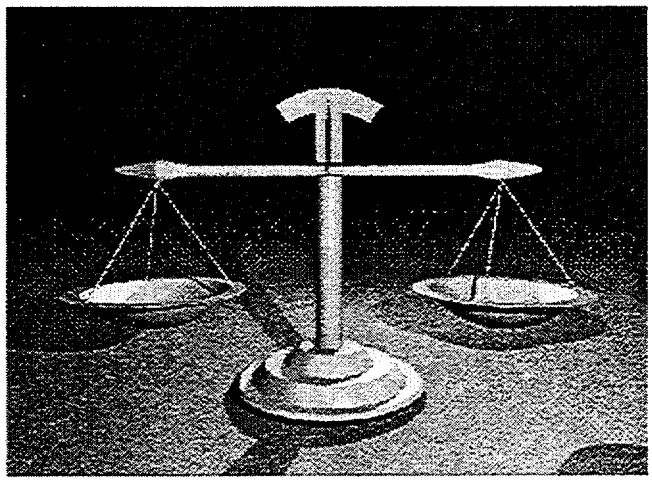
NUREG/CR-0200, Rev. 6
ORNL/NUREG/CSD-2/R6
Vol. 2, Part 3

SCALE

Version 4.4

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



Functional Modules, Part 3

Oak Ridge National Laboratory

Prepared for
U.S. Nuclear Regulatory Commission

NUREG/CR-0200, Rev. 6
ORNL/NUREG/CSD-2/R6
Vol. 2, Part 3

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

Functional Modules
F16 – F17

Manuscript Completed: February 2000
Date Published: May 2000

Oak Ridge National Laboratory
Managed by Lockheed Martin Energy Research Corp.
Oak Ridge, TN 37831

Prepared for
Spent Fuel Project Office
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
NRC Job Code B0009

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at www.nrc.gov/NRC/ADAMS/index.html.
Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, in the Code of *Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents
   U.S. Government Printing Office
   P. O. Box 37082
   Washington, DC 20402–9328
   www.access.gpo.gov/su_docs
   202–512–1800
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–533–6847 or, locally, 703–805–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:
Address: Office of the Chief Information Officer,
Reproduction and Distribution
Services Section
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
E-mail: DISTRIBUTION@nrc.gov
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address www.nrc.gov/NRC/NUREGS/indexnum.html are updated regularly and may differ from the last printed version.

Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org
212–642–4900

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG–XXXX) or agency contractors (NUREG/CR–XXXX), (2) proceedings of conferences (NUREG/CP–XXXX), (3) reports resulting from international agreements (NUREG/IA–XXXX), (4) brochures (NUREG/BR–XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG–0750).

ABSTRACT

SCALE, a modular code system for Standardized Computer Analyses Licensing Evaluation, has been developed by Oak Ridge National Laboratory at the request of the U.S. Nuclear Regulatory Commission. The SCALE system utilizes well-established computer codes and methods within standard analysis sequences that (1) allow an input format designed for the occasional user and/or novice, (2) automate the data processing and coupling between modules, and (3) provide accurate and reliable results. System development has been directed at problem-dependent cross-section processing and analysis of criticality safety, shielding, heat transfer, and depletion/decay problems. Since the initial release of SCALE in 1980, the code system has been heavily used for evaluation of nuclear fuel facility and package designs. This revision documents Version 4.4 of the system.

CONTENTS

Volume 1, Part 1: Control Modules

Volume 1, Part 2: Control Modules

---

*Obsolete with SCALE-4.0 release.
**Not included in SCALE-4.4 release.
***Cancelled. Included in Section C4.

Volume 2, Part 1: Functional Modules

F1 BONAMI: RESONANCE SELF-SHIELDING BY THE BONDARENKO METHOD (N. M. Greene)

F2 NITAWL-II: SCALE SYSTEM MODULE FOR PERFORMING RESONANCE SHIELDING AND WORKING LIBRARY PRODUCTION (N. M. Greene, L. M. Petrie, R. M. Westfall)

F3 XSDRNPM: A ONE-DIMENSIONAL DISCRETE-ORDINATES CODE FOR TRANSPORT ANALYSIS (N. M. Greene, L. M. Petrie)

F4 XSDOSE: A MODULE FOR CALCULATING FLUXES AND DOSE RATES AT POINTS OUTSIDE A SHIELD (J. A. Bucholz)

F5* KENO IV/S: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM

F6 COUPLE: SCALE SYSTEM MODULE TO PROCESS PROBLEM-DEPENDENT CROSS SECTIONS AND NEUTRON SPECTRAL DATA FOR ORIGEN-S DATA ANALYSES (O. W. Hermann)

F7 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)

F8 ICE: MODULE TO MIX MULTIGROUP CROSS SECTIONS (N. M. Greene, L. M. Petrie, S. K. Fraley)

Volume 2, Part 2: Functional Modules

F9 MORSE-SGC FOR THE SCALE SYSTEM (J. T. West, T. J. Hoffman, M. B. Emmett)

F10 HEATING 7.2 USER'S MANUAL (K. W. Childs)

F11 KENO V.a: AN IMPROVED MONTE CARLO CRITICALITY PROGRAM WITH SUPERGROUPING (L. M. Petrie, N. F. Landers)

F12* JUNEBUG-II: A THREE-DIMENSIONAL GEOMETRY PLOTTING CODE

F13* HEATPLOT-S: A TEMPERATURE DISTRIBUTION PLOTTING PROGRAM FOR HEATING

---

*Obsolete with SCALE-4.0 release.
**Not included in SCALE-4.4 release.
***Cancelled. Included in Section C4.

Volume 2, Part 3: Functional Modules

Volume 3: Miscellaneous

---

*Obsolete with SCALE-4.0 release.
**Not included in SCALE-4.4 release.
***Cancelled. Included in Section C4.

Volume 3: Miscellaneous (continued)

*Obsolete with SCALE-4.0 release.
**Not included in SCALE-4.4 release.
***Cancelled. Included in Section C4.

PREFACE

Introduction

This Manual represents Revision 6 of the user documentation for the modular code system referred to as SCALE. The previous revision documented version 4.3 of SCALE, released in October 1995. This revision documents version 4.4a of SCALE. Prior to the release of version 4.4a, SCALE 4.4 was released in September 1998. Many minor corrections and enhancements have been made since that time and are being included in SCALE 4.4a. The corrections and enhancements in versions 4.4a and 4.4 are documented separately below. All modifications in version 4.4 are included in version 4.4a.

Background

The history of the SCALE code system dates back to 1969 when the current Computational Physics and Engineering Division at Oak Ridge National Laboratory (ORNL) began providing the transportation package certification staff at the U.S. Atomic Energy Commission with computational support in the use of the new KENO code for performing criticality safety assessments with the statistical Monte Carlo method. From 1969 to 1976 the certification staff relied on the ORNL staff to assist them in the correct use of codes and data for criticality, shielding, and heat transfer analyses of transportation packages. However, the certification staff learned that, with only occasional use of the codes, it was difficult to become proficient in performing the calculations often needed for an independent safety review. Thus, shortly after the move of the certification staff to the U.S. Nuclear Regulatory Commission (NRC), the NRC staff proposed the development of an easy-to-use analysis system that provided the technical capabilities of the individual modules with which they were familiar. With this proposal, the concept of the Standardized Computer Analyses for Licensing Evaluation (SCALE) code system was born.

The NRC staff provided ORNL with some general development criteria for SCALE: (1) focus on applications related to nuclear fuel facilities and package designs, (2) use well-established computer codes and data libraries, (3) design an input format for the occasional or novice user, (4) prepare "standard" analysis sequences (control modules) that will automate the use of multiple codes (functional modules) and data to perform a system analysis, and (5) provide complete documentation and public availability. With these criteria the ORNL staff laid out the framework for the SCALE system and began development efforts. The initial version (Version 0) of the SCALE Manual was published in July 1980. Then, as now, the Manual is divided into three volumes — Volume 1 for the control module documentation (Sections C4, C6, D1, S1- S5, and H1), Volume 2 for the functional module documentation (Sections F1- F17), and Volume 3 for the documentation of data libraries, and subroutine libraries, and miscellaneous utilities (Sections M1- M17).

System Overview

The original concept of SCALE was to provide "standardized" sequences where the user had very few analysis options in addition to the geometry model and materials. Input for the control modules has been designed to be free-form with extensive use of keywords and engineering-type input requirements. The more flexible functional modules have a more difficult input logic and require the user to interface the data sets necessary to run the modules in a stand-alone fashion. As the system has grown in popularity over the years and additional options have been requested, the control modules have been improved to allow sophisticated users additional access to the numerous capabilities within the functional modules. However, the most

important feature of the SCALE system remains the capability to simplify the user knowledge and effort required to prepare material mixtures and to perform adequate problem-dependent cross-section processing.

The modules available in Version 0 of SCALE were for criticality safety analysis sequences (CSAS) that provided automated material and cross-section processing prior to a one-dimensional (1-D) or multidimensional criticality analysis. Since that time the capabilities of the system have been significantly expanded to provide additional CSAS capabilities, new shielding analysis sequences (SAS) that also include depletion/decay capabilities for spent fuel characterization, and a heat transfer analysis sequence (HTAS). At the center of the CSAS and SAS sequences is the library of subroutines referred to as the Material Information Processor or MIPLIB (see Section M7). The purpose of MIPLIB is to allow users to specify problem materials using easily remembered and easily recognizable keywords that are associated with mixtures, elements, and nuclides provided in the Standard Composition Library (see Section M8). MIPLIB also uses other keywords and simple geometry input specifications to prepare input for the modules that perform the problem-dependent cross-section processing: BONAMI, NITAWL-II, and XSDRNPM. A keyword supplied by the user selects the cross-section library from a standard set provided in SCALE (see Section M4) or designates the reference to a user-supplied library. Several utility modules from AMPX[1] have been included to provide users with the capability to edit the cross-section data and reformat user-supplied libraries for use in SCALE.

Over the history of the project several modules have been removed from the system because they are no longer supported by the development staff at ORNL. Tables 1 and 2 provide a summary of the major applications of each of the control modules and functional modules currently in the SCALE code system. The control modules were designed to provide the system analysis capability originally requested by the NRC staff. The CSAS module (sometimes denoted as the CSAS4 module and documented in Section C4) is the primary control module designed for the calculation of the neutron multiplication factor of a system. Eight sequences enable general analysis of a 1-D system model or a multidimensional system model, capabilities to search on geometry spacing, and problem-dependent cross-section processing for use in executing stand-alone functional modules. CSAS6 is a newer criticality control module to provide automated problem-dependent cross-section processing and criticality calculations via the KENO-VI functional module. The SAS1 and SAS3 modules (see Sections S1 and S3, respectively) provide general 1-D deterministic and 3-D Monte Carlo analysis capabilities. The SAS2 module (see Section S2) was originally developed to perform a depletion/decay calculation to obtain spent fuel radiation source terms that were subsequently input automatically to a 1-D, radial shielding analysis in a cylindrical geometry. Over time the depletion/decay portion of the SAS2 module has been significantly enhanced and interfacing to the other shielding modules has been provided. An alternative sequence for depletion/decay calculations is ORIGEN-ARP (Section D1), which interpolates pre-generated ORIGEN-S cross-section libraries versus enrichment, burnup, and moderator density. The SAS4 module (see Section S4) enables automated particle biasing for a Monte Carlo analysis of a transportation package-type geometry. The HTAS1 module (see Section H1) is the only heat transfer control module and uses the various capabilities of the HEATING code to perform different sequences of steady-state and transient analysis that enable the normal and accident conditions of a transportation package to be evaluated. Like SAS4, the HTAS1 module is limited to a package-type geometry. The capability to perform a point-kernel shielding analysis within the SCALE system has been developed in the QADS control module.

A 238-energy-group neutron cross-section library based on ENDF/B-V[2] is the latest cross-section library in SCALE. All the nuclides that are available in ENDF/B-V are in the library. A 44-group library has been collapsed from this 238-group library and validated against numerous critical measurements.[3] These libraries are available in this version of SCALE.

Technical Assistance and Updates

To obtain technical assistance regarding the installation and use of SCALE, download software updates, or report problems, you may contact us through the following channels:

- E-mail questions to **scalehelp@ornl.gov**
- The SCALE Users Electronic Notebook on the Web:
  http://www-rsicc.ornl.gov/ENOTE/enotscal.html
- SCALE Web Site (including Download, Training, Benchmarks, and Newsletter pages):
  http://www.cped.ornl.gov/cad_nea/text/scale-home.html
- SCALE Newsletter:
  http://www.cped.ornl.gov/cad_nea/text/scale_news.html
- FAX to SCALE Help, 815-327-6460 or 865-576-3513

Significant Updates in SCALE 4.4a

A significant number of updates have been made to SCALE since the initial release of SCALE 4.4 in September 1998. Most of these updates were minor corrections or enhancements. Because some of these updates could be important to SCALE users, this interim release of SCALE 4.4a is being made available.

SAS4 and PICTURE were enhanced to allow the generation of two-dimensional (2-D) plots when the "PARM=CHECK" option is used. This option is similar to the plotting option in the CSAS criticality sequences. Another innovation was the addition of an option that allows users to specify an X-Y, X-Z, or Y-Z plot and have the code automatically calculate the cosines used for the plot.

A discrepancy in scoring boundary crossings of surface detectors was corrected in MORSE. Contributions to user-specified surface detectors in MORSE in SCALE 4.4 could have been underestimated because of a failure to determine which surface detector to score. This failure was due to the comparison of a single precision variable to a double precision variable. Most affected cases would have a zero result for the surface detector, indicating that no particles have crossed the surface detector boundary. Detector location coordinates of four digits or less would not be expected to experience this problem.

A coding error introduced in QAD-CGGP in SCALE 4.4 has been identified and corrected. Because of inconsistent array dimensions, if more than a very limited number of bodies are input in one zone, the additional zone data are lost or stored incorrectly. This situation typically causes the code to fail. Though extremely unlikely, it might be possible for a case like this to run if the incorrectly stored geometry happened to be valid. SCALE 4.4 users should check under the "input zone data" header in the QAD-CGGP output to verify that the zone data agree with their input.

SAS2 was corrected to fix an error introduced in SCALE 4.4 that caused the PARM=OLDSAS2 option to fail. Another discrepancy introduced in SCALE 4.4 caused spent fuel isotopic data written to file FT72F001 to be incorrect in certain cases. This error, which has been corrected, occurred in cases where burnable poison rods or other inserts are removed from or inserted into the fuel assembly between fuel cycles. SAS2 can now correctly handle multiple fuel zones in the path B model. A minor discrepancy was corrected where invalid characters were being written to title records in the ORIGEN-S binary library. Some text editors could not read the SAS2 output file when invalid characters were present.

A large number of enhancements were made to XSDRNPM. The Fortran source for XSDRNPM was converted to Fortran 90 free format. The input/output units were all moved to the 0$ array. The energy of the

average lethargy causing fission was added to the balance tables. The output files from the balance tables and the activities were modified and converted to ASCII files. A new ASCII file was created that contains the input and derived data from a problem. The coarse mesh generation algorithm used in rebalancing the inner iterations was modified to correct a problem that prevented convergence for a very small class of problems. The code was modified to recycle if the final iteration performed after convergence failed the convergence test. For group banding cases, convergence is now reset after initial convergence to an order of magnitude less than overall convergence to prevent looping through iterations and never converging. The default value for flux convergence tolerance, PTC, was reduced from 10^{-4} to 10^{-5}. The calculation of activities by interval, an option that was available many years ago, was reintroduced in the code.

The XSDRNPM mesh generation algorithm in MIPLIB was modified to address two problems: (a) insufficient number of mesh intervals for thick reflectors of low absorbing material and (b) too many mesh intervals for highly absorbing regions. New input options to override the automatic mesh generation were added. Although this enhancement was designed primarily for CSAS1X, it potentially affects all control modules except SAS2H that use XSDRNPM.

KENO-VI was modified to detect intersecting HOLEs in the global unit. A problem will now terminate if intersecting HOLEs are detected in the global unit. Intersecting HOLEs are illegal in KENO-VI geometry but were not detected in the global unit in SCALE 4.4. Intersecting HOLEs in units other than the global unit are detected during tracking of particles through the intersecting regions. Several corrections were made to KENO-VI to prevent a particle from becoming lost and causing the code to enter an infinite loop.

MORSE was updated to correct a problem in determining the correct day of the week for dates after December 31, 1999.

Many other minor changes included in SCALE 4.4a are listed under "SCALE 4.4a Minor Modifications."

SCALE 4.4a Minor Modifications

In addition to the major enhancements noted above, SCALE 4.4a contains many minor modifications, including corrections to errors in SCALE 4.4 and changes to improve portability to different computing platforms. Note that some of these modifications may be duplicate listings of items mentioned in the previous section.

PICTURE **MRR98-056**

Updated to handle the call by SAS4 when the "PARM=CHECK" option (added to the SAS4 control module in MRR98-057) is used. Also, added an option that allows users to specify an X-Y, X-Z, or Y-Z plot and have the code automatically calculate the cosines used for the plot.

SAS4 **MRR98-057**

Added a "PARM=CHECK" option that calls PICTURE from within the SAS4 to plot geometry but not run MORSE. SAS4 prepares or reads MARS geometry input data, reads PICTURE input, and calls PICTURE. Several other changes were made to error messages and formats.

SAS2 **MRR98-058**

Updated to correct an error introduced in SCALE 4.4 that caused the PARM=OLDSAS2 option to fail. Also corrected another problem introduced in SCALE 4.4 that caused spent fuel isotopic data written to file FT72F001 to be incorrect in certain cases where burnable poison rods or other inserts are removed from or inserted into the fuel assembly between fuel cycles.

MORSE **MRR98-059**

Corrected a discrepancy in scoring boundary crossings of surface detectors. A roundoff error caused by comparison of a double precision variable to a single precision constant resulted in boundary crossings not being scored. The epsilon value for the comparison was also increased from 0.0005 to 0.001.

SAS4 **MRR98-060**

Updated to correct the dimensions on two arrays. Also changed a test comparing 2 floating point variable names equivalenced to integer variables to use function ISET. (This test has previously caused floating point underflows on some platforms.)

QADS **MRR98-061**

Added a test on the MIPLIB error flag that terminates execution of the problem if an error occurred.

KENO-VI **MRR98-062**

Updated to allow a particle to cross from one hole directly into an adjacent hole even if the crossing is outside the allowed tolerances. This prevents some cases from entering an infinite loop.

