheat power. Calculations indicate that correctly representing the (β,n) decay transitions will yield a calculated total afterheat power about 2-1/2 percent lower after ten years decay of "typical" pressurized water reactor fuel irradiated to a burnup of 33,000 MWd/MTU.

The (β,n) decay transitions have not been added to the card-image nuclear data libraries for two reasons. First, a new format would be required. Second, the integral option in ORIGEN-S (Sect. F7.2.7) will not work with a fission product library containing transitions which change the mass number of the daughter nuclide from that of the parent. Therefore, the binary library being generated in this example cannot be

used with the integral option. If an attempt is made to use the integral option with a library containing transitions with mass number changes, subroutine INTSET in ORIGEN-S will call subroutine ERROR which, at ORNL, generates a program ABEND (abnormal end).

The third COUPLE case generates an edit of the cross-section transitions in the binary working library on logical unit 33.

The ORIGEN-S case replaces the photon yield data in the binary working library with 18-group photon yields for the light elements, actinides, and fission products. The 18-group energy boundaries are those of Table M6.3.2. The photon yield data are read from a binary master photon data base on logical unit 26 which includes decay gamma- and x-rays, gamma rays accompanying spontaneous fission and (α,n) reactions, and bremsstrahlung for a UO fuel matrix. The input binary working library from COUPLE is read from logical unit 33. The final binary working library is written by ORIGEN-S on logical unit 35. This library has an ID of 1104, and is known at ORNL as the Basic LWR ORIGEN-S Binary Working Library. Details of the input data given below can be determined by studying Sect. F6.5.8 of the COUPLE document and Sect. F7.6.10 of the ORIGEN-S document.

```
=COUPLE
*********************************************************************************
*                                                                               *
*            BASIC LWR ORIGEN-S BINARY WORKING LIBRARY--ID = 1104               *
*     MADE FROM NEW CARD-IMAGE ORIGEN-S LIBRARIES ON 09/21/84 BY J.C. RYMAN     *
* DATA FROM LARGE LIGHT ELEMENT, ACTINIDE, AND LARGE FISSION PRODUCT LIBRARIES  *
*                                                                               *
*       NEUTRON FLUX SPECTRUM FACTORS THE SAME AS IN ORNL-4628 TEST CASE        *
*                (THERM = 0.632 -- RES = 0.333 -- FAST = 2.0)                   *
*                                                                               *
*   ENDF/B-V BETA-DELAYED NEUTRON DECAY TRANSITIONS FOR 101 FISSION PRODUCTS    *
*                                                                               *
*     PHOTON LIBRARIES USE NEW ACTINIDE 18-GROUP STRUCTURE OF O.W. HERMANN      *
*        PHOTON DATA FROM REVISED MASTER PHOTON DATA BASE OF 01/28/82           *
*             PHOTON DATA INCLUDE BREMSSTRAHLUNG FROM UO2 MATRIX                *
*                                                                               *
*        LIBRARY STORED IN DSNAME=E.CJR20161.ORIGENS.BINRYLIB(BASICLWR)         *
*                                                                               *
*********************************************************************************
(blank card)
0$$ A2 28 E
1$$ 2R1 A11 1103 E 2** A2 0.632 0.333 2.0 E 1T
3$$ 28 A18 1609 687 101 821 9500 E 5$$ 2 3T
35$$ 0 5T
(blank card)
0$$  A4 32 A6 33 E
1$$  A2 3R1  A11 1104  A16 18 E 1T
16$$ 0 0 202 6T
```

```
81$$
 300790 300790 310790 310790 310800 310800 310810 310810 310820 310820
 310830 310830 320830 320830 320840 320840 320850 320850 320860 320860
 330840 330840 330850 330850 330860 330860 330870 330870 340870 340870
 340880 340880 340890 340890 340900 340900 340910 340910 350870 350870
 350880 350880 350890 350890 350900 350900 350910 350910 350920 350920
 350930 350930 360920 360920 360930 360930 360940 360940 360950 360950
 370920 370920 370930 370930 370940 370940 370950 370950 370960 370960
 370970 370970 370980 370980 370990 370990 380970 380970 380980 380980
 380990 380990 381000 381000 390970 390970 390980 390980 390990 390990
 391000 391000 401040 401040 401050 401050 411030 411030 411040 411040
 411050 411050 411060 411060 421090 421090 421100 421100 431090 431090
 431100 431100 471220 471220 471230 471230 481280 481280 491270 491270
 491280 491280 491290 491290 491300 491300 491310 491310 491320 491320
 501330 501330 501340 501340 501350 501350 511340 511340 511350 511350
 511360 511360 511370 511370 521360 521360 521370 521370 521380 521380
 521390 521390 531370 531370 531380 531380 531390 531390 531400 531400
 531410 531410 531420 531420 531430 531430 541410 541410 541420 541420
 541430 541430 541440 541440 551410 551410 551420 551420 551430 551430
 551440 551440 551450 551450 551460 551460 551470 551470 561470 561470
 561480 561480 561490 561490 561500 561500 571470 571470 571490 571490
 571500 571500
82$$
 310790 310780 320790 320780 320800 320790 320810 320800 320820 320810
 320830 320820 330830 330820 330840 330830 330850 330840 330860 330850
 340840 340830 340850 340840 340860 340850 340870 340860 350870 350860
 350880 350870 350890 350880 350900 350890 350910 350900 360870 360860
 360880 360870 360890 360880 360900 360890 360910 360900 360920 360910
 360930 360920 370920 370910 370930 370920 370940 370930 370950 370940
 380920 380910 380930 380920 380940 380930 380950 380940 380960 380950
 380970 380960 380980 380970 380990 380980 390970 390960 390980 390970
 390990 390980 391000 390990 400970 400960 400980 400970 400990 400980
 401000 400990 411040 411030 411050 411040 421030 421020 421040 421030
 421050 421040 421060 421050 431090 431080 431100 431090 441090 441080
 441100 441090 481220 481210 481230 481220 491280 491270 501270 501260
 501280 501270 501290 501280 501300 501290 501310 501300 501320 501310
 511330 511320 511340 511330 511350 511340 521340 521330 521350 521340
 521360 521350 521370 521360 531360 531350 531370 531360 531380 531370
 531390 531380 541370 541360 541380 541370 541390 541380 541400 541390
 541410 541400 541420 541410 541430 541420 551410 551400 551420 551410
 551430 551420 551440 551430 561410 561400 561420 561410 561430 561420
 561440 561430 561450 561440 561460 561450 561470 561460 571470 571460
 571480 571470 571490 571480 571500 571490 581470 581460 581490 581480
 581500 581490
83**
   2.19361E+00   2.43981E-02   2.30703E-01   3.46575E-04   4.13719E-01
   3.84155E-03   5.35360E-01   2.81768E-02   9.12647E-01   2.42602E-01
   9.83827E-01   1.25214E+00   3.64196E-01   6.20187E-04   5.19862E-01
   5.77625E-02   2.21843E+00   5.54608E-01   2.19102E+00   6.17981E-01
   2.30750E+00   3.00365E-03   2.66334E-01   7.51197E-02   6.89299E-01
   8.08675E-02   1.29388E+00   1.01662E+00   1.23542E-01   2.35176E-04
```

```
 4.59789E-01   2.31050E-03   1.60608E+00   8.45304E-02   1.44403E+00
 1.78475E-01   2.02810E+00   5.39116E-01   1.21494E-02   2.94931E-04
 3.96329E-02   2.89167E-03   1.37680E-01   2.05729E-02   3.03446E-01
 8.16376E-02   1.02933E+00   1.25922E-01   1.50182E+00   4.23592E-01
 2.31994E+00   1.61216E+00   3.76588E-01   1.24315E-04   5.26988E-01
 1.07548E-02   3.11257E+00   1.88141E-01   8.04232E-01   8.44221E-02
 1.54702E-01   1.85665E-05   1.16605E-01   1.67965E-03   2.30878E-01
 2.67984E-02   1.66356E+00   1.60519E-01   2.92550E+00   4.72293E-01
 2.94385E+00   1.13350E+00   5.29314E+00   1.00822E+00   4.06722E+00
 7.17744E-01   1.73114E+00   1.73287E-03   1.06073E+00   5.65184E-03
 1.11597E+00   3.92785E-02   1.06561E+00   5.60846E-02   1.86720E-01
 6.18215E-04   3.44703E-01   1.87150E-03   4.89166E-01   5.94128E-03
 8.18783E-01   4.76541E-02   2.69118E-01   2.96356E-04   1.38734E+00
 1.96986E-02   4.61499E-01   6.00730E-04   1.43381E-01   1.02528E-03
 2.40374E-01   7.17905E-03   6.55027E-01   3.81232E-02   4.89511E-01
 2.60823E-03   2.46794E-01   3.25058E-03   4.86690E-01   8.41682E-03
 8.09232E-01   2.58887E-02   4.55631E-01   6.46940E-03   1.69555E+00
 8.17561E-02   6.57476E-01   7.24020E-04   1.82818E-01   1.53009E-03
 8.11150E-01   1.40280E-02   3.24649E-01   1.28240E-02   1.28965E+00
 1.81788E-02   2.43444E+00   4.10939E-02   5.53942E+00   2.36826E-01
 4.71436E-01   9.43061E-05   5.53187E-01   1.13303E-01   1.51648E+00
 1.42688E-01   6.29594E-02   5.41917E-05   3.27532E-01   5.33192E-02
 6.50885E-01   1.94420E-01   1.16045E+00   2.90112E-01   3.61532E-02
 3.28334E-04   1.93686E-01   4.35694E-03   4.08958E-01   2.42602E-02
 1.11976E+00   7.52878E-02   2.62548E-02   2.03701E-03   1.05489E-01
 2.81592E-03   2.61533E-01   2.97064E-02   6.28671E-01   1.77317E-01
 9.19177E-01   5.87670E-01   2.91123E+00   5.54520E-01   1.41709E+00
 3.11070E-01   4.02821E-01   1.73287E-04   5.65826E-01   2.32944E-03
 7.13367E-01   8.66437E-03   6.25536E-01   4.59999E-03   2.78226E-02
 1.47538E-05   4.09369E-01   7.79281E-04   3.83180E-01   6.23056E-03
 6.73761E-01   1.86963E-02   1.01035E+00   1.64476E-01   1.76549E+00
 2.73183E-01   9.53036E-01   3.17679E-01   3.74505E-01   2.05425E-02
 1.58657E-01   4.98279E-02   9.96781E-01   2.99124E-04   7.18659E-01
 1.72893E-03   1.37937E-01   6.93150E-04   2.85533E-01   2.33171E-03
 1.12918E+00   1.07150E-02   9T
EDIT
(blank card)
0$$ A4 33 E
1$$ A2 1 A5 1 A13 -1 E 1T
DONE
END
=ORIGENS
0$$ A4 33 A8 26 A11 35 E 1T
(blank card)
3$$ 33 A3 1 A6 1 A33 18 2T
4T
56$$ A2 1 A5 2Z A13 1 A17 1 E 5T
60** 1 73$$ 922380 74** 1 75$$ 2 81$$ 11 A3 26 1 E 82$$ 21
83** 6+6 5+6 4.5+6 4+6 3.5+6 3+6 2.6+6 2.2+6 1.8+6 1.35+6 9+5
     4+5 2+5 1.4+5 1+5 7+4 4+4 2+4 0 6T
*
1$$ 0 1T
END
```