SAS2 **MRR98-063**

Corrected minor discrepancy that resulted in invalid characters being written to title records in ORIGEN-S binary library. Some text editors could not read SAS2 output file when invalid characters were present.

MIPLIB **MRR98-064**

Updated to allow control modules to specify a sensitivity library from NITAWL and to allow number density input for an element that has multiple isotopes.

XSDRNPM **MRR98-065**

The Fortran source for XSDRNPM was converted to Fortran 90 free format. The input-output units were all moved to the 0$ array. The energy of the average lethargy causing fission was added to the balance tables. The output files from the balance tables and the activities were modified and converted to ASCII files. A new file was created which contains the input and derived data from a problem. The flux file was changed to double precision. The code was modified to not run with fluxes out of core unless explicitly requested in the input. The coarse mesh generation algorithm used in rebalancing the inner iterations was modified to correct a problem that prevented convergence a very small class of problems.

C5TOC6/K5TOK6 **MRR98-066/MRR98-067**

The input file generated for CSAS6/KENO VI incorrectly labeled regions generated to surround HOLEs if there were more than one region in a unit that contained HOLEs. Because of a change in KENO-VI, these regions should no longer need to be generated. Subroutine PUNCH_GEOM was modified to not generate these regions.

QAD-CGGP **MRR98-068**

Updated to correct an error introduced in SCALE 4.4. The dimension on one variable in the geometry was not updated when the input format was changed to match that of MARS input. This caused some jobs to fail. Also, updated to correct misspelled name of unit used for error output.

ARPLIB **MRR99-001**

Updated to accept either lower or upper case input.

PRISM **MRR99-002**
Updated to accept either lower or upper case input.

XSECLIST **MRR99-003**
Updated to accept either lower or upper case input.

SAS2 **MRR99-004**
Updated for compatibility with the newest revisions to XSDRNPM (MRR98-065). The routines that wrote the XSDRNPM input files needed to be changed to account for the changes to XSDRNPM input. Subroutine COPYNX had to be changed to add the 0$ array to the XSDRNPM input file, and to move setting the logical unit number of the flux output file from the 2$ array to the 0$ array.

UNIXLIB **MRR99-006**
Changes to update XSDRNPM required a double precision ERF function. This function is part of the Fortran intrinsic library for the DEC Alpha's and the IBM RS/6000, but is not part of that library for the HP or the SUN workstations. This modification provided an update for the necessary routines to compute the double precision ERF when it is not part of the intrinsic library.

BONAMI **MRR99-007**
Updated to correct a problem that caused cases to fail when zero number density input is used.

SAS4 **MRR99-008**
Updated to change the convergence criteria because the criteria in XSDRNPM were changed. Also added an input variable NDAB to allow the user to specify the number of direct access blocks allocated.

XSDOSE **MRR99-009**
Added option to turn off angular flux print and made the default to be no angular flux print.

MODIFY **MRR99-010**
Updated subroutine LODATA for compatibility with changes in MRR98-064(MIPLIB).

XSDRNPM **MRR99-011**
(1) The code was modified to re-cycle if the final iteration performed after convergence failed the convergence test. For group banding cases, convergence is now reset after initial convergence to an order of magnitude less than overall convergence to prevent looping through iterations and never converting. (2) The default value for flux convergence tolerance, PTC, was reduced from 10^{-4} to 10^{-5}. (3) Errors were corrected in the calculation of activities by interval. This previously undocumented option is now documented in the XSDRNPM input description.

KENO-VI **MRR99-012**
Modified KENO-VI to detect intersecting HOLEs in the global unit. A problem will now terminate if intersecting HOLEs are detected in the global unit. Intersecting HOLEs are illegal in KENO-VI geometry but were not detected in the global unit in SCALE 4.4. Intersecting HOLEs in units other than the global unit are detected during tracking of particles through the intersecting regions.

ORIGEN **MRR99-014**
A new subroutine was added to provide the user the option of more significant digits in the output tables. Unit 71 was set as the default file number for the binary file containing concentrations and spectral data.

MIPLIB **MRR99-015**

The XSDRNPM mesh generation algorithm was modified to address two problems: (a) insufficient number of mesh intervals for thick reflectors of low absorbing material and (b) too many mesh intervals for highly absorbing regions. New input options to override the automatic mesh generation were added too. Consistent with MRR99-011, the default value of PTC was reduced from 10^{-4} to 10^{-5}. Although this enhancement was designed primarily for CSAS1X, it potentially affects all control modules that use XSDRNPM.

KENO-VI **MRR99-016**

The code was corrected to define LCHK as a logical variable in subroutine POSIT. In addition, an IMPLICIT NONE statement has been added to the beginning of the subroutine. All variables have been explicitly typed as appropriate.

MODIFY **MRR99-017**

Program MODIFY was changed for consistency with the change in the direct access file made in MRR99-015(MIPLIB).

KMART **MRR99-018**

An error that resulted in calculated volumes of zero for hemicylinders and arrays (if an array number was skipped) was corrected.

CSAS6 **MRR99-019**

The argument list for the call to subroutine PRTPLT was modified for consistency with changes made to KENO-VI in MRR99-012.

SAS2 **MRR99-020**

Calls to subroutine EPSIG were changed for consistency with MIPLIB modifications in MRR99-015.

MORSE **MRR99-021**

The code was updated to correct a problem in determining the day of the week for dates after December 31, 1999.

KENO-VI **MRR99-022**

The code was updated to fix a roundoff problem that sometimes caused particles to get into an infinite loop when they transferred from one array location to another but in the process missed the unit boundary.

SUBLIB **MRR99-023**

Subroutine YREAD turns off the normal invalid character check done by the free form reading routines, but does not make any checks of its own for invalid characters. This can lead to erroneous results in some cases where a user mistypes a character when entering the array data to KENO. Checks were added to the array reading routine to give warning messages if illegal characters are read. Corrections were also made so YREAD would store correctly a double precision array.

KENO-VI **MRR99-024**

The code was updated to fix a problem where the unit boundary shares surfaces with other geometry regions and the unit is in an array. This problem could result in an infinite loop because the code fails to detect a particle crossing the boundary.

SAS2 **MRR99-025**

The following modifications were made: (1) The calculation of the light element concentrations in ORIGEN-S was corrected when multiple fuel zones (MX=500) are used in the PATH B model. The code previously assumed only one fuel zone was present, and did not sum the zone volumes when multiple zones were present, resulting in erroneous light element concentrations in the ORIGEN-S depletion calculations. (2) The depletion of light element nuclides with mixture numbers 50 through 59 is now permitted. (3) The use of 1$ data for MXT (input level 3) when reading a second working library in NITAWL is now permitted.

HEATING **MRR99-027**

The code was updated, including Fortran 90 dynamic memory allocation, to improve portability on both workstation and PC platforms.

UNIXLIB **MRR99-028**

Subroutine JSTIME was modified to return time to the precision supplied by the system.

SAS4 Sample Problems **DRR99-001**

Updated SAS4 sample problems 1, 3, and 5 to remove references to variables FR1, FR2, FR3, and FR4, which became obsolete in SCALE 4.4.

XSDOSE Sample Problem **DRR99-002**

The XSDOSE section of the SCALE Manual documents the output of the sample problem and includes the printing of the fluxes. The input was modified to turn on the new angular flux print option in XSDOSE (see MRR99-009).

KENO V.a Sample Problems **DRR99-003**

Input data for sample problems 17 and 18 were changed. The number of neutrons started in sample problem 18 was changed to agree with the number per generation. Problem 17 was changed to specify the NBK parameter because the default was not large enough.

238- and 44-Group ENDF/B-V Libraries **DRRs 99-004 and 99-005**

Changes were made because problems were discovered with ^{238}Np, ^{250}Cf, ^{253}Cf, ^{249}Bk, ^{242}Am, and ^{233}Pa. The corrections for ^{250}Cf, ^{233}Pa, and ^{249}Bk were very minor and should have no important effects. However, significant errors were identified for ^{238}Np, ^{253}Cf, and ^{242}Am. In addition, these three nuclides do not have fission cross sections specified in the fast region in ENDF/B-V. This omission is obviously wrong, and because it could lead to very non-conservative answers for k_{eff}, these three nuclides were removed from the library.

Major Enhancements in SCALE 4.4

Many enhancements and corrections were made to SCALE in the three years between the release of SCALE-4.3 and 4.4. SCALE 4.4 is compatible with the year 2000 (see "SCALE 4.4 is Year 2000 Compliant"). User-specified surface detectors have been added to SAS4/MORSE to improve its computational flexibility and efficiency (see "Improvements to SAS4 and MORSE"). The KENO-VI input requirements for HOLEs have been simplified and made more consistent with KENO V.a (see "KENO-VI HOLE Input is Simplified"). Additionally, some significant improvements to the speed and stability of KENO-VI have been made (see "KENO-VI Stability and Speed Improvements"). A large number of changes have been made to the SAS2H depletion module (see "SAS2H Corrections and Enhancements").

Several enhancements have been made to the PC version of SCALE 4.4. A significant effort has been made to minimize the programming differences between the PC and Unix workstation versions. Both versions

will contain the same modules. The heat transfer modules HTAS1, HEATING, and the HEATING auxiliary codes are now available in the PC version for the first time. The PC version can recognize MS-DOS, Windows 95, Windows 98, and Windows NT operating systems and run under any of these systems from a single user command. CSAS can now be run directly from the CSASIN input processor.

ORIGEN-ARP, which was first released in the PC version of SCALE-4.3, has been enhanced and now runs under the SCALE driver, so it can run easily on workstations as well as PCs. ORIGEN-ARP has been improved significantly. ARP now interpolates on moderator density as well as burnup and enrichment for BWR fuel types. Several auxiliary codes have been added that enable users to generate their own ORIGEN-ARP cross-section libraries via SAS2H.

The default number of histories in KENO V.a and KENO-VI have been increased to 200,000 to produce more statistically accurate results. Color plots are now the defaults in both these codes.

PICTURE has been upgraded to generate two-dimensional (2-D) color plots of MORSE/MARS and QADS/QAD-CGGP geometry models like the color plots generated by KENO V.a and KENO-VI in SCALE-4.3. A new utility, LEGEND, has been created that adds a color/material legend and title to the color plots generated by KENO and PICTURE.

KMART is a new module to allow post-processing of a KENO V.a restart file, along with a working format cross-section library, to generate activities and/or broad-group fluxes and to compute the fission production activity if the components are available in the working cross-section library for the requested nuclide.

The group banding procedure in XSDRNPM was modified to significantly improve convergence for many large problems. Two examples of improvement include a fixed-source calculation with an 85% reduction in run-time and a k_{eff} calculation with a 50% reduction in run-time.

A correction was made to MIPLIB to allow the use of moderator in the gap region of a lattice cell calculation. Prior to this correction, if the same mixture number was specified in the moderator and the gap regions, the moderator density was incorrectly increased by a factor of two in the Dancoff factor calculation. In CSAS or CSAS6, this error results in a non-conservative calculated k_{eff} value that is approximately 0.5 to 1% low.

Other additions to SCALE 4.4 include the capability to perform a one-dimensional criticality search in CSAS1X (see "Criticality Search in CSAS1X"); the new KENO biasing weights library for 16-, 27-, 44-, 218-, and 238-group problems (see "New KENO Weights Library and Modules to Generate Weights"); the C5TOC6 and K5TOK6 conversion utilities for KENO-VI, and the Q0RDPN binary to ASCII conversion utility for functional module FIDO input files (see "New SCALE Utility Programs"); and the new zirconium hydride cross section data in the 238- and 44-group ENDF/B-V libraries (see "Zirconium Hydride Cross Sections").

The SCALE manual is distributed in electronic format on CD with the software. The manual is formatted in PDF files that can be read, searched, and printed using Adobe Acrobat Reader with Search. Users who desire a hard copy of the manual may obtain one from RSICC for an additional charge to cover reproduction costs.

Many other minor changes included in SCALE 4.4 are listed under "SCALE 4.4 Minor Modifications."

SCALE 4.4 is Year 2000 Compliant

Current and earlier versions of SCALE should calculate results correctly beyond the year 2000. However, when the year 2000 occurs, the output from some codes in these earlier versions will incorrectly display the year as 1900 instead of 2000. All known instances of this problem have been corrected in SCALE 4.4.

Improvements to SAS4 and MORSE

SAS4 and MORSE have been enhanced to allow users to specify multiple non-overlapping surface detectors on each surface (previously defaulted to 4 locations). These surface detectors can be divided into "sub-detectors" that enable the user to obtain detailed dose rate profiles. The flexibility in the use of these surface detectors makes them suitable for the substitution of point detectors, which are much less computationally efficient. Another enhancement to SAS4 was the addition of two options to pass data to PICTURE for plotting. One option generates geometry data only for the purpose of running PICTURE to view 2-D slices of the geometry. The other option provides "PARM=CHECK" option that calls PICTURE from within the SAS4 to plot geometry but not run MORSE. SAS4 prepares or reads MARS geometry input data, reads PICTURE input, and calls PICTURE.

Improvements to MORSE include orderly termination of a problem when errors in tracking to detector exceed a limit, an option to print/not print flux output after each batch, user capability to specify the number of direct-access blocks allocated on scratch units, compatibility with the year 2000, and reduction of the amount of error output in some cases.

KENO-VI HOLE Input Is Simplified

HOLE input in KENO-VI has been simplified. These changes are significant improvements requested by many users. The HOLE boundary no longer needs to be specified in the unit containing the HOLE. The HOLE boundary is automatically added by the program based on the unit specified in the HOLE record and its ORIGIN and ROTATE data. HOLEs cannot intersect. An example of the original and the new methods for adding HOLEs to a unit is given below. The input data no longer required are highlighted in the old input.

```
********** Old KENO-VI input *************
unit 1
hexprism  10  1.0  10.0 -10.0
media  1  10
boundary  10
unit 2
cuboid    10  6p20.0
hexprism  20  1.0  10.0  -10.0  origin x=5.0 y=3.0  rotate a2=90
media 2  10  -20
hole  1  20 origin x=5.0 y=3.0 rotate a2=90
boundary  10
******** New KENO-VI input ***************
unit 1
hexprism  10  1.0  10.0 -10.0
media  1  10
boundary  10
unit 2
cuboid    10  6p20.0
media  2  10
hole   1  origin x=5.0 y=3.0 rotate a2=90
boundary  10
```

Note that in addition to the lack of a geometry record which defines the HOLE boundary, the HOLE record no longer has a vector definition array. The new version of KENO-VI should be able to read most old input files correctly, but they will take longer to run.

KENO-VI Stability and Speed Improvements

Improvements have been made to KENO-VI since the last Web update to increase the stability and the speed of KENO-VI. To improve the code's stability, logic has been added to KENO-VI to check if a particle is still in the boundary region of a unit when it is no longer in any region. If this occurs, an error message is printed and the program terminates. This situation is often caused by an undefined volume in a unit and could previously lead to the program entering an infinite loop.

To improve the execution speed of KENO-VI, the following modification has been made: When a particle is in an array, the particle is tracked both in the unit where it is currently within the array and in the unit containing the array. It needs to be tracked in the unit containing the array so it knows when it crosses out of the array. Previously, the crossing distance to every surface in that unit was calculated. The code has been changed to calculate only the crossing distance to the surfaces related to the array boundary. This change will significantly reduce the running time of problems where particles spend most of the time in an array or where the array is in a complex unit containing many additional regions unrelated to the array boundary. Running times have been reduced by as much as 15% for arrays contained in complex units.

SAS2H Corrections and Enhancements

A large number of corrections and enhancements have been completed in SAS2H for the release of SCALE 4.4. They are listed below.

- Two errors were corrected for cases where there were more than three zones prior to the mixture 500 zone in the Path B model: (1) The atomic densities were not updated with depleted values in the cross-section processing/spectrum calculations when fuel was input to more than one zone (including the cell-weighted mixture 500 zone). (2) Nuclides that only appear in the moderator were depleted. An example of a model that would be affected is a BWR Path B model with Gd-poisoned fuel pin, gap, clad, moderator, and mixture 500. Usually these discrepancies cause only slight errors in the neutronics part of such BWR cases, but could significantly impact the results for some unique fuel models.

- Input checks and error messages were improved.

- A programming error that caused problems with "MXREPEATS=0" cases to fail on PCs was corrected. These cases are typically used to remove or insert burnable poison rods from one cycle to the next in a depletion.

- A modification was made to correctly calculate the fuel bundle area printed in the shipping cask geometry for the triangular-pitch lattice type of fuel.

- The FUELBNDL input parameter was changed from integer to floating point to allow fractions of fuel assemblies.

- The temperatures of the zones (except the gap) in the Path A model may now be changed for each cycle, similar to the BFRAC and H2OFRAC variables.

- The limit on the total number of libraries (NCYC*NLIB/CYC) was increased to 9,999. However, because the number of unique output file names in SCALE is currently limited to 10,000 and there are typically 11 output files per pass in SAS2H, the practical limit for users is approximately 900 total libraries.

- A significant change was implemented to enable fixed sources (volumetric source or angular flux at a boundary) to be used with INPUTLEVEL=3 cases. This change gives users the capability to model cases such as the irradiation of target materials without explicitly including the irradiation facility in the SAS2H model. Previously the driver geometry and its specific power were required as input to govern the depletion calculation. This fixed-source option is specified in the INPUTLEVEL=3 data as either a volumetric or boundary source. SAS2H then determines the flux based on this fixed source and passes it to ORIGEN-S for use in a flux-driven depletion calculation.

- SAS2H was modified to allow two zones in the Path A model to contain the same nuclide, one at a density of 10^{-20} and the other at a density of greater than 10^{-10}.

- The fixed dimension of 1000 for the Path B mixing table arrays was removed where possible and increased otherwise. The size needed for these arrays can be as large as five times the Path A mixing table size (currently a maximum of about 300 in the 44-group library) plus the number of nuclides outside the zone of mixture 500. The dimension of the arrays that remain fixed was increased to 5000. The remaining arrays were variably dimensioned to the maximum of 2000 or the sum of the Path B mixing table size plus 100 (to allow increases of at least 100 nuclides for INPUTLEVEL=3).

Criticality Search in CSAS1X

MIPLIB has been updated to add input options to MORE DATA that allow specifying an XSDRN adjoint solution, a zone width search, a unit number for the balance table file, and suppressing the cross section weighting. The addition of the zone width search option now gives CSAS1X the capability to perform one-dimensional criticality searches on the size of a geometry zone in XSDRNPM.