The second example is a SAS2 run to generate three burnup-dependent binary working libraries for a "typical" pressurized water reactor (PWR) fuel assembly with a burnup of 33 GWD/MTU and the initial uranium composition of Table M6.7.3. The irradiation history is divided into three cycles having 80% full power irradiation at 37.5 MW/MTU and 20% downtime. Updated cross sections are computed from a 27-group unit cell calculation, at the midpoint of each irradiation cycle, for the nuclides given in Table M6.7.4. The resonance cross section processing and unit cell calculations are performed for the geometry and operating conditions of a 17x17 PWR fuel assembly. The cross sections used in the calculations are read from the SCALE 27-neutron-group ENDF/B-IV shielding library (Sect. M4) on logical unit 82. The input binary library is the Basic LWR ORIGEN-S Binary Working Library generated in the first example outlined above, which is read on logical unit 21. The burnup-dependent binary working libraries for cycles 1, 2, and 3 are written on logical units 33, 34, and 35, respectively. None of these libraries can be used with the integral option. Details of the SAS2 method and input data requirements can be found in Sects. S2.2 and S2.5, respectively, of the SAS2 documentation.

Table M6.7.3. Initial Uranium Composition of Typical PWR Assembly

Nuclide	Weight Percent
^{234}U	0.029
^{235}U	3.200
^{236}U	0.016
^{238}U	96.755

Table M6.7.4. Nuclides with Updated Cross Sections in Typical PWR Libraries

^{1}H, ^{10}B, ^{11}B, ^{16}O, ^{59}Co, ^{90}Sr, ^{89}Y, ^{94}Zr, ^{95}Zr, ^{94}Nb
^{99}Tc, ^{103}Rh, ^{105}Rh, ^{131}Xe, ^{135}Xe, ^{133}Cs, ^{134}Cs, ^{137}Cs, ^{136}Ba, ^{139}La
^{143}Pr, ^{143}Nd, ^{145}Nd, ^{147}Pm, ^{149}Sm, ^{151}Sm, ^{152}Sm, ^{153}Eu, ^{154}Eu, ^{234}U
^{235}U, ^{236}U, ^{238}U, ^{237}Np, ^{238}Pu, ^{239}Pu, ^{240}Pu, ^{241}Pu, ^{242}Pu, ^{241}Am
^{243}Am, ^{244}Cm

The input data for this SAS2 case are given below. The last four cards before the "END" card have no meaning for this case, but must be present for SAS2 to execute properly.

Table M6.8.3. Some Binary ORIGEN-S Libraries at ORNL

Member Name	Description
BASICLWR	Basic LWR ORIGEN-S Binary Working Library described in Sect. M6.7.3. It has (β,n) decay data for fission products, so it cannot be used with the integral option.
BASLMFBR	Basic LMFBR ORIGEN-S Binary Working Library. This library was converted from the large light element, actinide, and large fission product card-image nuclear data libraries. It has the same (β,n) decay data and photon yield data as the Basic LWR Binary Working Library. It cannot be used with the integral option.
MAPHH2OB	Binary master photon library with the photon data from members MPDKXGAM, MPSFANGM, MPBRH2OM, and MPBRH2OP of the card-image library.
MAPHNOBR	Binary master photon library with the photon data from members MPDKXGAM and MPSFANGM of the card-image library.
MAPHUO2B	Binary master photon library with the photon data from members MPDKXGAM, MPSFANGM, MPBRUO2M, and MPBRUO2P of the card-image library.
PWR33CY1	Binary working library for cycle 1 of a "typical" PWR, as described in Sect. M6.7.3. It cannot be used with the integral option.
PWR33CY2	Binary working library for cycle 2 of a "typical" PWR. It cannot be used with the integral option.
PWR33CY3	Binary working library for cycle 3 of a "typical" PWR. It cannot be used with the integral option.

NUREG/CR-0200
Volume 3, Section M8
ORNL/NUREG/CSD-2/V3/R2

STANDARD COMPOSITION LIBRARY

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

J. A. Bucholz
J. R. Knight
C. V. Parks
L. M. Petrie
J. C. Turner
R. M. Westfall

Manuscript Completed: February 1983
Date Published: June 1983
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

M8.2 THE STANDARD COMPOSITION LIBRARY

The Standard Composition Library describes the various compounds, alloys, elements and isotopes that can be used to define the material mixtures for a given problem.* Typically, the alphanumeric description of one or more of these materials will be used to define a material mixture as noted in Sects. C1.5.4 and C4.4.4.

When formulating such a mixture, it is often necessary to know the density (g/cc) of the various constituent materials. For convenience, the reference values used by the code have been listed in Table M8.2.1. Note that the given reference values represent the actual theoretical density, except in the case of some individal nuclides where a default value of 1.0 g/cc was used.

The temperature of a given material can be entered, but it is not required (cf. Sects. C1.5.4 or C4.4.4). However, the temperature probably should be entered if resonance data or Bondarenko data is available for the material, or if thermal scattering data is available at more than one temperature. YES/NO flags for both conditions are given in Table M8.2.1. Note that the list of nuclides having resonance data differs from one master cross-section library to another. Table C1.A.2 lists those nuclides having resonance data on each of the master cross-section libraries. The list of nuclides having thermal scattering cross-section data at more than one temperature also differs from one library to another. Table C1.A.3 lists those nuclides (and the associated temperatures) for each of the master cross-section libraries. While the user may find this information interesting and/or helpful, he should have no direct need of it.

Unfortunately, all nuclides are not presently available on all master cross-section libraries. Column 5 of Table M8.2.1 is designed to give a quick indication of which cross section libraries (denoted by the reference numbers 1-8) contain data for the specified nuclide. The reference numbers are matched to available SCALE libraries in Table M8.2.2. The 27GROUPSHLD library is based on ENDF/B-IV data and merely extends the 27GROUPNDF4 library to include several fission product nuclides which are not typically used in criticality analyses.

---

*In addition to the various materials listed here, one is also free to use any of the solutions found in Sect. M8.3.

Multiple sets of iron, nickel and chromium isotopes are in the Standard Composition Library. These sets correspond to different weighting functions used in generating the multigroup cross sections. The special weighting functions correspond to $1/E\sigma_t(E)$ where $\sigma_t(E)$ is the total cross section of either SS-304 stainless steel for use in mixing stainless steels or inconel for use in mixing inconel-like alloys. These weightings specifically apply to the 27- and 218-group library generated from ENDF/B-IV data. In order for the stainless steel and inconel standard composition identifiers to be usable with the Hansen Roach library, the 218-group library sets with the special weighting functions were collapsed to 16 groups and added to the Hansen Roach data set. For the 123-group library, the standard iron, nickel and chromium cross section sets were reproduced with the identifiers of the specially weighted sets. This allows the user to specify stainless steel or inconel when using the 123-group library, but the cross sections used will not have a special weighting.

The nuclide ID numbers are listed in column 6 of Table M8.2.1. Typically, the ID number is A + 1000 * Z where Z and A are the charge and mass numbers for the nuclide (i.e., 1001 for hydrogen and 8016 for oxygen, etc.). Exceptions to this rule include metastable nuclides and nuclides whose cross sections have a special weighting. Also, elements with isotopic mixtures (typically natural abundance) have ID numbers of Z*1000. The nuclides in brackets are those for which the isotopic weight percent can be specified (or else default to the value given in Table M8.4.1).

To more fully document the composition of each material and/or to document the assumptions used in producing the associated cross-section data, a brief description of each material in Table M8.2.1 is included here:

- LITHIUM — Lithium: natural isotopic distribution obtained by default
- BORON — Boron: natural isotopic distribution obtained by default
- BROMINE — Bromine: natural isotopic distribution obtained by default
- SILVER — Silver: natural isotopic distribution obtained by default
- INDIUM — Indium: natural isotopic distribution obtained by default
- TUNGSTEN — Tungsten: natural isotopic distribution obtained by default
- RHENIUM — Rhenium: natural isotopic distribution obtained by default
- B4C — Boron carbide: B_4C; natural isotopic distribution obtained by default
- H2O — Water: cross-sections developed using 1/E weighting everywhere