New KENO Weights Library and Modules to Generate Weights

Because there was a need to be able to automatically generate a set of weights for use in KENO for arbitrary group structure and material, a new control module GWAS and a new functional module GENWGTS have been added. GWAS sets up an adjoint XSDRNPM case and generates weights automatically from the fluxes. GENWGTS is called by GWAS to read the adjoint fluxes, automatically generate the KENO weighting functions from them, and write an output file for use by program WGT. The biasing weights library for KENO V.a and KENO-VI was updated using the new modules GWAS and GENWGTS. The library contains weights for paraffin, water, concrete, and graphite in 16, 27, 44, 218, and 238 energy groups. The new library was created because there were no biasing data for use with the new ENDF/B-V 44- and 238-group libraries that were released in SCALE-4.3. The old library only contained data for 16, 27, and 123 groups. Note that the 123-group library was removed in SCALE-4.3. Results using this new weights library with the 16- and 27-group cross-section libraries will be different but should agree within statistical uncertainty.

New SCALE Utility Programs

Several new utility programs have been developed for SCALE. A new utility LEGEND has been created that adds a title and legend to the color GIF files generated by KENO V.a or KENO-VI. LEGEND was released last summer with the updated version of KENO-VI (see the June 1996 issue of the Newsletter). The versions of KENO V.a and PICTURE in the next release of SCALE will use LEGEND as well.

K5TOK6 and C5TOC6 are new utilities that convert KENO V.a and CSAS input files to KENO-VI and CSAS6 input files by translating the KENO V.a geometry input to KENO-VI format. Since the converted input files are based on the KENO V.a geometry input, they are generally not the most effective in terms of the

KENO-VI geometry features. They do provide the user with a working KENO-VI input file that can be modified for improvements.

Another new utility is Q0RDPN. It converts a binary input file generated by a CSAS or SAS control sequence for one of the functional modules that use FIDO input such as BONAMI, NITAWL-II, ICE, and XSDRNPM, to an ASCII input file. The user can easily edit the ASCII input file to run a modified version of a problem. This capability allows the user to specify input parameters that are not available in the standard control sequences.

Zirconium Hydride Cross Sections

The ENDF/B-V cross-section libraries in SCALE 4.4 have been updated with thermal scattering data for zirconium hydride. New standard compositions have been added to the Standard Composition Library to allow access to these new cross sections. The new standard compositions are the following:

ZRH2 - density 5.61 g/cc, 1 zirconium to 2 hydrogen atoms
ZR5H8 - density 5.61 g/cc, 5 zirconium to 8 hydrogen atoms
H-ZRH2 - density 1.0 g/cc, the hydrogen in zirconium hydride
ZR-ZRH2 - density 1.0 g/cc, the zirconium in zirconium hydride

SCALE 4.4 Minor Modifications

In addition to the major enhancements noted above, SCALE 4.4 contains many minor modifications, including corrections to errors in SCALE-4.3 and changes to improve portability to different computing platforms. Note that some of these modifications may be duplicate listings of items mentioned in the previous sections.

18-Group Gamma Library: (1) Processed through CORECTOL to mark it as NITAWL-II compatible. Could not be processed by NITAWL-II prior to this correction. (2) Updated to replace the Henderson and Claiborne-Trubey dose factors because the data overestimated the doses by about 25%. The replacement data were taken from the 22n-18g group coupled library.

27-Group Burnup Library: Updated data on rhodium-103 so that Bondarenko factors are generated in the unresolved resonance range. A test case based on 4.5 wt % UO_2 burned to 54,585 MWD/MTU, cooled for 5 years, was run. The calculated k_{eff} increased by 0.06% with the new Rh-103 cross sections.

44-Group ENDF/B-V Library: The 44-group neutron cross-section library was recollapsed from the 238-group library using the corrected version of MALOCS. The impact of the MALOCS corrections should be negligible. See MALOCS corrections below for more information.

238-Group and 44-Group ENDF/B-V Libraries: (1) Corrected negative scattering and total cross sections for minor actinides, fission products, and beryllium metal. Also corrected thermal Bondarenko factors for potassium. Only significant impact should be on cases where potassium is important in the thermal range. (2) Updated to remove resonance parameters from specially weighted stainless steel nuclides because they were being doubly applied. Also, zirconium and hydrogen cross sections for zirconium hydride were added to both libraries.

AJAX: Corrected a portability problem in subroutine ANN caused by the array D being typed real by default, and then printing variables from it using an integer format.

ARP: Updated for optional interpolation on moderator density and made more general to handle user-created basic cross-section libraries. ARP now runs under SCALE driver on PCs and workstations.

ARPLIB: This is a new utility program that creates binary ORIGEN libraries for ARP. It extracts libraries at the desired burnups from large multi-burnup library files generated by SAS2H.

AWL: Added AWL to SCALE to convert AMPX working format libraries between ASCII and binary formats. It is required for the SCALE Criticality V&V package.

BONAMI: (1) Updated to improve error handling procedure and messages. (2) Corrected a problem that caused cases to fail when zero number density input was used.

C5TOC6/K5TOK6: The input file generated for CSAS6/KENO VI incorrectly labeled regions generated to surround HOLEs if there were more than one region in a unit that contained HOLEs. Because of a change in KENO-VI, these regions should no longer need to be generated. Subroutine PUNCH_GEOM was modified to no longer generate these regions.

COUPLE: Updated for year 2000 compatibility, PC version compatibility, uppercase or lowercase input files, and for printing the banner page only when COUPLE is first called.

COUPLE Sample Problem: Updated to change the inner radii in the 3$$ array to zero for consistency with the NITAWL-II input requirements.

CSAS/KENO V.a /KENO-VI/SAS2H Sample Problems: Updated to use the 44-group library.

CSAS and MODIFY: CSAS was updated to add additional required data to the direct access file written for a search problem. MODIFY was updated to read this file. A check for valid parameter constraints and the printing of an error message if they are invalid were also added.

H7MAP: For 1-D problems, if the number of nodes is large enough that the output exceeds one page in length, only part of the output is displayed. The output from the first page is repeated, and the rest of the output is never printed. Correcting this problem involved simply moving one statement from within a DO loop to a point before the DO loop.

H7TECPLOT and H7MONITOR: Outdated comment lines in the BLOCK DATA subroutine that are used to activate or deactivate computer-system-dependent blocks of code resulted in memory not being allocated for variably-dimensioned arrays. An additional correction was made in H7TECPLOT, where the x and y axes were reversed when a translation was done from spherical to Cartesian coordinates.

HEATING Sample Problems: The input file for the second HEATING sample problem was modified to first compile and run a simple Fortran program to convert an ASCII node connector file to binary format for use by HEATING. This modification improves installation portability on different Unix workstation platforms.

KENO V.a: (1) Updated subroutine RDPLOT to correct the format used to print the error message for incomplete input data. (2) Corrected variable type in format statements for debug prints. This discrepancy causes problems on some systems, including PCs when debug print is turned on (DBG=YES). (3) Changed default plot type to color. (4) Updated to correct an error in the k_{eff} calculation that caused a doubling of k_{eff}

when using an ICE mixed AMPX format working library. This error was introduced in SCALE-4.3. (5) Updated to allow printing the frequency distributions for 1-group problems. (6) Updated to match KENO-VI with respect to matrix calculations. The calculation of lifetime was corrected because it was not based on a fair game. These changes can cause the lifetime to be substantially different. The error in the lifetime calculation has probably been in KENO V.a since its initial release in SCALE-3.

KENO-VI: (1) Updated to correctly number error messages, replace the word PICTURE with the word PLOT throughout the program, and print plot symbol data only for character plots. (2) Updated subroutine TRACK to correctly sum fluxes. The fluxes didn't sum properly for units that were crossed by an array boundary. (3) Enhanced to allow HOLEs to be used without explicitly defining a geometry region where the HOLE was to be inserted. The code automatically adds to the unit containing the HOLE the equations that define the boundary of the unit contained within the HOLE, properly rotated and translated as specified on the HOLE record. (4) Fixed problem writing restart file on Sun workstation. (5) Modified the subroutine GEOMIN to correct an infinite loop problem. A pointer to the array that contained the unit boundary x, y, and z position was improperly specified. The pointer LBOXGM has been respecified. (6) Corrected a problem where a particle's inability to cross an array boundary due to round-off problems caused an infinite loop. (7) The code was updated to correct a problem that could cause cases containing arrays with complex boundaries to incorrectly calculate k_{eff}. (8) Corrected a discrepancy that caused the code to go into an infinite loop when boundaries consisted of a body with multiple sets of paired planes. (9) Corrected an error that prevented a restart problem from producing a readable file if it stored data in the generation before the code entered the infinite loop. (10) Corrected a problem involving nested arrays and hexprisms that sometimes caused the code to go into an infinite loop if a collision occurred very near a boundary. (11) Corrected a problem that occurred when a particle crossed a boundary and immediately had a collision that reversed its direction without traveling any distance. The particle sometimes got lost and entered an infinite loop. (12) Modified subroutine TRACK to correct a problem that occurred when an array shared a boundary with a hole that contained the array. If the distance to cross out of the array is less than EPS, the particle now exits the array instead of crossing from one unit to another within the array. (13) Corrected an error in placement of starting points for start type 6. (14) Corrected an error in the flux calculation for regions containing holes or arrays. (15) Corrected a roundoff problem with arrays offset a long distance from the origin. This problem could sometimes cause an infinite loop. (16) Corrected a variable that was misnamed and, as a result, was used without being initialized. (17) Set a lower limit for the calculated crossing tolerance to prevent the code from entering an infinite loop. Also made minor changes to the particle-tracking output when parameter TRK=YES. (18) Updated to allow starting points in a volume larger than the global unit. (19) Updated to terminate a problem if a particle in subroutine TRACK gets lost. Also, updated to allow problems that contain array data but do not reference the arrays in the GEOMETRY data block to run. (20) Updated to change the logic in calculating the array boundary crossing distance (decreases running time for some problems) and to change the default plot type to color. (21) Updated to correct a problem with non-cuboidal albedo boundaries and to add additional space for matrix data. The standard deviations for average k-effective by generation skipped are now accumulated in batches. Because of these changes, any matrix information and the table of average k-effective by generation skipped in the sample problem output will be different. (22) Corrected tracking to allow simultaneous crossing of multiple shared boundaries and to correctly sum fluxes after a collision. Also corrected error related to calculating the x-offset of an array. Changed input logic for ORIGIN and ROTATE data to sum values for an auxiliary keyword for a given geometry record rather than use the last value. This last change was made for compatibility with C5TOC6. (23) Updated to allow a particle to cross from one hole directly into an adjacent hole even if the crossing is outside the allowed tolerances. This prevents some cases from entering an infinite loop. (24) Modified to detect intersecting HOLEs in the global unit. A problem will now terminate if intersecting HOLEs are detected in the global unit. Intersecting HOLEs are illegal in KENO-VI geometry but were not detected in the global unit. Intersecting HOLEs in units other than the global unit are detected during tracking of particles through the intersecting regions.

KENO-VI Sample Problems: Sample problem 22 has been altered in the KENO-VI input file. The geometry data were changed to take advantage of the simplified method of adding HOLEs.

KMART: This new module was added to allow post processing of a KENO V.a restart file, along with a working format cross-section library, to generate activities and/or broad group fluxes and compute the fission production activity if the components are available in the working cross-section library for the requested nuclide. A resonance self-shielded value is used for the fission cross section.

MALOCS: (1) An error was corrected in weighting a coupled master library using a neutron spectrum from a neutron library combined with an explicitly specified gamma-ray spectrum. Also introduced several options for truncating upscattering terms. Changes were made to properly weight the delayed and prompt values of ν. (2) A discrepancy was corrected that caused the storage of invalid data in the temperature array. In the 44-group library this caused the data for the third temperature to be overwritten and to be used for a temperature that is effectively zero degrees Kelvin.

MARSLIB: (1) Updated to change the value of epsilon used to check for round-off errors in the geometry and, thereby, reduce the number of such errors. This modification eliminated the errors previously experienced with several of the SCALE Shielding V&V problems. (2) Variables IR in subroutine AZIP and IRET in subroutine UNIS are now initialized to 0 before they are used as arguments to function IREAD. In AZIP and in UNIS a 'CALL EXIT' was changed to a 'STOP'. In subroutine ALBERT, the nH was removed from two formats and replaced with quotes.

MIPLIB: (1) Updated to allow moderator mixture in a lattice cell to be used in the gap and to add the ability to specify the inner radius to the resonance data. (2) Updated to allow a control program to suppress certain output by setting flags. Added input options to MORE DATA to allow specifying an XSDRNPM adjoint solution, a criticality search in XSDRNPM using the zone width search option, a unit number for the balance table file, and suppressing the cross section weighting. (3) Corrected an error allowing the input of a number density for a compound or alloy. This error was introduced in SCALE-4.3. (4) Updated to allow number density input for an element that has multiple isotopes.

MIPLIB, SUBLIB, UNIXLIB, COMPOZ, MODIFY: Updated to use new direct access routines for character data and replaced references to specific intrinsic FORTRAN functions with their generic names for Fortran 90 compatibility. Also corrected an error in the Dancoff factor calculation that occurs for cylinders in a MULTIREGION problem. This error results in an error in the calculated k_{eff} value of approximately 0.1% for a cylinder the size of a typical fuel rod. Note that this error did not occur in the LATTICECELL geometry option.

MORSE: (1) Updated the limit on number of tracking errors, the unit number for surface detector results, and increased dimensions on surface detector arrays. (2) Updated to include changes to surface detectors for SAS4 cases, to correct a problem in DIREC for NDSG=17 case, to allow orderly termination of a problem when errors in tracking to detector exceed a limit, to add an option to print/not print flux output after each batch, to input the number of direct-access blocks allocated on scratch units, to change the way date is output (to handle the year 2000 and beyond) and to reduce the amount of error output in some cases.

MORSE Sample Problem 8: The 10** array was modified by adding a 22r0.0 at the end.

NITAWL: Corrected the potential cross section used for higher order resonances (L>0). The impact should be negligible in most cases.

ORIGEN-S: (1) Updated cross-section edit of binary libraries to add option to change cross-section values to quantities derived from total flux (as in ORIGEN2) instead of thermal flux. (2) Corrected calculation of printed average power. (3) Added error message if number of time steps is less than 4 for reactor startup case. (4) Updated to correct the loop index for re-normalizing the R8 array. (5) Updated for year 2000 compatibility and to correct calculation of He-3 and H-3 for long time steps and high flux. (6) Updated to allow saving concentrations and then continuing with a subcase using a new library. (7) Updated to allow the flux input value for the last time step to be zero.

ORIGEN-S Master Photon Library: The library was updated to correct the photon yield data for Ra-222 and Th-226, and the photon yields for gammas accompanying (α,n) and spontaneous fission reactions were updated to reflect small changes that occurred during the last decay data update.

OSBICO/OSBIRE: Updated for compatibility with lastest version of ORIGEN-S.

PERFUME: Improved the selection of new moments when a moment is found to be invalid and converted coding to a more standard Fortran 90.

PERFUME Sample Problem: The special cross-section data file required for the PERFUME sample problem has been added to SCALE, and the sample problem input data have been updated to use it. This problem has not been included in SCALE since SCALE was moved from the mainframe to the workstation several years ago.

PICTURE: (1) The module was updated to add option of generating 2-D color GIF plot files of the geometry model input for the SCALE shielding modules MORSE or QAD-CGGP. This capability already exists in the SCALE criticality modules KENO V.a and KENO-VI. (2) The module was also updated to handle the call by SAS4 when the "PARM=CHECK" option is used. (3) An option was added that allows users to specify an X-Y, X-Z, or Y-Z plot and have the code automatically calculate the cosines used for the plot.

PRISM: This is a new utility program for ARP that can read a single SAS2H or other type of input file and generate multiple copies by replacing generic symbols with specified values.

QADS/QAD-CGGP: (1) Updated to make the combinatorial geometry input data have the same format as the combinatorial portion of the MARS geometry input which is used in other SCALE modules. Old input files will no longer run. (2) Updated to add error checks for limits on number of compositions and elements and to fix the code to handle upper- or lower-case input. (3) Added a test on the MIPLIB error flag that terminates execution of the problem if an error occurred.

QADS and QAD-CGGP Sample Problems: Updated to change the geometry input format to agree with the changes made to QADS and QAD-CGGP.

RADE: Corrected an error in subroutine MCHEK that caused RADE to fail on a Sun workstation. A constant was passed as an argument to subroutine MCHEK to be used for dimensioning, but MCHEK later used the same variable for other purposes. The argument was renamed and used in the dimension statement.

SAS1: Scratch unit N16 was not opened when SCALE driver returned to SAS1 after cross-section processing and prior to XSDRNPM shielding calculation. This problem caused SAS1 to fail on the PC. The OPEN statement was moved to the beginning of main program so it would always be opened.

SAS2H: (1) Updated to fix a problem where the reload feature failed to reload correctly for the final cycle type. (2) Modified subroutine SZNSEG so that it would not cause the ORIGEN library creation to fail by not recognizing the cross-section library specified. The problem was an uninitialized variable ERSET. The change was to initialize the variable as "FALSE" before calling subroutine GETLIB. A change was also made so that the library name was passed to GETLIB instead of only the first 4 characters. (3) Updated to correct an error in the mass of the clad when the clad was input as an isotope and the mass was not input as a light element in Data Block 15. (4) A problem was corrected where the atomic densities were not updated with depleted values in the cross-section processing/spectrum calculations when fuel was input to more than one zone (including the cell-weighted mixture 500 zone) and there were more than three zones prior to the mixture 500 zone. For example, consider a BWR Path B model with Gd-poisoned fuel pin, gap, clad, moderator, and mixture 500, where there are four zones prior to the mixture 500 zone. Usually this discrepancy causes only slight errors in the neutronics part of such BWR cases, but could significantly impact the results for some unique fuel models. (5) The module was updated for compatibility with the newest revisions to XSDRNPM. The routines that wrote the XSDRNPM input files needed to be changed to account for the changes to XSDRNPM input. Subroutine COPYNX had to be changed to add the 0$ array to the XSDRNPM input file, and to move setting the logical unit number of the flux output file from the 2$ array to the 0$ array.

SAS3: (1) Variable IR in subroutine OAKTRE is now initialized to 0 before it is used as an argument to function AREAD. Subroutine RINPUP was updated to initialize the variables JMK and IML in COMMON JOMK because they are used when SAS3 calls MARSLIB routines and they were not being defined prior to the calls to JOMIN. (2) Updated to be compatible with the new MORSE input options and to implement the PARM=SIZE parameter which was not being passed to MORSE.