- H2O-X(E)-HR	Water: cross-sections developed using fission spectrum weighting at high energies and 1/E at lower energies
- D2O	Heavy water: D_2O
- ZIRCALLOY	Zircalloy-2 as tabulated in ENDF/B-IV (~97.91 wt % zirconium, 1.59 wt % tin, 0.5 wt % iron); Note: the thermal capture cross section and the resonance capture integrals of zirc-2 and zirc-4 are equal within the experimental measurement uncertainties and the variance on the contents of the alloying agents used in zirc-2 and zirc-4.
- BALSA	Balsa wood: C H O , 0.125 g/cm
- REDWOOD	Redwood: C H O , 0.4485 g/cm
- OAK	Oak wood: C H O , 0.75 g/cm
- SS304	Stainless steel - 304: 69.5 wt % iron, 19 wt % chromium, 9.5 wt % nickel, 2 wt % manganese
- SS316	Stainless steel - 316: 65.42 wt % iron, 17 wt % chromium, 12 wt % nickel, 2.5 wt % molybdenum, 2 wt % manganese, 1 wt % silicon, 0.08 wt % carbon
- INCONEL	Inconel: 73 wt % nickel, 15 wt % chromium, 7 wt % iron, 2.5 wt % titanium, 2.5 wt % silicon
- CARBONSTEEL	Carbon steel: 99 wt % iron, 1 wt % carbon; cross-sections developed using 1/E weighting
- ORCONCRETE	Oak Ridge Concrete: 41.02 wt % oxygen, 32.13 wt % calcium, 17.52 wt % carbon, 3.448 wt % silicon, 3.261 wt % magnesium, 1.083 wt % aluminum, 0.7784 wt. % iron, 0.6187 wt % hydrogen, 0.1138 wt % potassium, 0.0271 wt % sodium
- RFCONCRETE	Rocky Flats Concrete: 48.49 wt % oxygen, 23 wt % calcium, 15.5 wt % silicon, 5.52 wt % carbon, 2.17 wt % aluminum, 1.37 wt % potassium, 1.25 wt % magnesium, 1.01 wt % iron, 0.75 wt % hydrogen, 0.63 wt % sodium, 0.19 wt % sulfur, 0.1 wt % titanium, 0.02 wt % nitrogen
- MGCONCRETE	Magnuson's Concrete: 45.84 wt % oxygen, 20.77 wt % calcium, 9.674 wt % carbon, 8.666 wt % potassium, 8.644 wt % magnesium, 3.863 wt % silicon, 0.721 wt % aluminum, 0.5134 wt % iron, 0.4146 wt % zinc, 0.3046 wt % hydrogen, 0.2285 wt % sulfur, 0.1366 wt % titanium, 0.1294 wt % sodium, 0.048 wt % chlorine, 0.0469 wt % manganese
- REG-CONCRETE	Regular Concrete: 53.2 wt % oxygen, 33.7 wt % silicon, 4.4 wt % calcium, 3.4 wt % aluminum, 2.9 wt % sodium, 1.4 wt % iron, 1.0 wt % hydrogen
- PYREX	Pyrex: 53.5 wt % oxygen, 37.7 wt % silicon, 4.1 wt % sodium, 3.7 wt % boron, 1.0 wt % aluminum
- PLEXIGLASS	Plexiglas: $C_5H_8O_2$, 1.18 g/cc
- POLYETHYLENE	Polyethylene: C_2H_4, 0.92 g/cc
- PARAFFIN	Paraffin: $C_{25}H_{52}$, 0.93 g/cc
- HNO3	Nitric acid: HNO_3

- HFACID	Hydrafluoric acid: HF
- UO2	Uranium dioxide: UO_2
- UO3	Uranium trioxide: UO_3
- U3O8	Uranium oxide (yellowcake): U_3O_8
- UC	Uranium carbide: UC
- UN	Uranium nitride: UN
- UF4	Uranium tetrafluoride: UF_4
- UF6	Uranium hexafluoride: UF_6
- UO2F2	Uranyl fluoride: UO_2F_2
- UO2(NO3)2	Uranyl nitrate: $UO_2(NO_3)_2$
- URANIUM	Uranium metal (19.05 g/cc) having a variable isotope distribution - i.e., the user may specify the actual isotope distribution or use the natural isotope distribution supplied by default
- U(.27)METAL	Depleted uranium metal (19.05 g/cc) having a fixed isotope distribution: 0.27 wt % ^{235}U, 99.73 wt % ^{238}U; to specify a different distribution, the user should use URANIUM instead of U(.27)METAL
- PUO2	Plutonium oxide: PuO_2
- PUC	Plutonium carbide: PuC
- PUN	Plutonium nitride: PuN
- PUF4	Plutonium tetrafluoride: PuF_4
- PU(NO3)4	Plutonium nitrate: $Pu(NO_3)_4$
- PLUTONIUMALP	Plutonium metal - α phase; has density of 19.84 g/cc but is otherwise the same as the Δ phase
- PLUTONIUMDLT	Plutonium metal - Δ phase; has density of 15.92 g/cc but is otherwise the same as the α phase
- FISP-1	Short-lived lumped fission products.
- FISP-2	Long-lived lumped fission products.
- 1/VABSORBER	Fictitious material having a 1/v absorption cross section normalized to 1.0 at 0.0253 eV
- H	Hydrogen, cross-sections developed using 1/E weighting everywhere
- H-X(E)-HR	Hydrogen, cross-sections developed using fission spectrum weighting at high energies and 1/E weighting at lower energies
- H-POLY	Hydrogen, cross-sections developed using a thermal scattering kernel for hydrogen bound polyethylene
- D	Deuterium
- H-3	Hydrogen-3
- HE-3	Helium-3
- HE	Helium: He
- LI-6	Lithium-6
- LI-7	Lithium-7
- BE	Beryllium, cross-sections developed assuming a free atom
- BEBOUND	Beryllium, cross-sections developed assuming Be atom bound in a crystalline lattice
- B-10	Boron-10
- B-11	Boron-11
- C	Carbon: C
- C-GRAPHITE	Carbon, cross-sections developed using a thermal scattering kernel for graphite
- N	Nitrogen: N

- N-15	Nitrogen-15
- O	Oxygen
- O-17	Oxygen-17
- F	Flucrine: F
- NA	Sodium: Na
- MG	Magnesium: Mg
- AL	Aluminum: Al
- SI-NAT	Silicon
- SI	Silicon: Si
- P	Phosphorous: P
- S	Sulfur
- CL	Chlorine: Cl
- K-NAT	Potassium
- K	Potassium
- CA-NAT	Calcium
- CA	Calcium: Ca
- SC-45	Scandium-45
- TI	Titanium: Ti
- V-NAT	Vanadium
- V	Vanadium
- CR	Chromium, cross-sections developed using 1/E weighting
- CRSS	Chromium, cross-sections developed using $[E\sigma_t(E)]^{-1}$ weighting where $\sigma_t(E)$ is that of SS-304
- CRINCONEL	Chromium, cross-sections developed using $[E\sigma_t(E)]^{-1}$ weighting where $\sigma_t(E)$ is that of inconel
- MN	Manganese, cross-sections developed using 1/E weighting
- MNSS	Manganese, cross-sections developed using $[E\sigma_t(E)]^{-1}$ weighting where $\sigma_t(E)$ is that of SS-304
- FE	Iron, cross-sections developed using 1/E weighting
- FESS	Iron, cross-sections developed using $[E\sigma_t(E)]^{-1}$ weighting where $\sigma_t(E)$ is that of SS-304
- FEINCONEL	Iron, cross-sections developed using $[E\sigma_t(E)]^{-1}$ weighting where $\sigma_t(E)$ is that of inconel
- CO , CO-59	Cobalt-59
- NI	Nickel, cross-sections developed using 1/E weighting
- NISS	Nickel, cross-sections developed using $[E\sigma_t(E)]^{-1}$ weighting where $\sigma_t(E)$ is that of SS-304
- NIINCONEL	Nickel, cross-sections developed using $[E\sigma_t(E)]^{-1}$ weighting where $\sigma_t(E)$ is that of inconel
- CU	Copper: Cu
- ZN	Zinc: Zn
- ZN-64	Zinc-64
- GA-NAT	Gallium
- GE-72	Germanium-72
- GE-73	Germanium-73
- GE-74	Germanium-74
- GE-76	Germanium-76
- AS-75	Arsenic-75
- SE-76	Selenium-76
- SE-77	Selenium-77

- SE-78	Selenium-78
- SE-80	Selenium-80
- SE-82	Selenium-82
- BR-79	Bromine-79
- BR-81	Bromine-81
- KR-78	Krypton-78
- KR-80	Krypton-80
- KR-82	Krypton-82
- KR-83	Krypton-83
- KR-84	Krypton-84
_ KR-85	Krypton-85
_ KR-86	Krypton-86
_ RB-85	Rubidium-85
_ RB-86	Rubidium-86
_ RB-87	Rubidium-87
_ SR-86	Strontium-86
_ SR-87	Strontium-87
_ SR-88	Strontium-88
_ SR-89	Strontium-89
_ SR-90	Strontium-90
_ Y-89	Yttrium-89
_ Y-90	Yttrium-90
_ Y-91	Yttrium-91
_ ZR	Zirconium: Zr
_ ZR-90	Zirconium-90
_ ZR-91	Zirconium-91
_ ZR-92	Zirconium-92
_ ZR-93	Zirconium-93
_ ZR-94	Zirconium-94
_ ZR-95	Zirconium-95
_ ZR-96	Zirconium-96
_ NB	Niobium: Nb
_ NB-93	Niobium-93
_ NB-94	Niobium-94
_ NB-95	Niobium-95
_ MO	Molybdenum: Mo
_ MO-94	Molybdenum-94
_ MO-95	Molybdenum-95
_ MO-96	Molybdenum-96
_ MO-97	Molybdenum-97
_ MO-98	Molybdenum-98
_ MO-99	Molybdenum-99
_ MO-100	Molybdenum-100
_ TC-99	Technetium-99
_ RU-99	Ruthenium-99
_ RU-100	Ruthenium-100
_ RU-101	Ruthenium-101
_ RU-102	Ruthenium-102
_ RU-103	Ruthenium-103
_ RU-104	Ruthenium-104
_ RU-105	Ruthenium-105
_ RU-106	Ruthenium-106
_ RH-103	Rhodium-103

- RH-105	Rhodium-105
- PD-104	Palladium-104
- PD-105	Palladium-105
- PD-106	Palladium-106
- PD-107	Palladium-107
- PD-108	Palladium-108
- PD-110	Palladium-110
- AG-107	Silver-107
- AG-109	Silver-109
- AG-111	Silver-111
- CD	Cadmium
- CD-108	Cadmium-108
- CD-110	Cadmium-110
- CD-111	Cadmium-111
- CD-112	Cadmium-112
- CD-113	Cadmium-113
- CD-114	Cadmium-114
- CD-115M	Cadmium-115 (metastable)
- CD-116	Cadmium-116
- IN-113	Indium-113
- IN-115	Indium-115
- SN	Tin: Sn
- SN-115	Tin-115
- SN-116	Tin-116
- SN-117	Tin-117
- SN-118	Tin-118
- SN-119	Tin-119
- SN-120	Tin-120
- SN-122	Tin-122
- SN-123	Tin-123
- SN-124	Tin-124
- SN-125	Tin-125
- SN-126	Tin-126
- SB-121	Antimony-121
- SB-123	Antimony-123
- SB-124	Antimony-124
- SB-125	Antimony-125
- SB-126	Antimony-126
- TE-122	Tellurium-122
- TE-123	Tellurium-123
- TE-124	Tellurium-124
- TE-125	Tellurium-125
- TE-126	Tellurium-126
- TE-127M	Tellurium-127 (metastable)
- TE-128	Tellurium-128
- TE-129M	Tellurium-129 (metastable)
- TE-130	Tellurium-130
- TE-132	Tellurium-132
- I-127	Iodine-127
- I-129	Iodine-129
- I-130	Iodine-130
- I-131	Iodine-131
- I-135	Iodine-135
- XE-124	Xenon-124

- XE-126 Xenon-126
- XE-128 Xenon-128
- XE-129 Xenon-129
- XE-130 Xenon-130
- XE-131 Xenon-131
- XE-132 Xenon-132
- XE-133 Xenon-133
- XE-134 Xenon-134
- XE-135 Xenon-135
- XE-136 Xenon-136
- CS-133 Cesium-133
- CS-134 Cesium-134
- CS-135 Cesium-135
- CS-136 Cesium-136
- CS-137 Cesium-137
- BA-134 Barium-134
- BA-135 Barium-135
- BA-136 Barium-136
- BA-137 Barium-137
- BA-138 Barium-138
- BA-140 Barium-140
- LA-139 Lanthanum-139
- LA-140 Lanthanum-140
- CE-140 Cerium-140
- CE-141 Cerium-141
- CE-142 Cerium-142
- CE-143 Cerium-143
- CE-144 Cerium-144
- PR-141 Praseodymium-141
- PR-142 Praseodymium-142
- PR-143 Praseodymium-143
- ND-142 Neodymium-142
- ND-143 Neodymium-143
- ND-144 Neodymium-144
- ND-145 Neodymium-145
- ND-146 Neodymium-146
- ND-147 Neodymium-147
- ND-148 Neodymium-148
- ND-150 Neodymium-150
- PM-147 Promethium-147
- PM-148 Promethium-148
- PM-148M Promethium-148 (metastable)
- PM-149 Promethium-149
- PM-151 Promethium-151
- SM-147 Samarium-147
- SM-148 Samarium-148
- SM-149 Samarium-149
- SM-150 Samarium-150
- SM-151 Samarium-151
- SM-152 Samarium-152
- SM-153 Samarium-153
- SM-154 Samarium-154
- EU-151 Europium-151
- EU-152 Europium-152
- EU-153 Europium-153
- EU-154 Europium-154

- EU-155	Europium-155
- EU-156	Europium-156
- EU-157	Europium-157
- GD	Gadolinium: Gd
- GD-152	Gadolinium-152
- GD-154	Gadolinium-154
- GD-155	Gadolinium-155
- GD-156	Gadolinium-156
- GD-157	Gadolinium-157
- GD-158	Gadolinium-158
- GD-160	Gadolinium-160
- TB-159	Terbium-159
- TB-160	Terbium-160
- DY-160	Dysprosium-160
- DY-161	Dysprosium-161
- DY-162	Dysprosium-162
- DY-163	Dysprosium-163
- DY-164	Dysprosium-164
- HO-165	Holmium-165
- ER-166	Erbium-166
- ER-167	Erbium-167
- LU-175	Lutetium-175
- LU-176	Lutetium-176
- HF	Hafnium: Hf
- HF-174	Hafnium-174
- HF-176	Hafnium-176
- HF-177	Hafnium-177
- HF-178	Hafnium-178
- HF-179	Hafnium-179
- HF-180	Hafnium-180
- TA-181	Tantalum-181
- TA-182	Tantalum-182
- W-182	Tungsten-182
- W-183	Tungsten-183
- W-184	Tungsten-184
- W-186	Tungsten-186
- RE-185	Rhenium-185
- RE-187	Rhenium-187
- IR-191	Iridium-191
- IR-193	Iridium-193
- AU	Gold: Au
- PB	Lead: Pb
- TH-232	Thorium-232
- PA-233	Protactinium-233
- U-233	Uranium-233
- U-234	Uranium-234
- U-235	Uranium-235
- U-236	Uranium-236
- U-238	Uranium-238
- NP-237	Neptunium-237
- PU-238	Plutonium-238

Revised 12/84

- PU-239	Plutonium-239
- PU-240	Plutonium-240
- PU-241	Plutonium-241
- PU-242	Plutonium-242
- AM-241	Americium-241
- AM-242M	Americium-242 (metastable)
- AM-243	Americium-243
- CM-243	Curium-243
- CM-244	Curium-244
- CM-245	Curium-245
- CM-246	Curium-246
- ACTIVITIES	Available activity responses (see Table S3.3.4)

Table M8.2.1 THE STANDARD COMPOSITION LIBRARY

ALPHANUMERIC DESCRIPTION STANDARD COMPOSITION	THEORETICAL DENSITY th IN GRAMS/CC	RESONANCE DATA OR BONDARENKO DATA AVAILABLE?	SCATTERING X-SECT DATA AVAILABLE AT MULTIPLE TEMPERATURES?	X-SECT LIBRARIES FOR THIS STD. COMPOSITION (SEE TABLE M8.2.2)	LIST OF NUCLIDES IN THIS STD. COMPOSITION (WITH ISOTOPES THE USER MAY SPECIFY SHOWN IN BRACKETS)
LITHIUM	0.5340	NO	YES	1 2 3 4 5 6 7 8	[3006 3007]
BORON	2.5350	NO	YES	1 2 3 4 5 6 7 8	[5010 5011]
BROMINE	3.1200	YES	YES	- 2 - 4 5 - 7 8	[35079 35081]
SILVER	10.5000	YES	YES	1 2 - 4 5 - 7 8	[47107 47109]
INDIUM	7.3100	YES	YES	1 2 - 4 5 - 7 8	[49113 49115]
TUNGSTEN	19.3000	YES	YES	1 2 - 4 5 - 7 8	[74182 74183 74184 74186]
RHENIUM	21.0200	YES	YES	1 2 - 4 5 - 7 8	[75185 75187]
B4C	2.5200	NO	YES	1 2 3 4 5 6 7 8	[5010 5011] 6012
H2O	0.9982	NO	YES	1 2 3 4 5 6 7 8	1001 8016
H2O-X(E)-HR	0.9982	NO	YES	1 - - - - - - -	1201 8016
D2O	1.1053	NO	YES	1 2 3 4 5 - 7 8	1002 8016
ZIRCALLOY	6.4400	YES	NO	1 2 3 4 5 - 7 8	40302
SS304	7.9200	YES	NO	1 2 3 4 5 - 7 8	24304 25055 26304 28304
SS316	7.7500	YES	YES	1 2 3 4 5 - 7 8	6012 14028 24304 25055 26304 28304 42000
INCONEL	8.3000	NO	NO	1 2 3 4 5 - 7 8	14028 22000 24404 26404 28404
CARBONSTEEL	7.8212	YES	YES	1 2 3 4 5 6 7 8	6012 26000

Table M8.2.1 (Continued)

ALPHANUMERIC DESCRIPTION STANDARD COMPOSITION	THEORETICAL DENSITY th IN GRAMS/CC	RESONANCE DATA OR BONDARENKO DATA AVAILABLE?	SCATTERING X-SECT DATA AVAILABLE AT MULTIPLE TEMPERATURES?	X-SECT LIBRARIES FOR THIS STD. COMPOSITION (SEE TABLE M8.2.2)	LIST OF NUCLIDES IN THIS STD. COMPOSITION (WITH ISOTOPES THE USER MAY SPECIFY SHOWN IN BRACKETS)
ORCONCRETE	2.2994	YES	YES	1 2 3 4 5 6 7 8	1001 6012 8016 11023 12000 13027 14028 19039 20040 26000
RFCONCRETE	2.3210	YES	YES	1 2 3 4 5 - 7 8	1001 6012 7014 8016 11023 12000 13027 14028 16032 19039 20040 22000 26000
MGCONCRETE	2.3298	YES	YES	1 - - - - - 7 -	1001 6012 8016 11023 12000 13027 14028 16032 17000 19039 20040 22000 25055 26000 30000
REG-CONCRETE	2.3000	YES	YES	1 2 3 4 5 6 7 8	1001 8016 11023 13027 14028 20040 26000
BALSA	0.1250	NO	YES	1 2 3 4 5 6 7 8	1001 6012 8016
REDWOOD	0.4485	NO	YES	1 2 3 4 5 6 7 8	1001 6012 8016
OAK	0.7500	NO	YES	1 2 3 4 5 6 7 8	1001 6012 8016
PYREX	2.2300	NO	YES	1 2 3 4 5 6 7 8	[5010 5011] 8016 11023 13027 14028
PLEXIGLASS	1.1800	NO	YES	1 2 3 4 5 6 7 8	1001 6012 8016
POLYETHYLENE	0.9200	NO	YES	1 2 3 4 5 6 7 8	1001 6012
PARAFFIN	0.9300	NO	YES	1 2 3 4 5 6 7 8	1001 6012
HNO3	1.0000	NO	YES	1 2 3 4 5 6 7 8	1001 7014 8016

Table M8.2.1 (Continued)

ALPHANUMERIC DESCRIPTION STANDARD COMPOSITION	THEORETICAL DENSITY th IN GRAMS/CC	RESONANCE DATA OR BONDARENKO DATA AVAILABLE?	SCATTERING X-SECT DATA AVAILABLE AT MULTIPLE TEMPERATURES?	X-SECT LIBRARIES FOR THIS STD. COMPOSITION (SEE TABLE M8.2.2)	LIST OF NUCLIDES IN THIS STD. COMPOSITION (WITH ISOTOPES THE USER MAY SPECIFY SHOWN IN BRACKETS)
HFACID	1.0000	NO	YES	1 2 3 4 5 6 7 8	1001 9019
UO2	10.9600	YES	YES	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 8016
UO3	7.2900	YES	YES	1 2 3 4 5 6 7 8	[92233 92234 92235 92236 92238] 8016
U3O8	8.3000	YES	YES	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 8016
UC	13.6300	YES	YES	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 6012
UN	14.3100	YES	NO	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 7014
UF4	6.7000	YES	YES	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 9019
UF6	4.8500	YES	YES	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 9019
UO2F2	6.3700	YES	YES	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 8016 9019
UO2(NO3)2	2.8030	YES	YES	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238] 7014 8016
URANIUM	19.0500	YES	NO	1 2 3 4 5 - 7 8	[92233 92234 92235 92236 92238]