SAS4: (1) Subroutine MORINP was updated to add common JOMK and to initialize the variables JMK and IML in common JOMK because they are used when SAS4 calls MARSLIB routines and they were not being defined prior to the calls to JOMIN. (2) The code was modified to translate the user input to lowercase. This change was necessary to make SAS4 capable of handling input files in either upper or lowercase, as the other SCALE neutronic codes already do. (3) The code was updated to correct the dimensions on two arrays. Also changed a test comparing 2 floating point variable names equivalenced to integer variables to use function ISET. (This test has previously caused floating point underflows on some platforms.) (4) Added a "PARM=CHECK" option that calls PICTURE from within the SAS4 to plot geometry but not run MORSE. SAS4 prepares or reads MARS geometry input data, reads PICTURE input, and calls PICTURE. Several other changes were made to error messages and formats. (5) The convergence criteria was updated because the criteria in XSDRNPM were changed. (6) The input variable NDAB was added to allow the user to specify the number of direct access blocks allocated.

SAS4 Sample Problems: (1) A ninth sample problem was added to illustrate the new enhanced surface detector option. (2) SAS4 sample problems 1, 3, and 5 were updated to remove references to variables FR1, FR2, FR3, and FR4, which became obsolete in SCALE 4.4.

SCALE Driver: (1) Updated to allow processing the rest of the input data after invalid input data are detected. (2) The driver has been updated to obtain and act on error codes from the modules. The driver now prints error codes and stops further sequence execution.

Standard Composition Library: (1) The default density of B4C was corrected from 2.54 to 2.52 g/cc. This error was introduced in SCALE-4.3. For an LWR fuel problem with B_4C pins between fuel assemblies, the calculated k_{eff} value increased less than 0.2%. (2) Updated to reference the nuclides used for zirconium

hydride which have been added to ENDF/B-V libraries and to add four new standard composition names related to zirconium hydride. (3) The densities for SS304 nuclides were made identical to the standard versions of the same nuclides. (4) Updated the standard composition ZIRC2 for consistency with current technical standard and updated densities for SS304 and SS316. (5) Mass of copper was corrected (it was in atomic mass units instead of C-12 mass units). Density of C-GRAPHITE was changed from 1.0 to 2.3 g/cc. The following compositions were added: GRAPHITE, KEROSENE, KERO(H2O), NORPAR13, NORPAR(H2O), POLYVINYLCL, PVC, PVC(H2O), TBP, TBP(H2O).

SUBLIB/UNIXLIB: (1) Updated to remove year 2000 problems. These changes basically changed the year format for the QA verification table to 4 digits. Additionally, the date format was changed to use a 3-character month abbreviation so that the date would be unambiguous. A new line was added to the QA verification table printout to identify the machine on which the program was run. (2) Updated to remove an artificial limit of 8-character-length filenames for non-standard files in subroutine OPNFIL. (3) Modified subroutines LISTQA and VERGET for consistency of the length of the string containing the executable name, the creation date, and the directory path to the executable. The directory path was increased to 256 characters. (4) Updated subroutine FINDQA to place underscores in place of the blanks in the date to simplify the automatic updating of the QA verification table. (5) Replaced the CHARACTER*8 type of variable CAT with a variable length CHARACTER type in subroutine NOTE. This corrected a problem in WAX on the Sun workstation. (6) Added comments to subroutine OPENDA indicating how to replace the Fortran 90 specific INQUIRE statement with a Fortran 77 compatible statement. (7) Replaced all STOP statements with calls to EXIT with the appropriate error return code for proper detection by the driver. (8) Modified subroutine DREAD to correctly process data following the second digit of an exponent when called by the array reading subroutine YREAD. Previously, exponents of 10 or greater sometimes caused errors in the reading of FIDO-type input arrays. This discrepancy was discovered in an ORIGEN-S case. (9) Changes to update XSDRNPM required a double precision ERF function. The necessary routines were added to compute the double precision ERF when it is not part of the intrinsic library.

XSDOSE: An option was added to turn off angular flux print and no angular flux print was made the default.

XSDOSE Sample Problem: The input was modified to turn on the new angular flux print option in XSDOSE.

XSDRNPM: (1) The special activity file and balance table file were not written correctly, and the correct file structure is not what was documented. Subroutine SETUP was changed such that it would not read or write dummy records after the files were opened. These read/writes were the only way to open the files before Fortran 77, but when the code was converted to Fortran 77 and OPEN statements were added to explicitly open the files, the extra statements were not removed. (2) Updated to correct the accumulation of zone fluxes when inner-cell weighting is selected. (3) Updated to correct the value of productions/absorptions when a direct buckling search is done. (4) The code was also modified to collapse prompt υ and delayed υ using the same procedure as used to collapse the total υ. (5) Corrected calculation of broad group balance tables to be consistent with fine group tables. Broad group cross sections were not in balance when upscatters were collapsed. (6) Updated to print clearer messages when allocated memory is insufficient. Also, in these cases if an output file could not be written, any previously existing file was deleted to prevent subsequent calculations from reading it. (7) The Fortran source for XSDRNPM was converted to Fortran 90 free format. (8) The input/output units were all moved to the 0$ array. The energy of the average lethargy causing fission was added to the balance tables. (9) The output files from the balance tables and the activities were modified and converted to ASCII files. A new file was created which contains the input and derived data from a problem. The flux file was changed to double precision. (10) The code was modified to not run with fluxes out of core unless explicitly requested in the input. (11) The coarse mesh generation algorithm used in rebalancing the

inner iterations was modified to correct a problem that prevented convergence of a very small class of problems.

XSECLIST: This is a new utility program for ARP which prints lists of absorption and fission cross sections vs burnup for nuclides from ORIGEN-S multi-burnup binary libraries.

Portability

Version 4.4a of the SCALE system has been developed to ensure portability among various computing platforms. The system is maintained and enhanced at ORNL under quality assurance and configuration management plans. The system has been routinely tested on IBM and DEC workstations. In addition, a version for personal computers (PCs) is included in the package. The PC version runs on Windows 95, 98, and NT4.0 and platforms. The system also has been installed and tested by ORNL on SUN and HP workstations. Information needed to install and run SCALE on each of these systems is included in README files with the software package distributed by the code center.

Related developments

The definition of "easy-to-use" has changed considerably since the late 1970s. As funding has allowed, the ORNL development staff has sought to develop user interfaces that provide a distinct aid to novice or occasional users of the system.

The ORIGNARP input processor is a MS-DOS PC program designed to assist a user in creating an ORIGEN-S input file. It is coupled with the ARP code, which interpolates on standard LWR ORIGEN-S binary libraries, in the ORIGEN-ARP system (Section D1).

CSPAN (Criticality Safety Input Processor for Analysis) is the Windows GUI replacement for the CSASIN input processor for the CSAS criticality sequences in SCALE. CSASIN was an MS-DOS program developed in 1990–91 to assist new and occasional SCALE users. Because CSASIN is incompatible with Pentium II and later PCs, a new easier to use and more powerful Windows program has been developed. CSPAN can be used to read and modify an existing SCALE input file or to create a new input file. CSPAN can call SCALE to execute CSAS using the input file it creates. The SCALE Standard Composition library and the selected SCALE cross-section library are read by CSPAN and the user is only allowed access to those compositions available on the selected cross-section library. The program handles the entry of basic standard compositions, solutions, and arbitrary materials, unit cell data, optional parameter data, and KENO V.a input data. CSPAN can call SCALE to execute any CSAS case. CSPAN runs under Windows 95, 98, or NT. Checks for errors are included throughout the program to verify that the input is valid. The initial version distributed with SCALE 4.4a is considered a beta test version. Help files have not been developed yet, but will be made available soon.

The initial version of a Windows-based GUI for HEATING named Visual Heating is also included in the SCALE 4.4a release. Visual Heating assists the user in preparing a HEATING input file and includes a 3-D graphics display of HEATING geometry models using OpenGL. Visual HEATING can execute the HEATING case in SCALE and display the output file in a text editor. It includes an HTML Help system similar to many commercial Windows programs. The help system is accessible both from the main menu bar and by pressing the F1 key. Most of the information in the HEATING User's Manual (Sect. F10 of the SCALE Manual) is included in the help system along with explanations of Visual HEATING input screens.

Availability

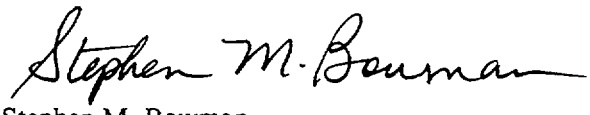
The SCALE code system and the other software designated under "Related Developments" have been packaged by the Radiation Safety Information Computational Center (RSICC). The SCALE system and the related software may be obtained by contacting

Radiation Safety Information Computational Center
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6362
Telephone: (865) 574-6176
FAX: (865) 574-6182
E-mail: rsic@ornl.gov
Internet: http://www-rsicc.ornl.gov

Acknowledgments

The SCALE system is maintained at ORNL and enhanced to keep pace with normal technical advancements in the analysis areas of interest. Although the NRC continues its role as the controlling sponsor of the SCALE system, the U.S. Department of Energy (DOE) began assisting in the maintenance of the SCALE system in 1987. Over the years numerous individuals within these sponsoring organizations have played key roles in ensuring that the SCALE system remained a readily available, reliable system for the analysis of nuclear fuel facilities and packages. The individuals who have worked with the ORNL staff to coordinate maintenance and development activities include R. H. Odegaarden (NRC, ret.), G. H. Bidinger (NRC, ret.), C. Mauck (DOE, ret.), E. P. Easton (NRC), W. H. Lake (NRC and DOE), M. E. Wangler (DOE), and M. G. Bailey (NRC).

As demonstrated by this Manual, there are also numerous individuals from the ORNL staff who have contributed significantly to the development and enhancement of the SCALE system. Most are credited by their authorship of the sections in this Manual that correspond to their work. A few individuals have been essential to the development and maintenance of SCALE but are not credited by authorship. These individuals include: S. K. Lichtenwalter, who is responsible for implementing and controlling software system changes; C. H. Shappert, who provided the editorial review of this Manual; and L. F. Norris (ret.) and W. C. Carter, who prepared the entire manuscript. Special acknowledgement is also due to R. M. Westfall and G. E. Whitesides (ret.) who, together with R. H. Odegaarden of the NRC, developed the concept and long-range goals of the SCALE system in the late 1970s. Finally, this Project Leader will always be grateful to C. V. Parks, who served as the SCALE Project Leader for the first 15 years, and L. M. Petrie, who for 20 years has consistently provided consultation and advice on the technical direction that should be taken in development of nearly every module and cross-section library that are in the present system.



Stephen M. Bowman
SCALE Project Leader
December 1999

References

1. N. M. Greene, W. E. Ford III, L. M. Petrie, and J. W. Arwood, *AMPX-77: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Cross-Section Libraries from ENDF/B-IV and/or ENDF/B-V*, ORNL/CSD/TM-283, Martin Marietta Energy Systems, Inc., Oak Ridge National Laboratory, October 1992.

2. N. M. Greene, J. W. Arwood, R. Q. Wright, C. V. Parks, *The LAW Library–A Multigroup Cross-Section Library for Use in Radioactive Waste Analysis Calculations*, ORNL/TM-12370, Martin Marietta Energy Systems, Inc., Oak Ridge National Laboratory, August 1994.

3. M. D. DeHart and S. M. Bowman, *Validation of the SCALE Broad Structure 44-Group ENDF/B-V Cross-Section Library for Use in Criticality Safety Analyses*, NUREG/CR-6102 (ORNL/TM-12460), U.S. Nuclear Regulatory Commission, Oak Ridge National Laboratory, September 1994.

Table 1 Analysis capabilities summary of the SCALE control modules

Control module	Analysis function(s)	Functional modules executed	Section reference
CSAS	1-D deterministic calculation of neutron multiplication 3-D Monte Carlo calculation of neutron multiplication Problem-dependent cross-section processing Multiplication search or spacing	BONAMI NITAWL-II XSDRNPM KENO V.a ICE	C4
CSAS6	3-D Monte Carlo calculation of neutron multiplication	BONAMI NITAWL-II XSDRNPM KENO-VI	C6
ORIGEN-ARP	Point depletion/decay of nuclear fuel and radioactive material	ARP ORIGEN-S	D1
SAS1	1-D deterministic calculation of radiation transport through shield and dose evaluation at a point Calculation of dose at detector based on leakage from critical volume	BONAMI NITAWL-II XSDRNPM XSDOSE	S1
SAS2	Point depletion/decay of nuclear fuel 1-D radial shielding analysis in cylindrical geometry	BONAMI NITAWL-II XSDRNPM COUPLE ORIGEN-S XSDOSE	S2
SAS3	Dose evaluation using MORSE Monte Carlo code	BONAMI NITAWL-II XSDRNPM MORSE-SGC	S3
SAS4	Calculation of dose outside of transportation package using MORSE code and automated biasing techniques	BONAMI NITAWL-II XSDRNPM MORSE-SGC	S4
QADS	3-D point-kernel gamma-ray shielding analysis	QAD-CGGP	S5
HTAS1	R-Z steady-state and transient analyses of a transportation package	OCULAR HEATING	H1

Table 2 Analysis capabilities summary of the SCALE functional modules

Module	Function	Section reference
BONAMI	Resonance self-shielding of cross sections with Bondarenko factors	F1
NITAWL-II	Resonance self-shielding of cross sections with resolved resonance data	F2
XSDRNPM	General 1-D, discrete-ordinates code for: • zone-weighting of cross sections • eigenvalue calculations for neutron multiplication • fixed-source calculation for shielding analysis • adjoint calculation for determining importance functions	F3
XSDOSE	Module for calculation of dose at a point based on the 1-D leakage flux from a finite shield	F4
COUPLE	Interface module for preparation of cross-section and spectral data for ORIGEN-S	F6
ORIGEN-S	General-purpose point-depletion and decay code to calculate isotopic, decay heat, radiation source terms, and curie levels	F7
ICE	Cross-section utility module for mixing cross sections	F8
MORSE-SGC	Monte Carlo code with combinatorial and array geometry features used to perform radiation shielding analysis	F9
HEATING7.2	Finite-volume, multidimensional code for conduction and radiation heat transfer	F10
KENO V.a	Monte Carlo code for calculation of neutron multiplication factors	F11
OCULAR	Calculation of radiation exchange factors	F16
KENO-VI	Monte Carlo code for calculation of neutron multiplication factors for complex geometries	F17

NUREG/CR-0200
Revision 6
Volume 2, Section F16
ORNL/NUREG/CSD-2/V2/R6

Computational Physics and Engineering Division

OCULAR: A RADIATION EXCHANGE FACTOR COMPUTER PROGRAM

C. B. Bryan*
G. E. Giles

Date Published: March 2000

Prepared for the
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555
Under Interagency Agreement DOE 1886-C000-9B
NRC JCN No. B0009

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
managed by
LOCKHEED MARTIN ENERGY RESEARCH CORP.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-96OR22464

*Formerly with Oak Ridge National Laboratory.

ABSTRACT

OCULAR is a radiation exchange factor computer code compatible with the axisymmetric r-z geometry of HEATING. The code may be used to generate graybody exchange factors, as well as geometric configuration factors, which describe the radiative exchange in the cylindrical or annular enclosures that are characteristic of an axisymmetric geometry.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Funding for this project was provided originally by the Storage and Transport Systems Branch (now the Spent Fuel Project Office), Office of Nuclear Material Safety and Safeguards, of the U.S. Nuclear Regulatory Commission. A portion of this work, the capability to model cylindrical enclosures, was funded by A. A. Khan of the Y-12 Development Division.

The author (C. B. Bryan) is indebted to L. M. Petrie for the use of the SCALE free-form reading routines. Gratitude is extended to C. V. Parks for his constructive reviews of the manuscript.

Special gratitude is extended to R. V. Arimilli of the University of Tennessee for providing insight into the phenomenon of thermal radiation heat transfer.

F16.1 INTRODUCTION

The radiative exchange factor computer program OCULAR was developed for the U.S. Nuclear Regulatory Commission (NRC) as part of their efforts to develop standardized methods for performing safety analyses of nuclear fuel shipping containers. The program is a functional module within the SCALE system.

Frequently, nuclear fuel shipping containers are modeled as an assembly of right circular cylinders. Historically, standardized thermal analyses of these containers have been performed using the axisymmetric geometry (type R-Z) capability of HEATING (see Sect. F10).

Version 6.0 of the HEATING code provided the analyst with the capability of modeling one-dimensional (1-D) radiant heat transfer. Radiation is present in the cylindrical and annular enclosures characteristic of the axisymmetric geometry. The HEATING program was enhanced with the release of version 6.1 to provide the capability of modeling multidimensional radiant heat transfer. Exchange factors defining the radiative interaction between enclosure surfaces must be supplied to HEATING as input.

OCULAR is designed to calculate graybody exchange factors for the cylindrical and annular enclosures characteristic of an axisymmetric geometry. The geometric configuration factor (view factor) is the fraction of energy leaving blackbody surface *i*, which arrives at blackbody surface *j* directly. For any blackbody surface *i* in the enclosure, conservation of energy requires that the sum of the configuration factors from surface *i* to itself and the remaining enclosure surfaces equal 1.0. The exchange factor is the fraction of the energy emitted by graybody surface *i* that is absorbed by graybody surface *j* directly and by all reflections. Similarly, for any graybody surface *i* in the enclosure, conservation of energy requires that the sum of the exchange factors from surface *i* to itself and the remaining enclosure surfaces equal the emissivity of surface *i*. OCULAR calculates the configuration factor between the two surfaces analytically. The exchange factors are calculated using Hottel's Matrix Method (Ref. 1).

The input data have been designed to be compatible with that of the other functional modules within SCALE. OCULAR uses SCALE's free-form reading subroutines (see Sect. M3). The input data are subdivided into data blocks that are identifiable by keywords.

The OCULAR input data have been developed with the intention that OCULAR can read and process the HEATING input data. OCULAR is geometrically compatible with the axisymmetric geometry of HEATING, and its output may be directly input to HEATING. The OCULAR input data contain additional data blocks required to describe the enclosure of interest. Thus, a user may develop the HEATING input data to model a heat transfer problem, include the additional data blocks required to perform the exchange-factor calculations, execute OCULAR, and then use the data set coupled with the exchange factors for the HEATING thermal analysis.

Essentially every array whose size is a function of the input data is variably dimensioned. The default values used to compute these variable dimensions may be overridden using input data, and thus the code does not have to be recompiled each time a change in array size is specified. OCULAR uses the SCALE computer memory allocation feature (see Sect. M2).