Table M8.2.1 (Continued)

ALPHANUMERIC DESCRIPTION STANDARD COMPOSITION	THEORETICAL DENSITY ρth IN GRAMS/CC	RESONANCE DATA OR BONDARENKO DATA AVAILABLE?	SCATTERING X-SECT DATA AVAILABLE AT MULTIPLE TEMPERATURES?	X-SECT LIBRARIES FOR THIS STD. COMPOSITION (SEE TABLE M8.2.2)	LIST OF NUCLIDES IN THIS STD. COMPOSITION (WITH ISOTOPES THE USER MAY SPECIFY SHOWN IN BRACKETS)
U(.27)METAL	19.0500	YES	NO	1 2 3 4 5 6 7 8	92235 92238
PUO2	11.4600	YES	YES	1 2 3 4 5 - 7 8	[94238 94239 94240 94241 94242] 8016
PUC	13.6000	YES	YES	1 2 3 4 5 - 7 8	[94238 94239 94240 94241 94242] 6012
PUN	14.2500	YES	NO	1 2 3 4 5 - 7 8	[94238 94239 94240 94241 94242] 7014
PUF4	7.0000	YES	YES	1 2 3 4 5 - 7 8	[94238 94239 94240 94241 94242] 9019
PU(NO3)4	2.4470	YES	YES	1 2 3 4 5 - 7 8	[94238 94239 94240 94241 94242] 7014 8016
PLUTONIUMALP	19.8400	YES	NO	1 2 3 4 5 - 7 8	[94238 94239 94240 94241 94242]
PLUTONIUMDLT	15.9200	YES	NO	1 2 3 4 5 - 7 8	[94238 94239 94240 94241 94242]
FISP-1	1.0000	NO	NO	- - - - - - - -	901
FISP-2	1.0000	NO	NO	- - - - - - - -	902
1/VASORBER	1.0000	NO	NO	1 2 3 4 5 - - 8	999

M8.4 ISOTOPE DISTRIBUTION TABLE

Those materials in the Standard Composition Library containing multiple isotopes of a single element are denoted in Table M8.2.1. For materials containing lithium, boron, bromine, silver, indium, tungsten, rhenium, uranium, or plutonium, the user is free to specify the isotopic distribution using items 6a and 6b of the Standard Composition Specification Card discussed in Sections C1.5 and C4.4.4. Alternately, the user may elect not to enter this data, thereby telling the code to assume the default values shown in Table M8.4.1. In all cases except plutonium, Table M8.4.1 lists the naturally occurring abundance of each isotope. In describing an arbitrary material, a nuclide ID of Z * 1000 (where Z is the nuclide charge number) can be used to denote the elements of Table M8.4.1. Then, if an isotope distribution other than the default values of Table M8.4.1 is desired, items 6a and 6b of the Standard Composition Specification Card (Sects. C1.5.4 and C4.4.4) must be included.

Table M8.4.1. Isotope Distribution Table.

Material	I.D.	wt%	Material	I.D.	wt%
Lithium	3006	6.4996	Uranium	92233	0.0000
	3007	93.5004		92234	0.0056
				92235	0.7050
Boron	5010	18.3026		92236	0.0000
	5011	81.6974		92238	99.2894
Bromine	35079	50.0650	Plutonium	94238	0.0
	35081	49.9350		94239	100.0
				94240	0.0
Silver	47107	51.3771		94241	0.0
	47109	48.6229		94242	0.0
Indium	49113	4.2283			
	49115	95.7717			
Tungsten	74182	26.0571			
	74183	14.2496			
	74184	30.7262			
	74186	28.9671			
Rhenium	75185	37.1482			
	75187	62.8518			

NUREG/CR-0200
Volume 3, Section M9
ORNL/NUREG/CSD-2/V3/R2

COMPUTING AND TELECOMMUNICATIONS DIVISION
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge. Tennessee 37830

MARS: A MULTIPLE ARRAY SYSTEM USING COMBINATORIAL GEOMETRY

J. T. West*
M. B. Emmett

Manuscript Completed: December 1980
Date Published: October 1981
Date Revised: December 1984

Prepared for the
Office of Nuclear Material Safety and Safeguards
U. S. Nuclear Regulatory Commission
Washington. DC 20555
Under Interagency Agreement DOE 40-549-75
NRC FIN No. B-0009

Martin Marietta Energy Systems, Inc.
operating the
Oak Ridge National Laboratory Oak Ridge Y-12 Plant
Oak Ridge Gaseous Diffusion Plant Paducah Gaseous Diffusion Plant
under Contract No. DE-AC05-840R21400
for the
DEPARTMENT OF ENERGY

*Formerly with the Computer Sciences Division

TABLE OF CONTENTS

		Page
LIST OF TABLES		M9.v
LIST OF FIGURES		M9.vii
ACKNOWLEDGMENTS		M9.ix
ABSTRACT		M9.xi
M9.1	INTRODUCTION	M9.1.1
M9.2	COMBINATORIAL GEOMETRY THEORY AND TRACKING LOGIC	M9.2.1
M9.2.1	BODY DESCRIPTIONS	M9.2.1
M9.2.1.1	Description of Body Types	M9.2.2
M9.2.2	COMBINATORIAL OPERATORS	M9.2.8
M9.2.3	COMBINATORIAL ZONE CONSTRUCTION	M9.2.13
M9.2.4	COMBINATORIAL TRACKING LOGIC	M9.2.14
M9.3	MARS GEOMETRY THEORY	M9.3.1
M9.3.1	THE UNIVERSE BUILDING BLOCK	M9.3.1
M9.3.2	MARS "ARRAY OF ARRAYS" CAPABILITY	M9.3.2
M9.3.3	ARRAY ROTATION	M9.3.4
M9.3.4	MODEL CONSTRUCTION WITH MARS	M9.3.5
M9.4	ARRAY ANALYSIS - COLLISION EDITS	M9.4.1
M9.4.1	ANALYSIS CRITERIA	M9.4.1
M9.4.1.1	Location Criteria	M9.4.1
M9.4.1.2	Media Criteria	M9.4.2
M9.4.1.3	Response Criteria	M9.4.2
M9.4.2	ILLUSTRATIVE EXAMPLES	M9.4.3
M9.5	COMBINATORIAL MARS OVERVIEW	M9.5.1
M9.5.1	MEDIA NOTATION	M9.5.1
M9.5.2	TERMS AND DEFINITIONS	M9.5.1
M9.5.3	RULES FOR ARRAY REPETITION	M9.5.3
M9.6	MARS TRACKING LOGIC, SHORTFALLS, AND CORRECTIONS	M9.6.1
M9.6.1	PARTICLE NESTING TABLE FOR TRACKING	M9.6.1
M9.6.2	MARS ARRAY PITFALLS	M9.6.2
M9.6.3	COMBINATORIAL MODELING PITFALLS	M9.6.5
M9.6.4	COMPUTER TRACKING AND MODELING	M9.6.5
APPENDICES:		
M9.A	FREE-FORM COMBINATORIAL MARS INPUT INSTRUCTIONS	M9.A.1
M9.A.1	TITLE CARD	M9.A.1
M9.A.2	OPTIONS CARD	M9.A.1

TABLE OF CONTENTS (Continued)

M9.A.3 BODY DEFINITION CARDS. M9.A.1
M9.A.4 INPUT ZONE DESCRIPTION CARDS M9.A.1
M9.A.5 REGION CARD. M9.A.3
M9.A.6 UNIVERSE CARD. M9.A.3
M9.A.7 MEDIA CARD M9.A.3
M9.A.8 ARRAY SIZE SPECIFICATIONS INPUT. M9.A.3
M9.A.9 ARRAY CONTENT DESCRIPTION. M9.A.4
M9.A.10 UNIVERSE TYPE. M9.A.7

M9.B ILLUSTRATE EXAMPLES OF MARS GEOMETRY USING THE JUNEBUG GRAPHICS CODE. M9.B.1
M9.B.1 AN UNCOMMON ARRAY SHAPE. M9.B.1
M9.B.1.1 Explanations and Discussions M9.B.2
M9.B.2 OVERLAPPING PINS IN A 2 BY 2 ARRAY M9.B.7
M9.B.3 PLEXIGLAS PIPE CROSS EXPERIMENT. M9.B.15
M9.B.4 A TYPICAL PWR SHIPPING CASK. M9.B.25
M9.B.5 THREE MILE ISLAND REACTOR CORE M9.B.41
M9.B.6 INHERENTLY SUBCRITICAL INSERT DESIGN M9.B.63

M9.C COMBINATORIAL GEOMETRY ZONE SUMMARY TABLE DESCRIPTION. M9.C.1

M9.D INTERFACING THE MARS SUBROUTINES MODULE TO OTHER COMPUTER CODES M9.D.1

M9.E MARS DATA STORAGE AND PROGRAM LOGIC. M9.E.1
M9.E.1 MARS SUBROUTINE DESCRIPTIONS M9.E.1

M9.F COMBINATORIAL GEOMETRY TRACKING. M9.F.1
M9.F.1 COMJOM TRACKING THEORY M9.F.1

M9.G DIAGNOSTIC MESSAGES FROM MARS. M9.G.1
M9.G.1 DIAGNOSTIC MESSAGES FROM MARS INPUT MODULE M9.G.2
M9.G.2 DIAGNOSTIC MESSAGES FROM MARS TRACKING EVENTS. M9.G.5

LIST OF TABLES

M9.A.1	Input Required for Each Body Type	M9.A.2
M9.E.1	MARS Subroutine	M9.E.6
M9.E.2	MARS Storage Requirements	M9.E.7
M9.E.3	Layout of Combinatorial Geometry Data in Blank Common.	M9.E.8
M9.E.4	Detailed Layout of the FPD Array in Blank Common.	M9.E.9
M9.E.5	Detailed Layout of the MA Array in Blank Common	M9.E.10
M9.E.6	Definitions of Variables in Common GOMLOC	M9.E.12
M9.E.7	Definitions of Variables in Common PAREM as Found in Combinatorial Geometry	M9.E.13
M9.E.8	Definitions of Variables in Common ORGI	M9.E.15
M9.E.9	Definitions of Variables in Common TAPE	M9.E.16
M9.E.10	Definitions of Variables in Common MGOMV.	M9.E.17
M9.E.11	Definitions of Variables in Common ARAR	M9.E.18
M9.E.12	Definitions of Variables in Common ARK.	M9.E.19

The gap illustrated may be small, less than 1.0E-09 cm, but it would be sufficient to cause a particle track to be lost. The correction is illustrated as the "overlap fix." If the overlap is greater than 1.0E-09 cm, then Subroutine FINEFI will fatally warn the user, "THE ARRAY ... IN BODY ... REFERENCED IN CODE ZONE ... DOES NOT FIT." Therefore, the overlap applied should be small or it may itself be a fatal error. This is rigid requirement. The overlap fix is beneficial in exiting the array correctly as well as entering the array correctly.

M9.6.3 COMBINATORIAL MODELING PITFALLS

Now that the user has some appreciation for the pitfalls in array tracking, it is necessary to discuss pitfalls in combinatorial geometry tracking. Whenever two floating point numbers resulting from separate calculations are compared on a computer of finite precision, it is bad practice to test and depend on direct equality. In combinatorial tracking it is not always possible to avoid this problem. User input description is the best place to avoid ambiguity on zone boundaries. When two bodies share a common surface, the user should never assume the computer is capable of recognizing the two surfaces are actually a single surface. Furthermore, try to avoid having two bodies sharing a common surface occur on an input zone boundary, especially if the two bodies are referenced in both the adjacent input zone descriptions. In your geometry model, when constructing code zones, always attempt to describe your geometry with combinatorial logic, the union or intersection of bodies, without falling into the dangerous trap of defining bodies with common surfaces. When this is not possible, a less dangerous alternative exists. A small overlap between two body surfaces is preferable to a single surface. Combinatorial geometry will unwittingly let the user double-define space. Combinatorial tracking is always happy with the first definition for a portion of space it finds. It does not check to see if any other definitions exist. The danger comes in trying to leave the doubly defined space and in depending on which definition of the space is used.

M9.6.4 COMPUTER TRACKING AND MODELING

This section deals with the combinatorial MARS tracking logic and considerations involved in geometry model development. It is essential at this point that the user understand the following concepts and definitions before continuing:

A. How combinatorial logic determines in which code zone a particle is located,

B. The meaning of the following combinatorial variables: RIN, ROUT, DIST, DISTO (see Table M9.E.6 for label common /PAREM/),

C. The different roles of combinatorial geometry tracking and MARS array tracking (see Sect. M9.6.3). The complex modeling capabilities of combinatorial MARS are accompanied with a complex debugging output dump requiring user understanding of the tracking methods.

The most important questions combinatorial geometry and MARS must answer during tracking are:

MARS Tracking Questions:

A. In which universe is the particle located?

B. Will the particle exit the current universe? If so, then what will be the universe entered and in what array?

Combinatorial Tracking Questions:

A. In which input zone in the current universe is the particle located?

B. Will the particle exit the current input zone? If so, then what will be the next input zone the particle will enter?

These questions for MARS and combinatorial geometry are related in the following way. The first question in each system is asking the basic question:

Where is the particle located?

Each system, MARS and combinatorial, is answering the question from its own perspective. The second question in each system is asking the basic question:

Where is the particle going?

This question cannot be answered without knowing the answer to the first question in both the combinatorial and MARS systems. In changing lattice locations in an array, the dialogue between combinatorial tracking and MARS tracking may go:

Combinatorial: The particle is entering Media -1000; therefore, the particle is exiting the current universe.

MARS: The lattice cell being entered is in the same array as the current lattice cell location; therefore, the new universe being entered is NLU. The new coordinates are computed as X, Y, Z for the new universe.

Combinatorial: The particle is in input zone IR in the current universe. The particle will be entering input zone IRPRIM in the new universe.

M9.E MARS DATA STORAGE AND PROGRAM LOGIC

Subroutine CALI answers the question: What is the particle location? It is given particle coordinates, usually in the absolute universe, although this is not a requirement. It proceeds to determine if the particle is in an array or in a valid material media. If the particle is in a valid media, it determines this and returns. If the particle is in an array, it then checks the next level of the geometry as described in the array description. It repeats the process until it determines a valid positive material media the particle coordinates are inside. In this process, it starts from a lower, more global level of the geometry hierarchy, and checks from there to a higher, more local geometry level. The process is completed when the particle is located in a positive real media.

During the process of checking successive levels of a geometry model, the nesting table is always updated as the particle coordinates are translated. Therefore, as MARS goes from a lower global level to a higher, local geometry level, MARS always maintains the knowledge of where the particle is located relative to the lower, more global geometry levels.

Subroutine PILOT goes through the same process during tracking, except it has the logic to translate from higher levels to lower levels as well as from lower levels to higher levels. This is necessary for tracking. PILOT uses the nesting table generated by CALI when the particle initial location was determined. Both PILOT and CALI call Subroutine STORE to update the nesting table.

M9.E.1 MARS SUBROUTINE DESCRIPTIONS

ABEND - ABEND is called for an abnormal termination of a calculation. It gives a traceback of where it was called and calls EXIT.

ABOX - ABOX is called from JOMIN1. It reads the BPP and WPP body data and converts the data to the BOX and WED body input format. The BPP and WPP are alternative ways of describing the BOX and WED.

ALBERT - ALBERT is called from GENI to construct the floating point data to be stored in the FPD table for the ARB body, the arbitrary polyhedron body.

ARGEN - ARGEN is called from AZIP. It generates the size of each array from the array content tables and NLV table. It must compute the size of the simplest arrays first. These are the arrays that contain only universes which do not contain subarrays. ARGEN comes from the term "array generator." The x-width, y-length, and z-height of each array is stored in the WLH array. This array is not needed in tracking or particle locating. It is needed to verify the fit of each array in each reference location.

AZIP - AZIP reads the combinatorial MARS array size and content input. It calls several other subroutines and generates a series of tables from the input data that is needed for MARS array tracking. It is called after JOMIN2 and GENI have set up the combinatorial data tables in memory. It uses much of the combinatorial input data from memory to construct the MARS data tables.

BOD - BOD is called from CELL after the lattice cell position of a particle has been determined. It returns the cell contents. If the contents of a cell is vacant, it sets a flag. This is legal for exiting an array but fatal for entering an array. A particle cannot enter a vacant lattice position of an array.

CALI - CALI is called from outside the MARS subroutine module to determine a particle location. It first determines the universe and array the particle is inside. Frequently this requires checking several levels of the geometry model. CALI calls LOOKZ to determine the combinatorial geometry input zone a particle is inside for each universe or array. When CALI has checked several levels of geometry logic and determines the positive media the particle is inside, it then returns the information in labeled common PAREM, with the nesting table set by calls to STORA.

CELL - CELL is called from both CALI and PILOT. It is the guts of entering or exiting an array. It is called for a particle inside of or moving through an array. It returns the lattice cell position and content for the current particle position. CELL serves a dual role. It says where a particle is in an array when the particle is not moving, and it says the next cell the particle will enter when the particle is moving. CELL calls subroutines SORT and BOD.

CLEV - CLEV is called from PILOT to translate a particle coordinate from a larger level, more local coordinate system to a lower, more global geometry level number given the nesting table. Because of the presence of several labeled commons in CLEV, it changes quite a few variables in a few commons. The user should be careful before calling CLEV so as not to inadvertently change some important information needed for tracking.

CORNER - CORNER is called from TRENTE and RTEXIT. It returns the origin or vertex for a reference body. This information is essential to particle translation for both entering and exiting a universe or an array.

CTRAN - CTRAN is called to determine the particle coordinate relative to either a lattice cell in an array or given the lattice position and the particle coordinate, it can determine the coordinate of the particle relative to the origin of the array. It requires the array number, the lattice position, and the particle coordinate to perform the proper translation. The argument NX determines the proper translation to perform.

DELTA - DELTA is called from ARGEN. It computes the size of each dimension of each lattice cell from the reference body of the cell. This dimension is summed in ARGEN to compute the total array size on each dimension.

DIPR - DIPR is called from several subroutines to dump MARS array information after an error has been detected. It calls DUPR. DIPR is called to simplify the call to DUPR from the error detecting coding.

DUPR - DUPR is called from DIPR. It dumps the MARS array tables from memory and the nesting table of the current particle location. It generates quite a bit of output and should not be called frequently during debugging. The argument IXD in DUPR can suppress much of the geometry dump to allow frequent printouts of the nesting array during model debugging.

FINEFI - FINEFI is called from AZIP. It checks the MARS array input with the combinatorial input for snug fit. Most of the MARS input error messages regarding array content, reference, or fit comes from FINEFI. It is essential to have exact geometry fits before successful tracking can be initiated.

GENI - GENI is called from JOMIN2. It reads the binary file generated by JOMIN1 and stores the data into memory in the combinatorial geometry array tables. It edits and prints the output edit of the combinatorial input.

GG - GG is called from LOOKZ and G1. It is the guts and essence of the combinatorial geometry system. It computes, given the body number, the RIN and ROUT for the body.

GTVLIN - GTVLIN is called from JOMIN2 to calculate input zone volumes. The present version of GTVLIN is next to useless and is normally never used.

G1 - G1 is called from PILOT to determine the next input zone a particle may enter. It is in three parts. The first part of G1 determines the distance to exit the present zone if the zone will be exited. The second part of G1 determines the next zone of entry by checking those previous zones the particles entered when exiting the current zone by way of the body causing the particle to exit. If a new zone of entry is not determined from this table, then the third part of G1 proceeds to check all other possible zones of entry in the current universe. Errors in combinatorial geometry tracking come from G1.

IPACK - IPACK is called from JOMIN1 to pack several integers into one integer.

JOMIN1 - JOMIN1 reads the free-form combinatorial input for the body data, zone description data, region data, universe data, and media data. The input is written on a scratch file for later use by GENI.