OCULAR will read the input for a case until it encounters an end-of-case indicator. The program checks the input data for errors and inconsistencies and issues messages identifying any data problems that may have been encountered. Input processing is continued but the problem is not executed. This procedure results in the code identifying all or most of the input data errors with one execution. If no input data errors are encountered, OCULAR will proceed with the specific calculations for the case.

OCULAR solves for the graybody exchange factors (or blackbody configuration factors) for the nodal mesh employed by HEATING. The emissivity of the enclosure surfaces may be positionally

dependent. The positional dependence may be specified through the use of analytical functions, tabular functions, or a user-supplied subroutine. Multiple surfaces and materials may be used to define an enclosure. The surface definitions must be consistent with the axisymmetric geometry. Calculations may be performed for multiple enclosures. Nodal spacing need not be equal.

OCULAR output includes the node numbers i and j, the exchange (or configuration) factor between i and j, and the sum of the factors for each emitting node i. This output is provided for verification purposes. Additional binary output is generated which will serve as unformatted input to HEATING through the CONNECTOR data block.

OCULAR has been developed for the axisymmetric geometry. In the course of the development, efforts have been made to make the code easily adaptable to the other HEATING geometry types.

F16.2 NUMERICAL TECHNIQUE

F16.2.1 Statement of the Problem

The HEATING program offers twelve possible geometries and currently OCULAR is designed to analytically calculate graybody exchange factors for the annular and cylindrical enclosures characteristic of an r-z axisymmetric geometry. The code has been developed with a general format to facilitate future inclusion of the other two-dimensional (2-D) and three-dimensional (3-D) geometries.

As stated previously, OCULAR is geometrically compatible with the r-z axisymmetric geometry of HEATING. The nodal mesh developed by OCULAR is identical to that of HEATING. Thus, OCULAR solves for the radiative exchange factors for nodes that lie on enclosure surfaces using HEATING's node-numbering convention.

HEATING employs a finite-difference mesh generation technique to develop an approximation to the physical problem. In order to define the nodal mesh, a system of orthogonal curvilinear coordinate surfaces is superimposed on the problem. The surfaces may be unevenly spaced, but must extend to the outer boundaries of the problem. A typical node is defined at the intersection of any three surfaces. For an r-z axisymmetric geometry, a node may lie at the intersection of four different materials. The axisymmetric geometry of interest is illustrated in Fig. F16.2.1.

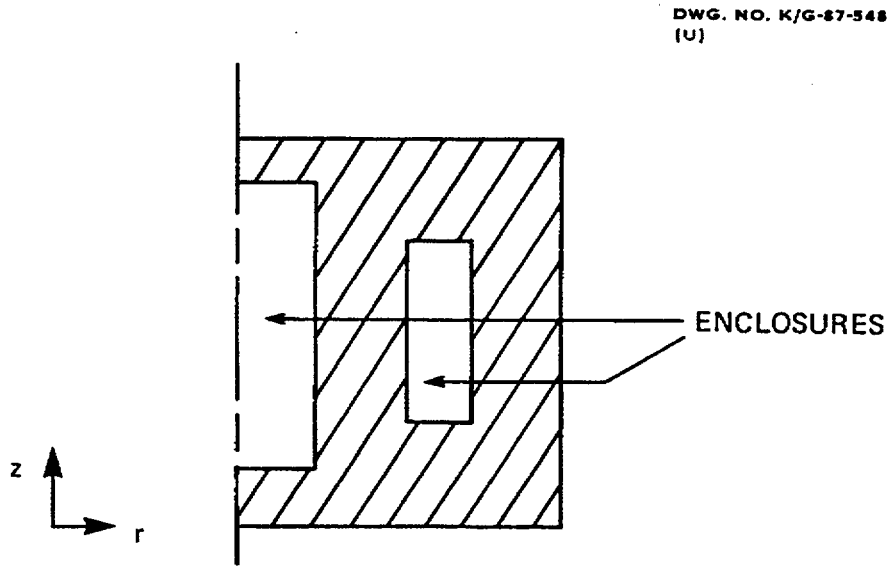


Figure F16.2.1 OCULAR axisymmetric geometry

The enclosure within the finite-difference mesh may be characterized for an axisymmetric geometry as a series of finite-length cylindrical surfaces or annular rings. Each node has an associated area in the radial and axial directions. With reference to Fig. F16.2.2, these nodal areas are defined by:

$$A_r(i,k) = \pi\left[\left(\frac{r(i+1) + r(i)}{2}\right)^2 - \left(\frac{r(i) + r(i-1)}{2}\right)^2\right] , \qquad \text{(F16.1)}$$

$$A_z(i,k) = \pi r(i)(z(k+1) - z(k-1)) . \qquad \text{(F16.2)}$$

OCULAR calculates exchange factors between areas associated with nodes that lie on defined enclosure surfaces.

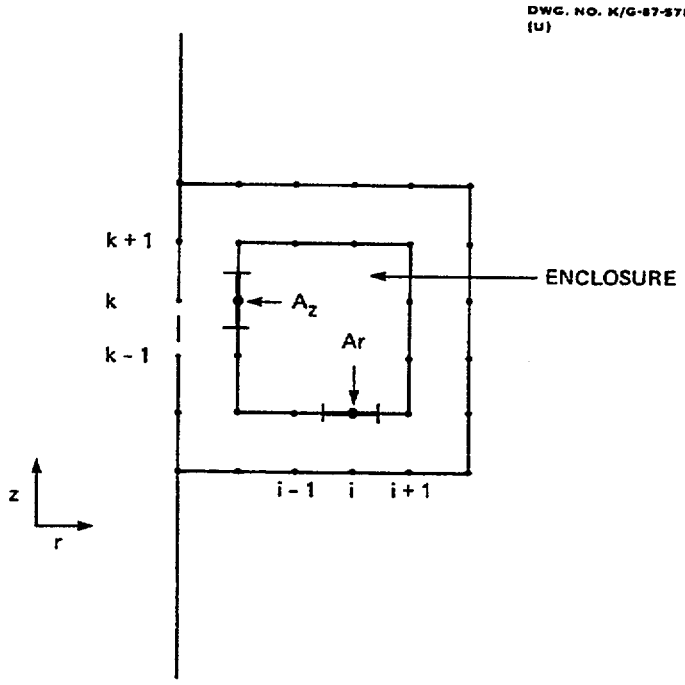


Figure F16.2.2 Nodal description for axisymmetric geometry

F16.2.2 Enclosure Definition

An enclosure is defined as a closed set of surfaces that lie within the model boundaries. OCULAR has been developed to model enclosures characteristic of the r-z axisymmetric geometry, with the capability of extension to other geometries based on this enclosure definition. A general OCULAR model is illustrated in Fig. F16.2.1.

The following conventions are incorporated into the program:

1. multiple enclosures may be present in a model,

2. each enclosure is defined by surfaces that must be parallel to the coordinates axes,

3. each enclosure may be composed of multiple surfaces,

4. the nodal areas on parallel surfaces need not be equal,

5. surface boundaries must correspond to fine gridline locations,

6. the emissivity of a surface may vary with position,

7. the emissivity of a nodal area must be constant across the area,

8. the emissivity of a nodal area composed of different enclosure surfaces is an area-weighted emissivity, and

9. the enclosure may be bounded by multiple materials.

The annular and cylindrical enclosures modeled by OCULAR are illustrated in Figs. F16.2.3 and F16.2.4. The components of these enclosures may be described as a cylinder, lower ring, upper ring, and shell. Each component may be subdivided into smaller components through the use of the variable spacing capabilities of the mesh generator.

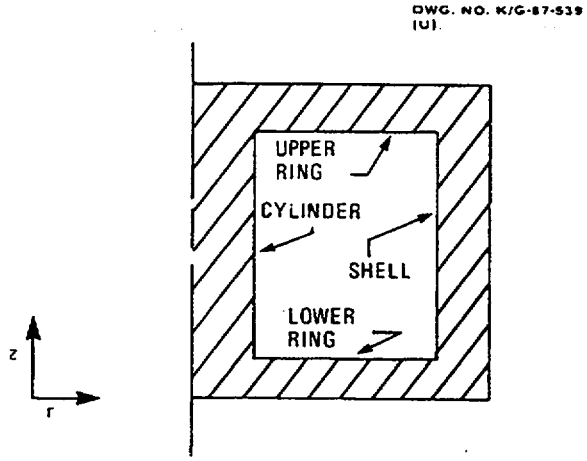


Figure F16.2.3 Annular axisymmetric enclosure

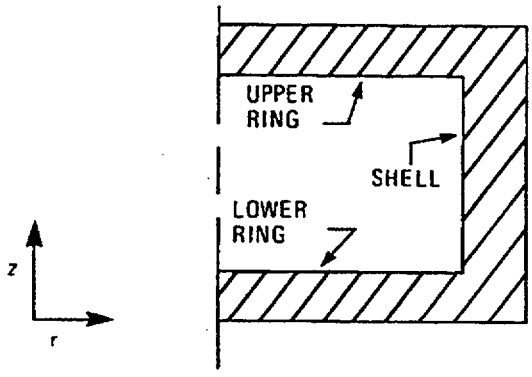


Figure F16.2.4 Cylindrical axisymmetric enclosure

Direct radiant exchange in the annulus is described by interaction between the following:

1. the cylinder and the lower ring,

2. the cylinder and the upper ring,

3. the cylinder and the shell,

4. the lower ring and the shell,

5. the upper ring and the shell,

6. the lower ring and the upper ring, and

7. the shell and itself.

The analytic expressions for the geometric configuration factors for this enclosure are described in Sect. F16.2.3. Note that these expressions account for the shading effects in the enclosure which result from the presence of the cylinder.

Direct radiant exchange in the cylindrical enclosure may be described by interaction between:

1. the lower ring and the shell,

2. the upper ring and the shell,

3. the lower ring and the upper ring, and

4. the shell and itself.

The analytic expressions for the geometric configuration factors for this enclosure are described in Sect. F16.2.4.

F16.2.3 Annular Geometric Configuration Factors

The annular enclosure may be described as consisting of a cylindrical surface, two ring surfaces, and a shell surface, as illustrated in Fig. F16.2.3. The analytic expressions for the configuration factor between nodal areas that lie on these major surfaces are described in the following sections. The view factor algebra necessary to calculate reciprocal factors is not presented.

F16.2.3.1 Cylinder-to-ring factors

The configuration factor from cylindrical area 1 to ring area 2, $F_{1\text{-}2}$, illustrated in Fig. F16.2.5, is a function of the cylinder-to-disk configuration factor $F_{(a\text{-}b)}$, illustrated in Fig. F16.2.6 (Ref. 2). The factor $F_{1\text{-}2}$ may be expressed as:

$$F_{1\text{-}2} = \frac{A_1 + A_4}{A_1}\left(F_{((1+4)\text{-}(2+3))} - F_{((1+4)\text{-}(3))}\right) - \frac{A_4}{A_1}\left(F_{((4)\text{-}(2+3))} - F_{((4)\text{-}(3))}\right), \quad \text{(F16.3)}$$

where

$$F_{(a-b)} = \frac{1}{2\pi}\left\{\cos^{-1}\left(\frac{h^2 - r_d^2 + r_c^2}{h^2 + r_d^2 - r_c^2}\right) - \left(\frac{r_c}{2h}\right)\left[\frac{\left(h^2 + r_d^2 + r_c^2\right)^2}{r_c^4} - 4\left(\frac{r_d}{r_c}\right)^2\right]^{1/2}\right.$$

$$\cdot\cos^{-1}\frac{r_d\left(h^2 - r_d^2 + r_c^2\right)}{r_d\left(h^2 + r_d^2 - r_c^2\right)} + \left[\frac{h^2 - r_d^2 + r_c^2}{r_c^2}\right]\sin^{-1}\left(\frac{r_c}{r_d}\right) \qquad \text{(F16.4)}$$

$$\left.- \left(\frac{\pi}{2}\right)\left[\frac{h^2 + r_d^2 - r_c^2}{r_c^2}\right]\right\}.$$

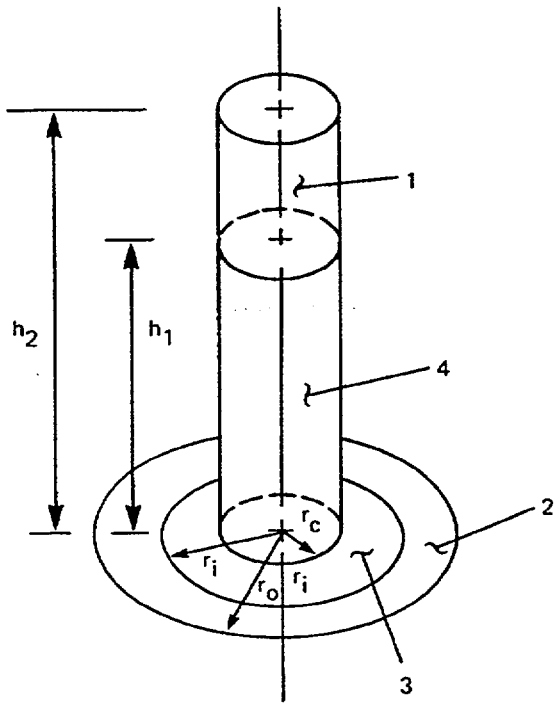


Figure F16.2.5 Cylinder-to-ring model

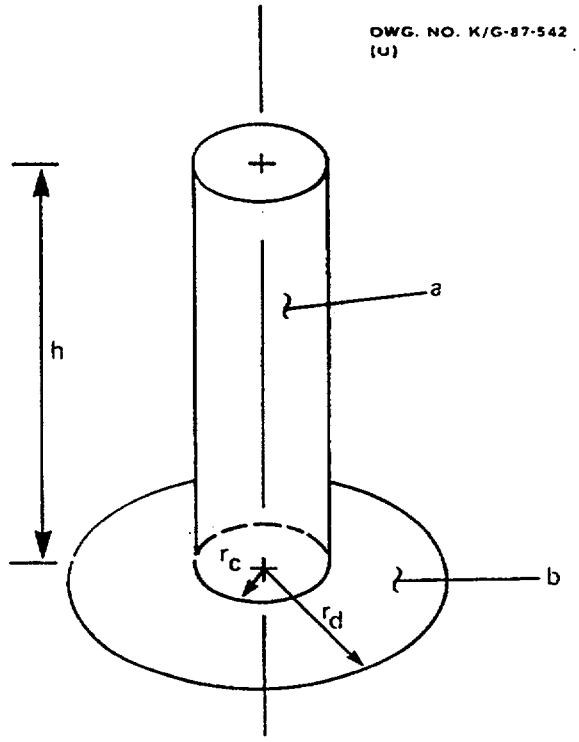


Figure F16.2.6 Cylinder-to-disk model

F16.2.3.2 Cylinder-to-shell factors

The configuration factor from cylindrical area 1 to shell area 2, $F_{1\text{-}2}$, is a function of the cylinder-to-disk configuration factor, defined in Sect. F16.2.3.1, and the relative position of area 2 with respect to area 1 (Ref. 2). The four configurations are illustrated in Fig. F16.2.7 and will be referred to as I, II, III,

and IV. In each case, the radius of the cylinder is r_c, and the shell is r_o. A shorthand notation is used where $F_{(h_1-h_2)}$ represents the cylinder-to-disk factor $F_{(a-b)}$, with $h = h_1 - h_2$. The factors for each configuration are as follows:

(a) Configuration I

$$F_{1-2} = \frac{1}{h_1}\left[h_3 + h_2 F_{(h_2)} + h_4 F_{(h_4)} - (h_2 + h_3)F_{(h_2 + h_3)} - (h_3 + h_4)F_{(h_3 + h_4)}\right], \quad \text{(F16.5)}$$

(b) Configuration II

$$F_{1-2} = \frac{1}{h_1}\left[h_2 F_{(h_2)} + h_4(1 - F_{(h_4)}) + (h_3 - h_4)F_{(h_3 - h_4)} - (h_2 + h_3)F_{(h_2 + h_3)}\right], \quad \text{(F16.6)}$$

(c) Configuration III

$$F_{1-2} = \frac{1}{h_3}\left[h_3 + h_2 F_{(h_2)} + h_4 F_{(h_4)} - (h_2 + h_3)F_{(h_2 + h_3)} - (h_3 + h_4)F_{(h_3 + h_4)}\right], \quad \text{(F16.7)}$$

(d) Configuration IV

$$F_{1-2} = \frac{1}{h_1}\left[(h_1 + h_2)F_{(h_1 + h_2)} + (h_2 + h_3)F_{(h_2 + h_3)} - h_2 F_{(h_2)} - (h_1 + h_2 + h_3)F_{(h_1 + h_2 + h_3)}\right]. \quad \text{(F16.8)}$$

DWG. NO. K/G-87-534 (U)

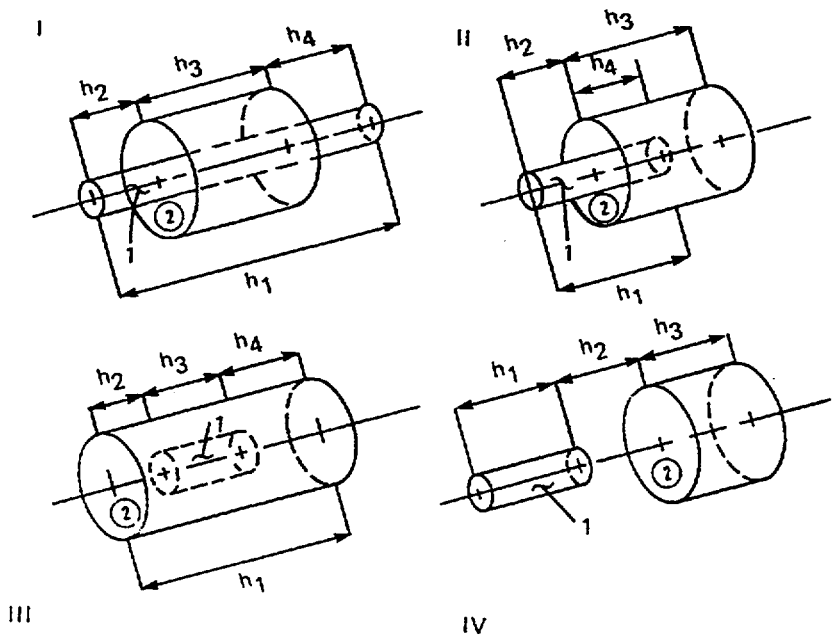


Figure F16.2.7 Cylinder-to-shell configurations

F16.2.3.3 Ring-to-shell factors

The configuration factor from ring area 1 to shell area 2, F_{1-2}, illustrated in Fig. F16.2.8 may be expressed as (Ref. 3):

$$F_{1-2} = \frac{2}{\rho_o^2 - \rho_i^2} \int_{\rho_i}^{\rho_o} r F_{dA_1-A_2}\, dr, \tag{F16.9}$$

where

$$F_{dA_1-A_2} = \frac{1}{\pi} \tan^{-1}\left[\frac{r_s + r}{r_s - r} \tan\left(\frac{\omega}{2}\right)\right] - \frac{(r_s^2 - r^2 - h^2)}{\pi\sqrt{(r_s^2 + r^2 + h^2)^2 - (4r^2 r_s^2)}} \tag{F16.10}$$

$$\tan^{-1}\left[\frac{\sqrt{(r_s^2 + r^2 + h^2)^2 - 4(r^2 r_s^2)}}{r_s^2 + r^2 + h^2 - 2rr_s} \tan\left(\frac{\omega}{2}\right)\right],$$

for

$$\omega = \cos^{-1}\left(\frac{r_c}{r}\right) + \cos^{-1}\left(\frac{r_c}{r_s}\right). \tag{F16.11}$$