JOMIN2 - JOMIN2 is called during combinatorial geometry input processing. It sets up the combinatorial geometry array pointers from information determined by JOMIN1. It then calls GENI, and the input data is stored in the proper arrays. JOMIN2 can call GTVLIN but usually does not.

LEVEL - LEVEL is called from AZIP. It determines the highest level number each array is referenced inside. This data is placed in the NLV table for later use in AZIP. The NLV table is not used for tracking or locating a particle.

LOOKZ - LOOKZ is called from CALI and PILOT to determine the particle location in a universe. It calls GG to determine RIN and ROUT for each body in a zone description table. It then determines the code and input zone the particle is inside. This information is returned in labeled common PAREM.

ORTHOM - ORTHOM is called from AZIP. It computes the rotational matrices of each BOX body, which references either an array or a universe. A BOX can be arbitrarily positioned in space. When a particle enters a universe or an array and when the reference body is a BOX, then both the particle coordinates and direction cosines must be rotated. Subroutine ROTA is the only routine which makes use of these matrices constructed by ORTHOM.

PILOT - PILOT is called from outside the MARS subroutine module to determine the next input zone a particle may enter. The first zone the particle enters with a positive media number is returned. PILOT cannot be called without first knowing the particle location as determined by calling CALI.

PR - PR is called from combinatorial geometry errors discovered in LOOKZ or G1. It generates even more output.

RESTOR - RESTOR is called by JOMIN when RINPU2 calls JOMIN. When executed in SCALE, the geometry and array data are written out on a binary file for later recovery. RESTOR reads labelled commons REPEAT, ARAR, GOMLOC, and PAREM. It also reads the MA, FPD arrays and all of the MARS array storage from the binary file. The file read by RESTOR is written by RINPUP in the SAS3 control module in SCALE or by RINPU1 in MORSE-SGC.

RISK - RISK translates coodinates defined on geometry level NL1 to geometry level NL2 using the nesting table stored in array LP. This translation is from a global coordinate to a local coordinate system. NL2 must be greater than NL1.

ROTA - ROTA performs particle and direction cosine rotation upon entering or exiting a coordinate system defined by a BOX reference body. It is called from TRENTE and RTEXIT. It uses the rotational matrices constructed by ORTHOM during input processing.

RTEXIT - RTEXIT is called on exiting either a universe or an array. The "RT" means rotate - then translate. The rotation must be performed first and then the translation can occur. RTEXIT is given the reference body for the universe or the array. If the body is an RPP, only translation of coordinates is required. If the body is a BOX, then both rotation and translation are required.

SAZAR - SAZAR is called by JOMIN if NDSN, the number of array analysis collision edits (input in MORSE-SGC) required, is >0. SAZAR reads the location, media, and response criteria for each edit. This allows the user to selectively sample specific collisions.

SKPBLK - SKPBLK is called from JOMIN1 to skip blanks while reading free-form input cards.

SORCER - SORCER is called after completion of a batch of histories. It sorts the next zone of entry table in the order of most probable entry. This is a simple, fast routine used to speed up tracking through a complex geometry model.

SORT - SORT is called from CELL. It is called for each dimension of an array. It determines the cell position on each dimension for the particle location. It determines when a particle is exiting an array lattice and sets a flag for CELL to take appropriate action.

STORA - STORA is called from CALI and PILOT. It updates the nesting table during array tracking and locating. This is one of the most important aspects of array nesting logic. It is essential to the MARS logic.

TRENTE - TRENTE is called on entering either a universe or an array. The "TR" means translate - then rotate. The translation of the particle coordinate must occur and then the rotation. TRENTE is the opposite of RTEXIT. It requires the reference body of the universe or the array to determine the appropriate coordinate transformation.

UNIS - UNIS is called by AZIP to read, verify, and edit the universe type input. If a universe is erroneously declared simple, UNIS changes the universe type specification to combinatorial and writes warnings to the user.

ZEXITS - ZEXITS is called to dump the combinatorial summary of zone transfers at the end of a calculation.

Table M9.E.1 MARS Subroutine

Model

Input Routines		Tracking Routines	
Combinatorial	MARS Array Input	Combinatorial	MARS Array Tracking
JOMIN1	AZIP	LOOKZ	CALI
JOMIN2	LEVEL	G1	PILOT
GENI	ARGEN	GG	CELL
ABOX	DELTA	PR	SORT
GTVLIN	ORTHOM	SORCER	BOD
	FINEFI	ZEXITS	ROTA
	UNIS	ALBERT	CORNER
	SAZAR		RTEXIT
			TRENTE
			CTRAN
			STORA
			CLEV
			RISK
			DIPR
			DUPR

Table M9.E.6. (Continued)

Variable	Definition
LDATA	Length of the integer data in the MA array excluding the words set aside for zone search information.
LTMA	Total length of the MA array.
LFPD	Length of the FPD array.
NUMR	Number of code produced zones.
IRTRU	Number of input zones.
NUMB	Number of bodies.
NIR	Number of MORSE geometry regions.

Table M9.L.7. Definitions of Variables in Common PAREM as Found in Combinatorial Geometry

Variable	Definition
XB(3)	Coordinates of the starting point of the present path. Changed in EUCLID, GOMST, and LOOKZ.
WB(3)	Direction cosines of particle trajectory. Equal to U, V, and W. Changed in EUCLID, GOMST, and LOOKZ.
WP(3)	Temporary storage of WB(3).
XP(3)	Temporary storage of XB(3). Changed in NORML.
RIN	Distance to entry calculated in GG.
ROUT	Distance to exit calculated in GG.
PINF	Machine infinity. (1.0E + 20)
DIST	Distance from XB(3) to present point.
IR	Combinatorial zone of present particle position.
IDBG	Set nonzero to initialize a debug printout.
IRPRIM	Next region to be entered after a call of G1.
NASC	Body number of last calculated intersection. Set negative to indicate source or collision point not on a body surface.

Table M9.E.7. (Continued)

Variable	Definition
LSURF	Surface of Body NASC where intersection occurred. Positive if particle is entering the body and negative when exiting.
NBO	Body number and a sign used to define zones. Input in zone description as positive when zone is contained in body and as negative if zone is outside body.
LRI	Entry surface calculated in GG.
LRO	Exit surface calculated in GG.
KLOOP	Trajectory index of present path incremented in G1.
LOOP	Index of last trajectory calculated for body NBO. If LOOP is equal to KLOOP, GG returns immediately with old values in RIN, ROUT, LRI, and LRO.
ITYPE	Body type of Body NBO (indicated BOX, SPH, etc.)
NOA	Not used.

NOTE: The order of the variables in PAREM in versions other than IBM-360 is XB, WB, IR, WP, XP, IDBG, IRPRIM, NASC, LSURF, NBO, LRI, LRO, RIN, ROUT, KLOOP, LOOP, ITYPE, PINF, NOA, DIST. Because the IBM-360 version is double precision, reordering the variables was required.

Table M9.E.8 Definitions of Variables in Common ORGI*

Variable	Definition
DISTO	Distance from point XB(3) to next scattering point. Used in G1 to avoid calculating the next zone if a scattering event occurs before the intersection.
MARKG	Set 1 in G1 if trajectory end point is reached before next intersection. Otherwise set to 0.
NMEDG	Input zone number for current collision from a LOOKZ call.
NBLZ	Packed word containing both input zone and code zone numbers for current collision stored in LOOKZ.
BLZOLD	Packed word containing code and input zone numbers for previous collision.
IRPOLD	Input zone number for previous collision.

*Variable names are not the same in all routines. Also on non-IBM-360 machines, the order of the variables is MARK, DIST, NMEDG. Reordering resulted from the conversiin of the IBM-360 version to double precision.

Table M9.E.9. Definitions of variables in COMMON TAPE

VARIABLE NAME	DEFINITION
INT	Logical number of standard input unit
IOT	Logical number of standard output unit
IOUT	Logical number of scratch unit for geometry data
IOU2	Logical number of second scratch unit
IDM(4)	Not used at present

Table M9.E.10. Definitions of variables in COMMON MGOMV

VARIABLE NAME	DEFINITION
MUS	Pointer for array indicating type of universe - simple or comjom.
MUZ	Pointer for array indicating last zone for a given universe.
LL	Current universe number.
IPRET	Signal - when = 1 PILOT returns at every boundary crossing rather than tracking until media changes.
IFLOW	Signal indicating direction of crossing. Inward if =0; outward if =1.
IECT	Error counter - prints messages first 10 times. MORSE-SGC abends on error number 11.
NLO	Signal when changing levels - = 0 - means same level. = -1 - means down one level. = +1 - means up one level.
IGX	Signal indicating entry to array if 0 or exit from array if 1.

Table M9.E.11. Definitions of variables in COMMON ARAR

VARIABLE NAME	DEFINITION
NBY	Body number
NLEV	Number of geometry levels.
NAR	Number of arrays.
NQ	= 1 for MORSE
IAW	Number of universes.
IAY	Length of lattice cell array.
NF	Maximum size of array lattice.
NX1(1)	Current x lattice position.
NX1(2)	Current y lattice position.
NX1(3)	Current z lattice position.
	NX1(1-3) = 0 if outside of array.

Table M9.E.12. Definitions of variables in COMMON ARK

VARIABLE NAME	DEFINITION
LM	Content of lattice position of current array > 0 for a universe; < 0 for an array.
NLU	Current array number.
NBB	Reference body for current array
NZY(1-3)	Current lattice position in array NLU; i.e., X,Y,Z position.
XD(1-3)	Current X,Y,Z of particle.

M9.G DIAGNOSTIC MESSAGES FROM MARS

There are numerous error messages printed out by the MARS geometry package. These messages aid the user in determining the cause of the trouble when an error is detected by MARS during the input step or during particle tracking. Sections M9.G.1 and M9.G.2 give the actual message that is printed, the subroutine name where the error was detected, and an explanation of the message and/or its cause.

M9.G.1 DIAGNOSTIC MESSAGES FROM MARS INPUT MODULE

Subroutine Printing Message	Message	Meaning
ARGEN	ARRAY NO _ IS IM-PROPERLY DEFINED - FATAL ERROR IN ARGEN	The array size exceeds the size of the array lattice (NCMAX).
ARGEN	ARGEN IXY= __ IX= __ MR= __ NA= __ NE= __ NMAX= __	Error was detected in DELTA. See message from DELTA.
AZIP	FATAL ERROR - UNIVERSE NO. __ IS NOT DEFINED.	Either the universe number is incorrect or the media number is not -1000.
DELTA	IN DELTA.UNIVERSE REFERENCE BODY NOT RPP OR BOX IBOD= __ IBO= __ MA INDEX= ITYPE= __ ND= __	Self-explanatory. Check the input data. IBO is body number.
FINEFI	WARNING - IN FINEFI NO REFERENCE WAS MADE TO ARRAY NO. _	One of the arrays specified is not referenced.
FINEFI	IN FINEFI INVALID ARRAY REFERENCE IN ARRAY __ PO-SITION ___ TO NON-EXISTENT ARRAY __	Array number exceeds maximum (NAR).
FINEFI	FINEFI FATAL ERROR - ARRAY POSITION _ RE-FERS TO AN INVALID OR UNDEFINED UNIVERSE -	Universe number exceeds number of universes (IBOD) in system.
FINEFI	IN FINEFI ARRAY NO._ POSITION ___ REFERS TO UNIVERSE _ DE-FINED BY BODY _ WHICH IS NOT AN RPP OR BOX _ FATAL ERROR	The reference body for an array must be either a BOX or an RPP. Check input data.
FINEFI	FINEFI - FATAL ERROR IN ARRAY _ ELEMENT POSITION ___ CONTAINS UNIVERSE MISFIT IN POSITION.	Self-explanatory. Check input.

Subroutine Printing Message	Message	Meaning
FINEFI	IN FINEFI ARRAY _ HAS NO VALID ELEMENTS -VERY FATAL	This message is usually in addition to one of the others from FINEFI. All lattice cell positions have been checked for fit prior to this message.
FINEFI	FINEFI - CODE ZONE _ CONTAINS ARRAY BUT IS DEFINED WITH MORE THAN ONE BODY _ WARNING _ FIRST BODY WILL BE REFERENCE BODY	Self-explanatory. Check input
FINEFI	FINEFI FATAL ERROR - CODE ZONE _ REFERENCES AN INVALID ARRAY _	The array number is > number of arrays (NAR) in the system.
FINEFI	FINEFI FATAL ERROR - CODE ZONE CONTAINS ARRAY _ INSIDE BODY _ REFERENCED BODY IS NOT AN RPP OR BOX	The reference body for an array must be either a BOX or an RPP. Check input data.
FINEFI	FINEFI FATAL ERROR - ARRAY _ IN BODY _ REFERENCED IN CODE ZONE DOES NOT FIT	The overlap of lattice cell boundary and input zone boundary is too large.
FINEFI	FINEFI FATAL ERROR - YOU HAVE AT LEAST ONE ARRAY BUT NO ARRAY WAS REFERENCED IN A LEVEL ZERO ZONE	Self-explanatory. Probable input error.
GTVLIN	**********ERROR IN VOLUME CALCULATION********** JF=_ NIR=_	The number of regions for which volumes were calculated does not equal the number in the geometry.
JOMIN1	ITYPE _ DOES NOT EQUAL ANY OF THE FOLLOWING _____	The body type given does not exist in the code.
JOMIN1	EXCEEDED CORE DURING CG INPUT ISTR= _ N= _ NADD= LIM= LTMA= LFPD= _ NUMR= _ IRTRU= _ NAZT= _IBODT= _	The space allocated for the data is insufficient. ISTR + N + 1 is amount required up to this point. LIM is space available.

Subroutine Printing Message	Message	Meaning
SAZAR	NMOST INCREASED TO _ BY SUBROUTINE SAZAR	An array edit was requested, but there was inadequate space to save it. Space was added.
UNIS	WARNING -- UNIVERSE _ IS NOT SIMPLE CODE ZONE _ CONTAINS ARRAY _.	A negative media number was found in a simple universe. UNIS changes it to combinatorial universe.
UNIS	CODE ZONE _ IN UNIVERSE _ CANNOT BE A SIMPLE UNIVERSE.	The number of bodies in this code zone is >2, so it is not simple. UNIS changes it.
UNIS	BODY _ IN UNIVERSE _ IS NOT REFERENCED POSITIVE IN THE UNIVERSE.	Fatal error.

M9.G.2 DIAGNOSTIC MESSAGES FROM MARS TRACKING EVENTS

Subroutine Printing Message	Message	Meaning
CALI	CALI NL,NLU,IR, IRPRIM,NMED,NLEV __	Printed only if debug switch IDBG is on (>0)
CALI	ERROR IN CALI FROM CELL * LZ=_ LU=_ IER=_	Either CELL has set the error signal IER>0 or lattice cell is vacant (LZ>0) on entering array.
CALI	ERROR IN CALI * NMED,IPS,LZ,IER,LU, IR,IRPRIM,LM,LL,NL _	When this message follows one above see that explanation. If IPS>2, there is error on exiting array.
CLEV	ERROR IN CLEV * NL1=_ NL2=_ NGY=_ .	Level NL1 is lower than level NL2. It should be higher because code translates from a higher to a lower level.
G1	NO VALID DISTANCE IN G1 * IR,XB,WB,DIST= _	G1 could not determine the next body that the ray will intersect. There is a probable error in user's geometry specifications, or he may have written over his geometry data.
G1	********** GEOMETRY SEARCH ARRAY FULL **********	In order to save some computer time, user may want to increase the value of NAZ in his input. Only harm done is increase in computer time which is often insignificant.
G1	IR=_ XB=_,_,_ WB=_,_,_ DIST=_ LL=_ MA ARRAY Entire MA array printed. FPD ARRAY Entire FPD array printed. ****EXIT BEING CALLED FROM G1. NEXT ZONE NOT FOUND	G1 could not find tne next zone that the particle would enter It checked all zones and is saying that the particle won't enter any of them. Probable error in zone specifications Subroutine PR called to print values from commons GOMLOC, DBG, and PAREM. Among the variables printed the following apply to the maximum body number and therefore have no significance - NBO,N,NUM,ITYPE,SIN,IRP and LOCAT. The most important variable is NASC which is the number of the next body the ray will intersect. That is, the code cannot find what zone this body is in. Check zone specification for this body.

Subroutine Printing Message	Message		Meaning
GG	IN GG IR=_	ITYPE= _ NBO=_ .	ITYPE is not one of the body types (1 - 9) allowed. User has either overstored MA array or has used a body number out of range in the zone description.
LOOKZ	IR=_ WB=_, DIST=_ LL=_ MA ARRAY Entire MA array printed. FPD ARRAY Entire FPD array printed. ****EXIT BEING CALLED FROM LOOKZ	XB=_,_,_ _,_	Zone has not been found. Check the zone specifications for error.
NORML	INVALID REGION OR BODY IN NORMAL. IR=_ NASC=_		NASC is not in region IR. Check whether IR or NASC has been overwritten.
NORML	ROUND-OFF ERROR IN NORMAL NBO=_ LSURF=_ XP=_,_,_		Self-explanatory. Applies only to ELL and BOX.
PILOT	PROBLEM ENCOUNTERED IN ARRAY PROCESSING, PARTICLE WILL BE TERMINATED AS ESCAPE.		Printed when PILOT is in a loop when processing an array. Particle is probably unable to enter or leave an array. May be input error.
PILOT	PILOT CRASH		Printed only if debug switch IDBG is on (>0).
PILOT	ERROR IN PILOT ON ENTERING ARRAY		If IER>0, previous error message from SORT explains error. If LZ>0 or LM=0, lattice cell is vacant or cell position is out-of-range (<0 or >maximum)
PILOT	ERROR IN PILOT ON EXITING ARRAY		If IER>0, previous error message from SORT explains error. If IEXIT=1 or LM=0, lattice position is out of range. NL<0 means error in level number.

Subroutine Printing Message	Message	Meaning
SORT	FATAL ERROR IN SORT *** PARTICLE LOST ON ENTERING OR EXITING ARRAY. M,NMAX,IGX,LX1,LX2, NXY ___ XB,RBY(LX2),SML,SMALL	On entering array, particle position is not at a lattice cell boundary (RBY) for array; on exiting array, either lattice position is > size of array lattice or is <0 or particle position is not at lattice cell boundary.
STORA	IN STORA EITHER AN ILLEGAL LEGAL TRANSFER WAS ATTEMPTED OR AN INCORRECT ARRAY REQUEST WAS MADE *NLO=_NL=_ NLU=_LM=_.	The level number NL is wrong. Either it is greater than the number of levels (NLEV) or it is 0.

NRC FORM 335 (2-84) NRCM 1102, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

SEE INSTRUCTIONS ON THE REVERSE

1 REPORT NUMBER (Assigned by TIDC add Vol. No., if any)
NUREG/CR-0200
Volumes 1, 2, and 3
Revision 3

2 TITLE AND SUBTITLE
SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation

3 LEAVE BLANK

4 DATE REPORT COMPLETED
MONTH | YEAR

5 AUTHOR(S)
C. V. Parks

6 DATE REPORT ISSUED
MONTH: December | YEAR: 1984

7 PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

8 PROJECT/TASK/WORK UNIT NUMBER

9 FIN OR GRANT NUMBER
B0009

10 SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Division of Fuel Cycle and Material Safety
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555

11a TYPE OF REPORT
Technical

b PERIOD COVERED (Inclusive dates)

12 SUPPLEMENTARY NOTES

13 ABSTRACT (200 words or less)

SCALE (Standardized Computer Analyses for Licensing Evaluation), a multi-faceted computational system, has been developed to provide a standard analysis tool for use by the NRC staff and licensees in evaluation of nuclear fuel facility and package designs. Revision 3 describes recent additions and modifications made to the SCALE computational system. Combining this material with previously issued material will update the SCALE Manual to be consistent with the latest publicly released version of SCALE, called SCALE-3.

14 DOCUMENT ANALYSIS - a KEYWORDS/DESCRIPTORS
Criticality
Shielding
Monte Carlo
Heat Transfer
Shipping Casks

b IDENTIFIERS/OPEN-ENDED TERMS

15 AVAILABILITY STATEMENT
Unlimited

16 SECURITY CLASSIFICATION
(This page) Unclassified
(This report) Unclassified

17 NUMBER OF PAGES

18 PRICE