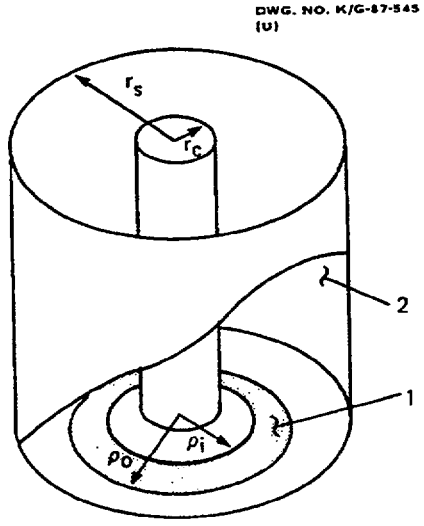


Figure F16.2.8 Annular ring-to-shell model

F16.2.3.4 Ring-to-ring factors

The configuration factor from ring 1 to ring 2 separated by a solid coaxial cylinder (Ref. 4), as shown in Fig. F16.2.9, may be expressed as

$$F_{1-2} = \frac{2}{(\rho_o^2 - \rho_i^2)} \int_{\rho_i}^{\rho_o} \rho F_{dA_1 - A_2} d\rho \,, \tag{F16.12}$$

where

$$F_{dA_1 - A_1} = \frac{\omega - \phi}{2\pi} + \frac{r_o^2 - \rho^2 - h^2}{\pi\sqrt{(r_o^2 + \rho^2 + h^2)^2 - 4\rho^2 r_o^2}} \tan^{-1}\left(\frac{\sqrt{r_o^2 + \rho^2 + h^2 + 2\rho r_o}\, \tan\frac{\omega}{2}}{\sqrt{r_o^2 + \rho^2 + h^2 - 2\rho r_o}} \right) \tag{F16.13}$$

$$- \frac{(r_i^2 - \rho^2 - h^2)}{\pi\sqrt{(r_i^2 + \rho^2 + h^2)^2 - 4\rho^2 r_i^2}} \tan^{-1}\left(\frac{\sqrt{r_i^2 + \rho^2 + h^2 + 2\rho r_i}\, \tan\frac{\phi}{2}}{\sqrt{r_i^2 + \rho^2 + h^2 - 2\rho r_i}} \right) ,$$

for

$$\omega = \cos^{-1}\left(\frac{r_c}{\rho}\right) + \cos^{-1}\left(\frac{r_c}{r_o}\right) , \tag{F16.14}$$

$$\phi = \cos^{-1}\left(\frac{r_c}{\rho}\right) + \cos^{-1}\left(\frac{r_c}{r_i}\right) . \tag{F16.15}$$

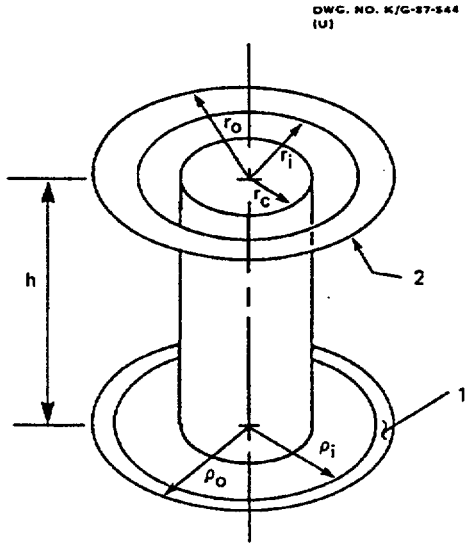


Figure F16.2.9 Annular ring-to-ring model

F16.2.3.5 Shell-to-shell factors

The configuration factor from the interior of a shell to itself for an annular enclosure (Ref. 5), as illustrated in Fig. F16.2.10, may be expressed as

$$F_{1-1} = 1 - \frac{1}{R} + \frac{2}{\pi R}\tan^{-1}\frac{2\sqrt{R^2-1}}{L}$$

$$- \frac{L}{2\pi R}\left[\frac{\sqrt{4R^2+L^2}}{L}\sin^{-1}\frac{4(R^2-1)+\left(\frac{L^2}{R^2}\right)(R^2-2)}{L^2+4(R^2-1)}\right] \qquad \text{(F16.16)}$$

$$- \sin^{-1}\frac{(R^2-2)}{R^2} + \frac{\pi}{2}\left(\frac{\sqrt{4R^2+L^2}}{L} - 1\right),$$

where

$$R = \frac{r_s}{r_c}, \qquad \text{(F16.17)}$$

$$L = \frac{h}{r_c} . \qquad \text{(F16.18)}$$

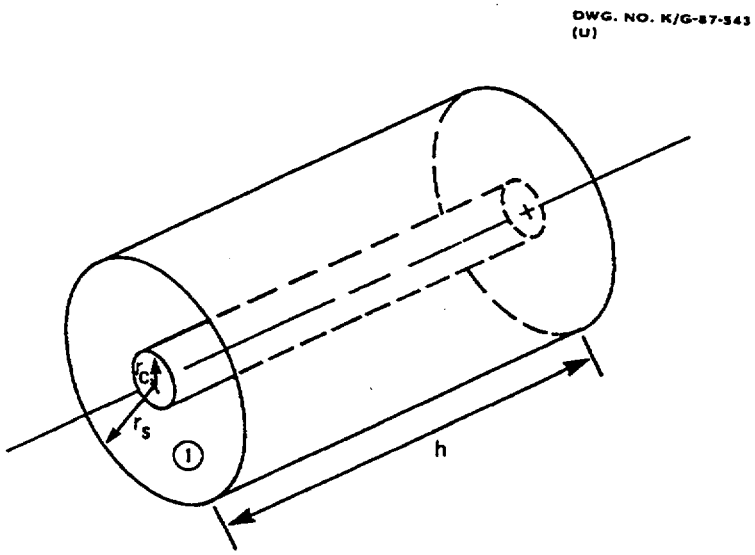


Figure F16.2.10 Annular shell-to-shell model

F16.2.4 Cylindrical Geometric Configuration Factors

The cylindrical enclosure consists of two ring surfaces and a shell surface, as illustrated in Fig. F16.2.4. In the model generation process, the program treats the centerline of the enclosure as a window. Thus, the centerline is a pseudosurface that must be included in the input to completely define the enclosure. Although this surface is used to define the enclosure, it is not included in the solution technique. The analytic expressions for the configuration factor between nodal areas that lie on these major surfaces are described in the following sections. The view factor algebra necessary to calculate reciprocal factors is not presented.

F16.2.4.1 Shell-to-ring factors

The configuration factor from the interior of shell area 1 to ring area 2, $F_{1\text{-}2}$, as illustrated in Fig. F16.2.11, may be expressed as (Ref. 6):

$$F_{1\text{-}2} = \frac{1}{4(L_2 - L_1)}\left[\sqrt{L_2^4 + 2L_2^2(1 + R_2)^2 + (1 - R_2^2)^2} - \sqrt{L_2^4 + 2L_2^2(1 + R_1^2) + (1 - R_1^2)^2}\right.$$
$$\left. + \sqrt{L_1^4 + 2L_1^2(1 + R_1^2) + (1 - R_1^2)^2} - \sqrt{L_1^4 + 2L_1^2(1 + R_2^2) + (1 - R_2^2)^2}\right], \qquad \text{(F16.19)}$$

where

$$L_1 = \frac{h_1}{r_s}, \tag{F16.20}$$

$$L_2 = \frac{h_2}{r_s}, \tag{F16.21}$$

$$R_1 = \frac{r_i}{r_s}, \tag{F16.22}$$

$$R_2 = \frac{r_o}{r_s}. \tag{F16.23}$$

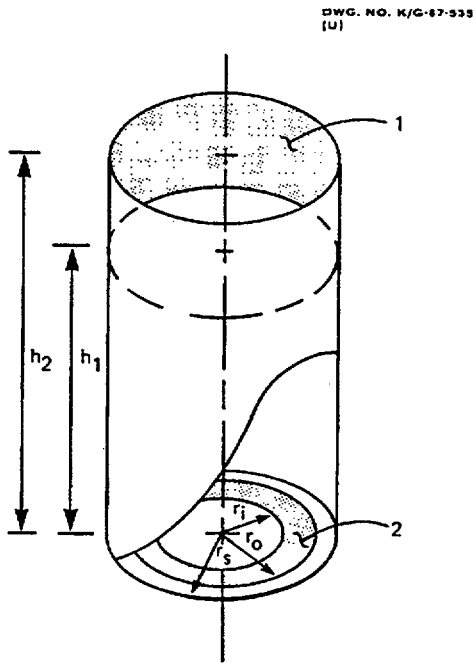


Figure F16.2.11 Cylindrical shell-to-ring model

F16.2.4.2 Ring-to-ring factors

For two coaxial rings, the configuration factor from ring 1 to ring 2, $F_{1\text{-}2}$, is a function of the configuration factor between two coaxial disks, $F_{(a\text{-}b)}$ (Ref. 5). The ring-to-ring configuration is illustrated in Fig. F16.2.12, the disk-to-disk configuration in Fig. F16.2.13.

$$F_{1-2} = \frac{A_1 + A_4}{A_1}\left(F_{((1+4)-(2+3))} - F_{((1+4)-3)}\right) - \frac{A_4}{A_1}\left(F_{(4-(2+3))} F_{(4-3)}\right), \quad \text{(F16.24)}$$

where

$$F_{(a-b)} = \frac{1}{2}\left[1 + \frac{1+R_b^2}{R_a^2} - \left(\left(1 + \left(\frac{1+R_b^2}{R_a^2}\right)\right)^2 - 4\left(\frac{R_b}{R_a}\right)^2\right)^{1/2}\right], \quad \text{(F16.25)}$$

$$R_a = \frac{a}{h}, \quad \text{(F16.26)}$$

$$R_b = \frac{b}{h}. \quad \text{(F16.27)}$$

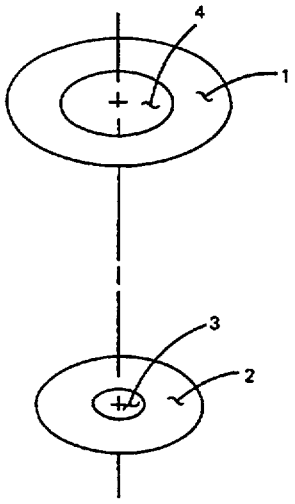


Figure F16.2.12 Cylindrical ring-to-ring model

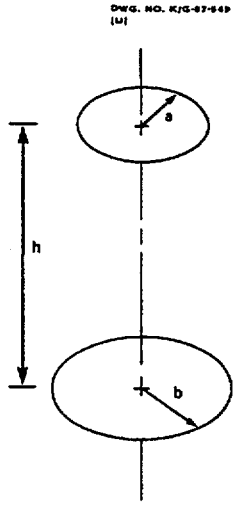


Figure F16.2.13 Disk-to-disk model

F16.2.4.3 Shell-to-shell factors

The configuration factor from the interior of a shell to itself, $F_{1\text{-}1}$, as illustrated in Fig. F16.2.14, may be expressed as (Ref. 6):

$$F_{1-1} = \left(1 + \frac{h}{2r_s}\right) - \left(1 + \left(\frac{h}{2r_s}\right)^2\right)^{1/2} . \qquad \text{(F16.28)}$$

F16.2.5 Exchange Factors

The exchange factor is defined as the fraction of energy emitted by surface i (if it is assumed to be a black surface) absorbed by surface j (Ref. 1). The exchange factor depends on the geometric arrangement of all surfaces in the enclosure and their emittances. OCULAR solves for the exchange factors using Hottel's Matrix Method. This portion of the code is modeled upon an earlier computer program, HOTTEL (Ref. 7). OCULAR outputs the factor between two nodal areas and the sum of the factors for each emitting node. This sum should equal the emittance of the emitting nodal area. Since OCULAR employs exact solutions for configuration factor computations, any differences that do exist are a result of numerical integration over a finite surface area. The magnitude of the configuration factor asymptotically approaches the exact solution for the annular ring-to-ring and ring-to-shell factors as the finite surface area approaches zero. These solutions are presented as Eqs. (F16.9) and (F16.12). Experience has shown that the configuration factor for the innermost ring contains this error. Thus, care should be exercised when defining the mesh spacing for the annular enclosure in this region. If the difference due to discretization error is unacceptable, the mesh should be refined in this region and the problem repeated.

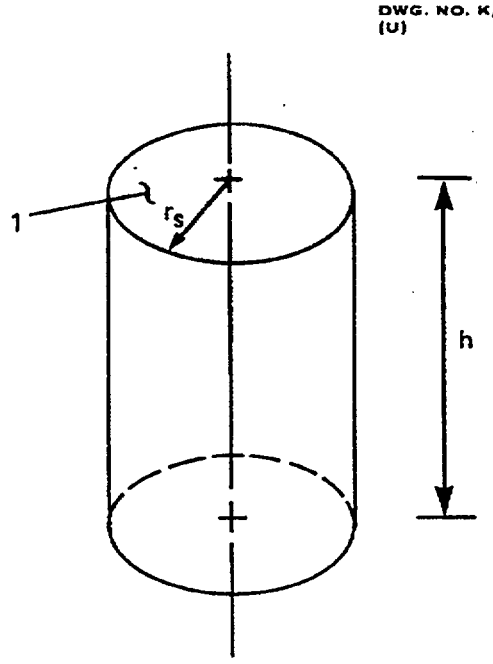


Figure F16.2.14 Shell-to-shell model

The exchange factors are calculated assuming that the following restrictions in the heat transfer analysis will be met:

1. the temperature is uniform across a nodal area,

2. the directional spectral emissivity and the directional spectral absorptivity are equal and independent of either angle or wavelength,

3. all energy is emitted and reflected diffusely, and

4. the incident and reflected energy flux is uniform over each nodal area.

F16.3 PROGRAM LOGIC FLOW AND SUBROUTINE DESCRIPTION

OCULAR reads input data for one case per computer run. Execution is halted after the input data are read and the exchange factors are generated. The calling sequence of routines is depicted in Fig. F16.3.1.

F16.3.1 Description of Subroutines Used by OCULAR

Descriptions of subroutines and entry points that comprise OCULAR are presented in Table F16.3.1.

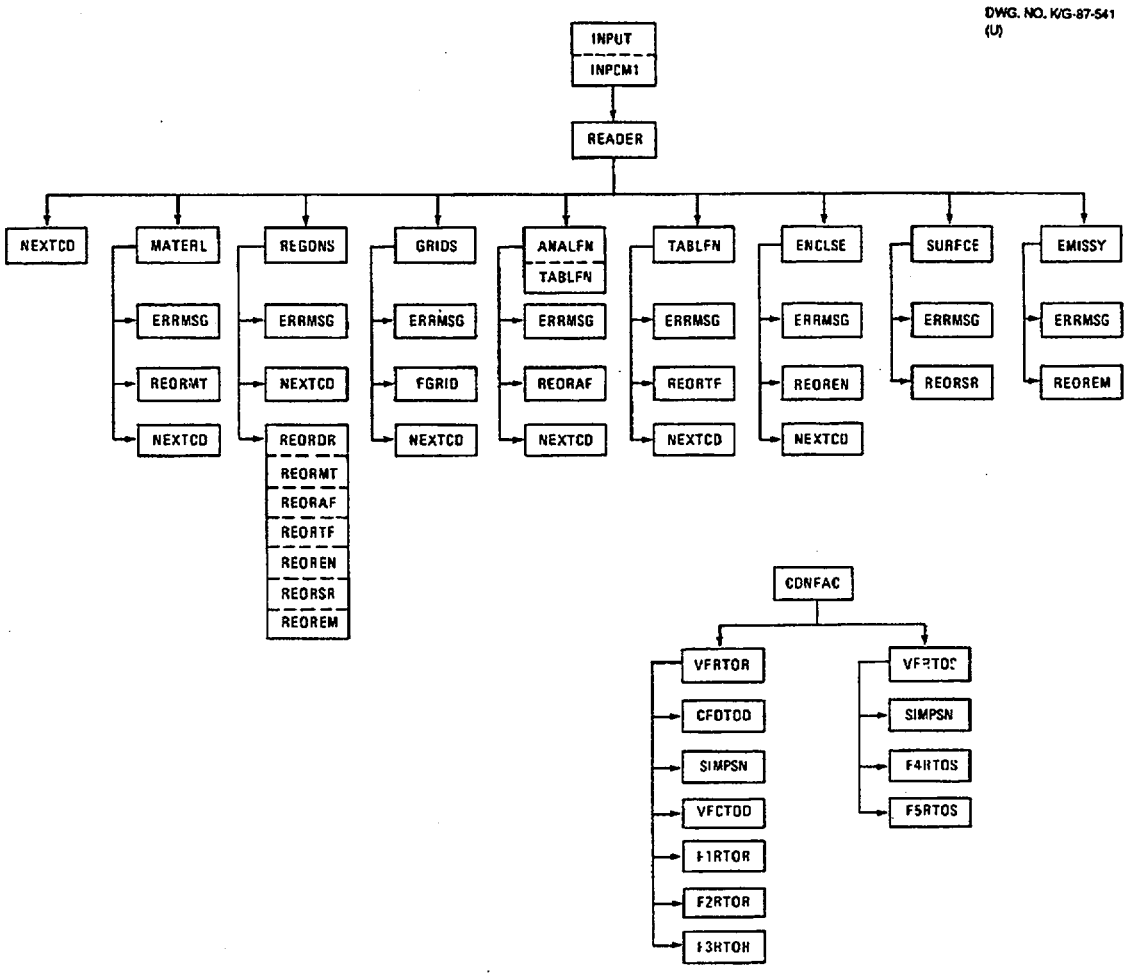


Figure F16.3.1 Calling sequence of routines

Table F16.3.1 Summary of subroutines and entries

BLOCK DATA	Initializes data in labeled commons /ARYLNG/ and /IOUNIT/
MAIN OCULRA	Directs ALOCAT to call subroutine ALCROR and opens buffers.
ALCROR	Serves as an interface between the OCULAR main routine and subroutines. Determines the maximum dimension of the array that holds all variably dimensioned arrays.
ANALFN	Reads card images for one analytical function from the analytical function data block for each call. Checks the data to determine if they are compatible and writes error messages if they are not. Called from READER. Calls ERRMSG, NEXTCD, REORAF, and REORTF. Calls SCALE free-form reading routines DREAD and IREAD.
ANALYT	Entry in PREANA. Evaluates the referenced analytical function or calls the specified user-supplied subroutine. Called from FUNCTN. Calls EMSSVY.
BLDENC	Builds the general-enclosure mesh that overlays the problem mesh. Called from OCULST.
CFDTOD	Calculates the view factor from one radial disk to another coaxial radial disk. Called from VFRTOR.
CFSTOS	Calculates the view factor from the interior of a right circular cylinder to itself for an annular or cylindrical enclosure. Called from VFSTOS.
CHKATF	Checks the external reference number (input data) to an analytical or tabular function to determine whether or not it has been defined. Calculates internal reference numbers to defined functions. Writes error messages for undefined functions. Called from ERRCK.
CONFAC	Serves as a driver for the calculation of the analytic geometric view factors and exchange factors. Called from OCULST. Calls EXCHNG, FUNCTN, VFCTOR, VFCTOS, VFRTOR, VFRTOS, and VFSTOS.
DECOMP	Decomposes a real matrix by Gaussian elimination and estimates the condition of the matrix. Called from EXCHNG. Calls SOLVE (Ref. 8).
DIFFER	Calculates the fine lattice lines that bound a region and a surface. Called from WRITER. Calls REGION.
ECHO	Reads remaining standard input data card images from unit IECHO, lists input data card images on unit IO, blanks out columns 73 through 80 except for title cards, comments, and end of data indicators, and writes the resulting input data 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 blank. Called from MAIN.

Table F16.3.1 (continued)

EMISSY	Reads card images for one emissivity function from the emissivity function data block for each call. Checks the data to determine if they are compatible and writes error messages if they are not. Called from READER. Calls ERRMSG and REOREM. Calls SCALE free-form reading routines DREAD and IREAD.
EMSSVY	Evaluates the user-supplied function defining the emissivity for the portion of the node whose properties are being evaluated. Writes an error message and stops when called unless the user-supplied algorithm has been added. Called from ANALYT.
ENCLSE	Reads card images for one enclosure function from the enclosure function data block for each call. Checks the data to determine if it is compatible and writes error messages if it is not. Called from READER. Calls ERRMSG and REOREN. Calls SCALE free-form reading routine IREAD.
ENREGN	Entry in REGION. Writes error message if the fine grid matching a surface boundary cannot be located. Called from DIFFER.
ERRCK	Checks references by the region, material, enclosure, surface, and emissivity input data to determine whether or not the referenced parameters have been defined. Called from INPUT. Calls CHKATF.
ERRMSG	Called when the number of parameters defined by the input data exceeds the maximum allowed by the code. Writes error message and sets flag indicating number of errors encountered. Called from ANALFN, EMISSY, ENCLSE, GRIDS, MATERL, REGONS, and SURFCE. Calls SCALE free-form reading routine AREAD.
EXCHNG	Uses Hottel's matrix solution to calculate the exchange factor terms. Called by CONFAC. Calls DECOMP and SOLVE.
F1RTOR	Function representing an integrand of the function in the $F_{dA_1 - A_2}$ ring-to-ring configuration factor calculations. Accessed by VFRTOR through SIMPSN.
F2RTOR	Function representing an integrand of the function in the $F_{dA_1 - A_2}$ ring-to-ring configuration factor calculations. Accessed by VFRTOR through SIMPSN.
F3RTOR	Function representing an integrand of the $F_{dA_1 - A_2}$ function in the ring-to-ring configuration factor calculations. Accessed by VFRTOR through SIMPSN.
F4RTOS	Function representing an integrand of the $F_{dA_1 - A_2}$ function in the ring to dA, shell configuration factor calculations. Accessed by VFRTOS through SIMPSN.
F5RTOS	Function representing an integrand of the $F_{dA_1 - A_2}$ function in the ring-to-shell configuration factor calculations. Accessed by VFRTOS through SIMPSN.

Table F16.3.1 (continued)

FGRID	Calculates the fine gridlines along each axis based on the respective set of gross gridlines and their related divisions. Called from GRIDS.
FUNCTN	Computes the value of an emissivity which can be positional-dependent. Value of the emissivity is a function of the independent variables X (or R), Y (or θ), and Z (or ϕ). Called from CONFAC. Calls entries ANALYT and TABLE.
GRIDS	Reads the card images from the XGRID, YGRID, and ZGRID data blocks. Called from READER. Calls ERRMSG, FGRID, and NEXTCD.
INITIL	Initializes variables and arrays to zero. Called from INPUT.
INPUT	Reads the first four input cards and calls routines to read and write the remaining input data. Job information and parameters are output. Called from OCULST. Calls ERRCK, INITIL, NEXTCD, OUTPUT, PREANA, PRETAB, READER, SETTYP, and WRITER. Calls SCALE free-form reading routines AREAD, DREAD, IREAD, and SCANON.
MATERL	Reads card images for one material from the material data block for each call. Checks the data to determine if they are compatible and writes error messages if they are not. Called from READER. Calls ERRMSG, NEXTCD, and REORMT. Calls SCALE free-form reading routines CREAD, DREAD, and IREAD.
NEXTCD	Reads the first word of the next card image. Writes message to include last card image successfully read if "end-of-file" or "end-of-data" is encountered and sets appropriate flag. Writes out entire card image if it is a comment and looks at next card image. Otherwise, resets flag so that the next read is from first word of this card image and returns to the calling subroutine. Called from ANALFN, ENCLSE, GRIDS, INPUT, MATERL, READER, REGONS, SURFCE, and TABLFN. Calls SCALE routine AREAD.
OCULST	Sets the size of the variably dimensioned arrays and initializes the parameters for the run. Called from ALCROR. Calls entries INPUT, NEXTCD, PLANES, POINTS, BLDENC, CONFAC, and PRINTR. Calls SCALE routine IONUMS.
OPTION	Reads the OPTION data block and writes out the values of the parameters that can be reset with this data block. Called from MAIN.
OUTPUT	Writes out the parameters from Card 2, checks the parameters for errors, and writes error messages if errors are encountered. Called from INPUT.
PLANES	Performs error checking to determine whether the surface data are properly defined and whether enclosures are defined by prescribed surfaces. Called from OCULST.
POINTR	Sets the size of the variably dimensioned arrays. Called from MAIN.

Table F16.3.1 (continued)

POINTS	Generates the nodal configuration and the arrays defining the fine lattice lines passing through each node. Called from OCULST.
PREANA	Serves as initial entry for passing variably dimensioned arrays for later use by entry ANALYT. Called from INPUT.
PRETAB	Serves as initial entry for passing variably dimensioned arrays for later use by entry TABLE. Checks tabular data for consistency and writes messages if errors are detected. Called from INPUT.
PRINTR	Writes out the exchange factor data and the HEATING CONNECTOR input data. Called from OCULST.
READER	Reads each card image in the standard input data beginning with the data immediately following the title card and card 2. Checks for keywords defining a data block and branches to associated subroutine where related input data from the block are read. Assumes card image belongs to currently open data block if a keyword is not found. Writes error message and continues with next card image if keyword is not found on first card image. Returns to INPUT upon encountering an "end-of-file" or an "end-of-data." Called from INPUT. Calls MATERL, NEXTCD, REGONS, GRIDS, ANALFN, ENCLSE, SURFCE, and EMISSY. Calls entry TABLFN.
REGION	Checks input data to ensure that the boundary of each region corresponds to a fine gridline. Called from DIFFER.
REGONS	Reads card images for one region from the region data block for each call. Checks the data to determine if they are compatible and writes error messages if they are not. Called from READER. Calls ERRMSG, NEXTCD, and REORDR. Calls SCALE free-form reading routines DREAD and IREAD.
REORDR	Reorders arrays containing the input data associated with the region data block when a region is deleted by the input data. Entries REORMT, REORAF, REORTF, REOREN, REORSR, and REOREM are independent sections of coding and perform similar tasks on other input data items. Called from REGONS.
REORAF	Reorders arrays containing the input data associated with the analytical function data block when an analytical function is deleted by the input data. Called from ANALFN.
REOREM	Reorders arrays containing the input data associated with the emissivity function data block when an emissivity function is deleted by the input data. Called from EMISSY.
REOREN	Reorders arrays containing the input data associated with the enclosure function data block when an enclosure function is deleted by the input data. Called from ENCLSE.

Table F16.3.1 (continued)

REORMT	Reorders arrays containing the input data associated with the material function data block when a material function is deleted by the input data. Called from MATERL.
REORSR	Reorders arrays containing the input data associated with the surface function data block when a surface function is deleted by the input data. Called from SURFCE.
REORTF	Reorders arrays containing the input data associated with the tabular function data block when a tabular function is deleted by the input data. Called from entry TABLFN.
SETTYP	Initializes flags and parameters that are a function of the geometry type. Called from INPUT.
SIMPSN	Integrates a specified function using Simpson's Rule. Called by VFRTOR and VFRTOS. Accesses functions F1RTOR, F2RTOR, F3RTOR, F4RTOS, and F5RTOS.
SOLVE	Solves the linear system AA * X = CK. Called from EXCHNG. Calls DECOMP (Ref. 8).
SURFCE	Reads card images for one surface from the surface data block for each call. Checks the data to determine if they are compatible and writes error messages if they are not. Called from READER. Calls ERRMSG, NEXTCD, and REORSR. Calls SCALE free-form reading routines DREAD and IREAD.
TABLE	Entry in PRETAB. Performs checking to ensure tabular function values bound the argument values. Writes warning message. Called from FUNCTN.
TABLFN	Entry in ANALFN. Reads card images for one tabular function from the tabular function data block for each call. Checks the data to determine if they are compatible and writes error messages if they are not. Called from READER. Calls ERRMSG, NEXTCD, and REORTF. Calls SCALE free-form reading routines DREAD and IREAD.
VFCTOD	Computes the configuration factor from the outside surface of a cylinder to a disk. Called from VFCTOR and VFCTOS.
VFCTOR	Calculates the configuration factor from the outer surface of a cylinder to a ring at the base of the cylinder for an annular enclosure. Called from CONFAC. Calls VFCTOD.
VFCTOS	Calculates the configuration factor from the exterior of an inner-right circular cylinder of finite length to the interior of an outer-right circular cylinder (shell) using cylinder-to-disk configuration factors and view factor algebra for an annular enclosure. Called from CONFAC. Calls VFCTOD.

Table F16.3.1 (continued)

VFRTOR	Calculates the configuration factor between coaxial rings. Called from CONFAC. Calls CFDTOD, SIMPSN, and VFCTOD. Accesses functions F1RTOR, F2RTOR, and F3RTOR.
VFRTOS	Calculates the configuration factor between a ring in the enclosure and the inner surface of the outer shell. Calls SIMPSN. Accesses F4RTOS and F5RTOS.
VFSTOS	Calculates the configuration factor from the interior of a right circular cylindrical shell of finite length to itself and from the interior of a right circular cylindrical shell to the interior of a coaxial shell. Called from CONFAC, calls CFSTOS.
WRITER	Writes out the input data values, except for the title card, card 2 and comment cards. Checks for errors and outputs messages if errors are detected. Called from INPUT. Calls DIFFER.

F16.4 INPUT DESCRIPTION FOR OCULAR

F16.4.1 General

A detailed discussion of the input data cards is presented, followed by a brief outline of the input data for the benefit of the user who is familiar with the data preparation. In preparing the input data, any consistent set of units may be used in the OCULAR program. The units associated with the algorithms which appear in the user-supplied subroutine EMSSVY must be consistent with those of the input data.

The input data set for OCULAR is derived from the HEATING input data set with the intention that OCULAR can read and process the HEATING input data. Thus, a user may develop a HEATING input deck to model a heat transfer problem, include the additional enclosure, surface, and emissivity information required by OCULAR, and use the same data set that will be used for the thermal analysis to produce the necessary exchange factors. OCULAR will read only the data needed from the HEATING input deck. By necessity, OCULAR data assume the same format as HEATING data. If a user chooses not to use a HEATING input data set, zeros should be supplied in appropriate HEATING data entries. The input description assumes that a HEATING data set is not being used.

Formatted input data are read from logical unit No. 4 and unformatted input data are read from logical unit No. 95. Formatted output is written to logical unit No. 6 and unformatted node-to-node connector data are written to logical unit No. 12.

The input data for OCULAR are read using the SCALE free-form reading routines (see Sect. M2). When OCULAR is executed within the SCALE system, =OCULAR or #OCULAR must be the first record in the data set. An input data set is terminated with a percent (%) sign. When executed within the SCALE system, an END card must be supplied as the last card of the data deck to signify the end of the input data to the SCALE driver.

Data are entered in an unformatted manner by separating each data item by one or more blanks or by a comma. With a few exceptions noted, the data items may be entered anywhere in columns 1 through 72. If the data that are to appear on a card will not fit on one card, an "@" must appear in column 1 of each continuation card. Otherwise, the code will read the remaining data for that card image as either zero or blank. There is no limit to the number of continuation cards. However, a continuation card may not immediately follow a title card, an end-of-data card, a comment card, or a keyword indicator card. Columns 73 through 80 of each card are reserved for identification to aid the user in the preparation and handling of data. Decimal data may be entered as FORTRAN input. For example, 1.733×10^{-4} may be entered as 1.733-4, 1.733E-4, 1.733D-4, or 0.0001733. Note that imbedded blanks are not allowed within a given number representation. SCALE free-form reading routines have provisions for multiple entries of the same data value. This step is done by entering the number of repeats, followed by either R, *, or $, followed by the data value to be repeated. For example, 5R2 or 5*2 enters five successive 2's in the input data. No blanks should be inserted between the multiplier and the repeat flag, but each multiple entry must be separated from the rest of the data by one or more blanks or by a comma. Because blanks are ignored, an embedded zero must be specified in the input data either as 0 followed by one or more blanks, as 0 followed by a comma, or as one or more blanks followed by a comma. However trailing blanks may be omitted. Multiple zeros may be specified as NZ, where N is the number of zeros to be repeated. Again, there must not be any blanks between the N and the Z, but the NZ must be separated from the rest of the data by one or more blanks or by a comma. All literal data fields, such as material names, must be delineated by a blank, they must not be delineated by a comma.

The input data for an execution of OCULAR consist of data defining the size of the variably dimensioned arrays and the options that will be in effect during the computer run. Case parameters that dictate the required size of the variably dimensioned arrays are input with the OPTIONS data block. This data block contains variables whose default values may be modified and it is placed at the beginning of an OCULAR data set.

This data block is followed by case data. A case consists of a title card, one parameter card, and the data blocks necessary to define the problem being modeled. The user may wish to add comment cards to the input data for documentation purposes. Any number of comment cards may appear anywhere in the input data following the title card, except a comment card may not precede a continuation card. The comment cards are identified by an asterisk (*) in column 1. The comments will be written on the standard output unit as they are encountered.

Ten different data blocks may be used in describing the model to the computer code. A data block consists of a block indicator card containing the keyword followed by the data for that block. No embedded blanks can be present in the first four characters of the keyword. The keyword for each data block is contained in Table F16.4.1.

Table F16.4.1 Data block keywords

REGIONS
MATERIALS
XGRID
YGRID
ZGRID
ANALYTICAL FUNCTIONS
TABULAR FUNCTIONS
ENCLOSURES
SURFACES
EMISSIVITIES

The data reference number blocks may appear in any order within the block and may have any positive integer value accepted by the machine. If a data reference number appears more than once within a data block, the last data read with that reference number are used.

The final data card in a data deck for a case is an end-of-data 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 OCULAR 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, OCULAR terminates reading and commences processing the input data. After processing is completed, the code performs the specified calculations unless an error was encountered.

In the sections that follow, the characters (A), (I), or (R) may appear after the heading for the entries to indicate that the entry is alphanumeric, integer, or real, respectively. Suggested data card labels follow each data block section heading. These labels may be entered in columns 73 through 80 of the related data cards to aid in identifying the data.

F16.4.2 OPTION Data Block

The variables labeled in commons /ARYLNG/ and /IOUNIT/ are initialized in a BLOCK DATA subprogram. In general, it is advantageous to tailor OCULAR to a specific problem by overriding the default values specified in the BLOCK DATA with values appropriate for the problem. This adaptation is accomplished through the use of the OPTION data block input feature described below. Each of the variables in the two common blocks whose default value can be modified appears in this data block.

The first record read by OCULAR contains data for the OPTION data block. This feature allows the user to change the default value of any or all variables in the data block by including the variable name and its desired value. The variables are input beginning with OPTION and ending with END. These parameters apply to OCULAR for the entire run. Table F16.4.2 contains each variable name and its default value.

Table F16.4.2 List of OCULAR variables whose default value may be modified through the OPTION data block

MAXANA	5	Maximum number of analytical functions
MAXCON	1	Maximum number of connectors
MAXEMS	5	Maximum number of emissivity functions
MAXENC	5	Maximum number of enclosures
MAXGGL	10	Maximum number of gross gridlines along any axis
MAXMAT	5	Maximum number of materials
MAXNEA	8	Maximum number of nodal areas in an enclosure
MAXPRS	10	Maximum number of pairs in each tabular function
MAXPTS	50	Maximum number of lattice points (nodes)
MAXREG	5	Maximum number of regions
MAXRFG	50	Maximum number of X (or R) fine gridlines
MAXSRF	5	Maximum number of enclosure surfaces
MAXTBL	5	Maximum number of tabular functions
MAXTFG	50	Maximum number of Y (or 0) fine gridlines
MAXZFG	50	Maximum number of Z (or 0) fine gridlines

F16.4.3 Problem Control Parameters

F16.4.3.1 Card 1

This card, which can contain alphanumeric characters in the first 72 columns, contains a descriptive title for the problem and must be the first card in the data deck. The card itself cannot be omitted, although it may be left blank. This title serves to identify the output associated with a problem, and it appears at the top of a number of the tables generated by the program. It is also included in messages output by the code.

F16.4.3.2 Card 2

F16.4.3.2.1 Entry 1

The first entry on card 2 is not processed by OCULAR. The entry itself cannot be omitted and should be set to zero if the user is not using a HEATING input data deck.

F16.4.3.2.2 Entry 2 - geometry type

The HEATING program offers twelve possible geometries (seven and eight are functionally the same) which are members of either the cylindrical, Cartesian, or the spherical coordinate system. The geometry types are listed in Table F16.4.3 below.

Table F16.4.3 HEATING geometry types

Cylindrical	Rectangular	Spherical
1 R-θ-Z	6 X-Y-Z	10 R
2 R-θ	7 X-Y	11 R-θ
3 R-Z	8 X-Z	12 R-θ-ϕ
4 R	9 X	
5 Z		

Currently OCULAR is designed to model enclosures for the R-Z geometry type. Thus, entry 2 on card 2 should be 3.

F16.4.4 Region Data Block (Cards R1 and R2)

Each region is described by two cards that must appear in pairs. The cards are repeated for each region. There must be at least one region specified. The maximum number of regions is MAXREG.

F16.4.4.1 Block indicator card

REGIONS

F16.4.4.2 Card R1

a. Region Number (I)

This entry contains the number of the region to be described. The region numbering system does not require that a region occupy any particular zone in the overall configuration.

b. Region Material Number (I)

The second entry of this card defines the material which occupies the region. The integer entry is the function number of the material listed in the material data block.

c. Region Dimensions (R)

Dimensions of the region boundaries are entered as floating-point numbers and are arranged in the following order:

1. smaller dimension of X or R region boundary,
2. larger dimension of X or R region boundary,
3. smaller dimension of Y or θ region boundary,
4. larger dimension of Y or θ region boundary,
5. smaller dimension of Z or ϕ region boundary, and
6. larger dimension of Z or ϕ region boundary.

These dimensions constitute the third through eighth entries, respectively. If the problem is 1- or 2-D, then the region boundary dimensions for the remaining unnecessary coordinate should be zero. For the R-Z geometry type, the fifth and sixth entries should be zero. The region volume must be nonnegative. Although it is usually more convenient to place the overall configuration at the origin, it is not necessary to do so; however, all region boundary dimensions are entered as their distance from the origin. For the axisymmetric geometry, the minimum inner radial dimension is zero. The axial region boundary dimensions need not be positive, but must extend in the positive Z direction.

F16.4.4.3 Card R2

This card must be included with the appropriate R1 card even if it is blank for an OCULAR run. The HEATING data contained on this card are not utilized.

F16.4.5 Material Data Block (Cards M and PC)

A group of cards, consisting of an M card and possibly a PC card, is required to describe each material for a HEATING model. A subset of the HEATING data is processed and used by OCULAR, and only this subset is described below.

F16.4.5.1 Block indicator card

MATERIALS

F16.4.5.2 Card M

a. Material Number (I)

The first entry, an integer, contains the number of the material which is to be described. This entry corresponds to the material specified as entry 2 on card R1. Each different material has a unique number.

b. Material Name (A)

The second entry contains the name of the material. The name, which may consist of up to eight alphanumeric characters, is used to aid in identification of output data. It cannot contain any embedded-blanks and must be terminated by one or more blanks. Note that OCULAR does not access the material properties library of SCALE. However, an asterisk is acceptable as the first character of the material name.

c. HEATING Data

Entries 3, 4, 5, 6, 7, and 8 are HEATING data that are not processed for use by OCULAR. If OCULAR is being used in a stand-alone fashion, these entries should be blank.

d. Phase Change Flag (1)

When employing a HEATING input deck, the ninth entry on the M card is a positive integer indicating that a material can undergo a change-of-phase. OCULAR uses this entry as a flag to indicate that a PC card immediately follows the M card. The entry should be blank for a stand-alone OCULAR run and for materials that do not undergo a phase change.

F16.4.5.3 Card PC

Information on this card is not processed for use by OCULAR. This card must be present if the ninth entry on card M is nonzero.

F16.4.6 X (or R) Grid Data Block (Cards L1 and N1)

Gross lattice data for the X (or R) axis are entered on two sets of cards: the first set specifies the lattice dimensions, and the second indicates the mesh divisions between gross lattice lines. All numbers on the first set (L1 cards) are of the floating-point type and are entered by specifying all of the gross lattice dimensions along the X (or R) axis sequentially on one or more cards. The second set (N1 cards) specify the number (an integer greater than or equal to 1) of equal increments which are between the gross lattice lines whose dimensions are given on the first set of cards. In particular, an entry on an N1 card specifies the number of equal increments between the gross lattice line in the corresponding entry on the L1 card and the gross lattice line immediately following it. Any L1 or N1 card that is a continuation card must contain an "@" in column 1.

F16.4.6.1 Block indicator card

XGRID

F16.4.6.2 Card L1

The L1 cards correspond to the X (or R) coordinate. The total number of entries must not exceed MAXGGL.

F16.4.6.3 Card N1

The N1 cards correspond to the X (or R) coordinate and have one less entry than the L1 cards. The total number of fine lattice lines along the X (or R) axis or the sum of the entries on the N1 card plus 1 must not exceed MAXRFG.

F16.4.7 Z (or ϕ) Grid Data Block (Cards L3 and N3)

The gross lattice data for the Z (or ϕ) axis are formed in the same manner as for the data along the X (or R) axis. Any L3 or N3 card that is a continuation card must contain an "@" in column 1.

F16.4.7.1 Block indicator card

ZGRID

F16.4.7.2 Card L3

The L3 cards correspond to the Z (or ϕ) coordinate. The total number of entries must not exceed MAXGGL.

F16.4.7.3 Card N3

The N3 cards correspond to the Z (or ϕ) coordinate and have one less entry than the L3 card. The total number of fine lattice lines along the Z (or ϕ) axis or the sum of the entries on the N3 card plus 1 must not exceed MAXZFG.

F16.4.8 Enclosure Data Block (Cards EN1 and EN2)

Each enclosure is described by two cards that must appear in pairs. The cards are repeated for each enclosure. There must be at least one enclosure for each problem. Each different enclosure is assigned a unique positive integer. Enclosures may be defined to exist in voided regions or in regions containing nonparticipating materials such as air. The surfaces that comprise the enclosure must be defined such that a complete enclosure is formed. For input purposes, the program treats the centerline of a cylindrical enclosure as a window. Thus, the centerline is a pseudosurface that must be input. This surface is used to define the enclosure, but it is not included in the solution technique. The maximum number of enclosures is MAXENC.

F16.4.8.1 Block indicator card

ENCLOSURES

F16.4.8.2 Card EN1

a. Enclosure Number (I)

The first entry on card EN1 is an integer that contains the number of the enclosure to be defined.

b. Boundary Number (I)

The second entry on the EN1 card contains the boundary number defining the HEATING boundary condition number which will be used to model the radiant heat transfer within the enclosure.

c. Number of Surfaces (I)

The third entry contains the number of surfaces that comprise the enclosure.

F16.4.8.3 Card EN2

The entries on the EN2 card define the surfaces that comprise the enclosure. The total number of entries must correspond to the number specified by the third entry on card EN1. The surface numbers must correspond to surfaces defined in the Surface Data Block.

F16.4.9 Surface Data Block (Card SU)

Surface data are described in a manner similar to regions. The location of the surface is defined by specifying where its endpoints lie in the model. A 2-D surface may be visualized as a line; a 3-D surface as an area. The surface dimension will always be one dimension less than the model's dimension. The input data have been generalized to provide for future implementation HEATING geometries. The number of surfaces cannot exceed MAXSRF.

F16.4.9.1 Block indicator card

SURFACES

F16.4.9.2 Card SU

a. Surface Number (I)

This entry contains the number of the surface to be described.

b. Emissivity of Surface (I)

The second entry of this card indicates the emissivity function number that describes the hemispherical total emissivity of the surface.

c. Surface Dimensions (R)

Dimensions of the surface boundaries are entered as floating-point numbers and are arranged in the following order:

1. smaller dimension of X (or R) surface boundary,
2. larger dimension of X (or R) surface boundary,
3. smaller dimension of Y (or θ) surface boundary,
4. larger dimension of Y (or θ) surface boundary,
5. smaller dimension of Z (or ϕ) surface boundary, and
6. larger dimension of Z (or ϕ) surface boundary.

For the axisymmetric geometry, the Y (or θ) boundary is set to zero by OCULAR. The two X (or R) boundaries must be equal or the two Z (or ϕ) entries must be equal. For a 3-D model, either the X, the Y, or the Z surface boundary dimensions must be equal.

F16.4.10 Emissivity Data Block (Card EM)

Total hemispherical emissive property data are entered on the EM cards. At least one emissivity function must be defined for each problem. The total number of emissivity functions cannot exceed MAXEMS. An area-weighted average emissivity is used for nodal areas comprised of two different enclosure surfaces.

F16.4.10.1 Block indicator card

EMISSIVITIES

F16.4.10.2 Card EM

Each different emissivity function is assigned a unique positive integer. The emissivity function may be expressed as:

$$\varepsilon(x,y,z) = \varepsilon_c F_i(x) F_j(y) F_K(z) \ , \qquad \text{(F16.29)}$$

where $\epsilon(x,y,z)$ is a total hemispherical emissivity at a location (x, y, z) in the model. Since the emissivity associated with a surface may be a function of position, the data card consists of five entries. All of them are integers except the second, which is a floating-point number. The first entry contains the emissivity function number. The second entry contains the constant factor describing the emissivity function. This term corresponds to ϵ_c, in Eq. (F16.29). The remaining entries, which identify analytical or tabular functions *F(x)*, *F(y)*, *F(z)*, contain the *x- (or r-)*, *y- (or θ-)*, and *z- (or ϕ-)* dependent function parameters corresponding to the i, j, and k, respectively, in Eq. F16.29. If the emissivity does not vary along a particular axis of the surface, then the position-dependent function parameter associated with the coordinate must be zero, and the corresponding function value will be set to 1.0 by the code. The emissivity function value may not be set to zero. To model a surface with an emissivity of zero, a small value, such as 1.0×10^{-6} is suggested. A value of 1.0 is recommended to model black surfaces such as holes in the enclosure.

F16.4.11 Analytical Function Data Block (Cards A1 and A2)

Each analytical function, defined as

$$F(v) = A_1 + A_2 v + A_3 v^2 + A_4 \cos(A_5 v) + A_6 e^{A_7 v} + A_8 \sin(A_9 v) + A_{10} \ln(A_{11} v) \qquad \text{(F16.30)}$$

is described by an A1 card and one or more A2 cards. The total number of analytical functions must not exceed MAXANA.

F16.4.11.1 Block indicator card

ANALYTICAL FUNCTIONS

F16.4.11.2 Card A1

The first entry on Card A1 is a unique analytical function number, an integer. The second entry, also an integer, is the number of coefficients, A_i, which are on the A2 cards. If the second entry is blank or zero, then the code assumes that a user-supplied subroutine will be supplied for the emissivity. The second entry cannot exceed MAXPAR which is currently 11.

F16.4.11.3 Card A2

The A2 cards contain up to MAXPAR (currently set to 11) ordered pairs, where each ordered pair is defined as follows: the first element of an ordered pair consists of an integer i; the second element

consists of the value of the coefficient A_i. The integer i must not exceed MAXPAR. The A2 cards will be continued until each coefficient in the analytical function is defined. If the second entry on Card A1 is blank or zero, then the related A2 card is omitted. A continuation card must contain an "@" in column 1.

As an example, the cards which are necessary to describe

$$f(x) = 100.0 \sin(5x) + e^{-3x} \tag{F16.31}$$

are presented in Table F16.4.4.

Table F16.4.4 A1 and A2 cards necessary to describe the function $100.0 \sin(5x) + e^{-3x}$

Entry	1	2	3	4	5	6	7	8	Card
	1	4							A1
	8	100.0	9	5.0	6	1.0	7	-3.0	A2

F16.4.11.4 User-supplied subroutine EMSSVY

The dummy FORTRAN subroutine, EMSSVY, incorporated in OCULAR to model the emissivity of an enclosure surface is presented in Fig. F16.4.1. The user must overwrite this dummy routine with FORTRAN coding designed to describe the emissivity behavior.

F16.4.12 Tabular Function Data Block (Cards T1 and T2)

The tabular function is assumed to be a set of linearly connected points. The function is described by specifying a set of ordered pairs v and $F(v)$. Each ordered pair contains an independent variable and its functional value. A maximum of MAXPRS points is allowed, and linear interpolation is performed between the points. The function is evaluated according to

$$F(v) = \begin{cases} F(v_1) & v < v_i \\ F(v_n) & v > v_n \end{cases} \tag{F16.32}$$

for points outside the defined domain. The total number of tabular functions must not exceed MAXTBL.

F16.4.12.1 Block indicator card

TABULAR FUNCTIONS

F16.4.12.2 Card T1

The first entry on the T1 card (an integer) is a unique tabular function number. The number of points (an integer) is the second entry.

```
      subroutine emssvy(rvalue,r,th,z,value,number)
c*****************************************************************
c
c
 when it is referenced, this subroutine must be supplied by the user to
c
 evaluate the user-supplied function defining the emissivity.
c
 this parameter may be positional-dependent
c
 the variables in the argument list are defined in the following table.
c
 (y) indicates that the variable has been defined when the routine
c
 is called.  (n) indicates that it has not been defined.
c
c      variable      type definition
c      rvalue        real value of the user-supplied function
c                    evaluating the emissivity property
c                    this value must be computed by the
c                    routine (n)
c      r      real   x or r coordinate of the enclosure nodely)
c      th     real   y or theta coordinate of the enclosure nodely)
c      z      real   z or phi coordinate of the enclosure nodely)
c      value  real constant value of the parameter
c                    which appears on its respective input
c                    card (1) if it is nonzero
c                    if = zero, value=1.0 (y)
c                    name in calling sequence - emis(nemvyt)
c      number        integer index of parameter being evaluated (y)
c                    name in calling sequence - nemvyt
c
c*****************************************************************
      double precision rvalue,r,th,z,value
      common /iounit/ ibin,iecho,in,io,ireclg,ibout
c
c insert algorithm to compute the value of the emissivity function
c 'number' associated with node 'n' here
c
       write(io,9000)
9000   format('1*** error *** you have called subroutine emssvy',
     *  ' which is a user-supplied subroutine',/
     *  ' *** error *** this subroutine has not been supplied.',
     *  ' either supply subroutine emssvy',/
     *  ' *** error *** or correct the input data to',
     *  ' not reference this subroutine.')
       stop
       end
```

Figure F16.4.1 User-supplied subroutine EMSSVY

F16.4.12.3 Card T2

The T2 cards contain the ordered pairs, all floating-point numbers. The first member of the pair is the independent variable; the second, the function value. A continuation card must contain an "@" in column 1.

An example of an acceptable tabular function is presented in Fig. F16.4.2. The input for this example is presented in Table F16.4.5.

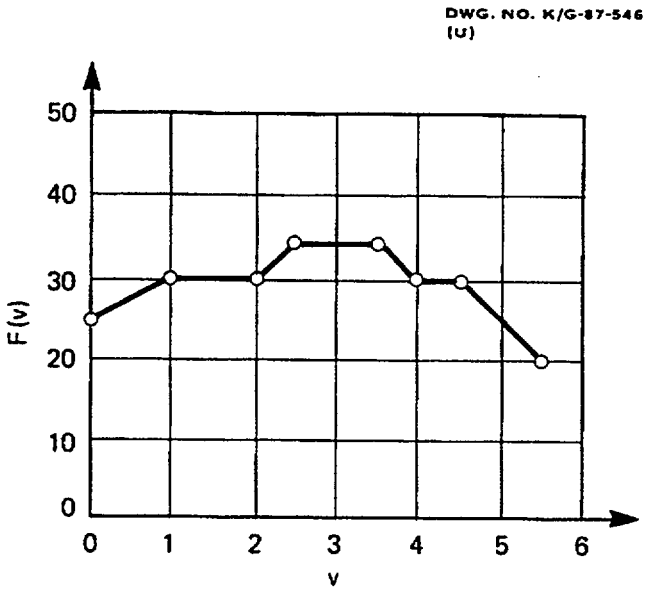


Figure F16.4.2 Tabular function example

Table F16.4.5 T1 and T2 cards necessary to describe the tabular function depicted in Fig. F16.4.2

Entry	1	2	3	4	5	6	7	8	Card
	1	8							T1
	0.0	25.0	1.0	30.0	2.0	30.0	2.5	35.0	T2
	@3.5	35.0	4.0	30.0	4.5	30.0	5.5	20.0	T2

F16.4.13 Data Block Termination Card

Each problem (or case) must be terminated with a data block termination card. This card consists of a percent sign (%) in column 1, followed by a blank in column 2.

F16.4.14 Summary and Format of Input

A summary of the format and information needed to prepare the input data deck is presented in Table F16.4.6. Columns 73 through 80 of each card are reserved for identification. This input format assumes that an OCULAR input data deck is not being enhanced for use as a HEATING input data deck. Thus all HEATING dependent input data are reflected as blank entries. The user should supply zeros in

these entries when OCULAR is used in a stand-alone fashion. The table provides proposed identification names for the cards, and parenthetically, the section of the report where more information can be found. The first line in each format box includes the variable name actually used in the program, and in the parentheses, the variable type. The rest of the box includes a short explanation of the input in that box.

Table F16.4.6 Summary of OCULAR input data

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 (F16.4.3.1)
	NGEOM (I) Geometry Type Cylindrical Cartesian 1 r-θ-z 6 x-y-z 2 r-θ 7 x-y 3 r-z 8 x-z 4 r 9 x 5 z Spherical 10 r 11 r-ϕ 12 r-θ-ϕ								Card 2 (F16.4.3.2)
REGIONS									
NOREG(I) Region number	MATL(I) Region material number. (Card M) Enter zero for gap region	RRIN(R) Smaller X or R region dimension	RROT(R) Larger X or R region dimension	THLT(R) Smaller Y or θ region dimension	THRT(R) Larger Y or θ region dimension	ZZBK(R) Smaller Z or ϕ region dimension	ZZFR(R) Larger Z or θ region dimension		Card R1 (F16.4.4.2) Maximum MAXREG
	NOTE: Although not used, this card cannot be omitted.								Card R2 (F16.4.4.3)
MATERIALS									
MAT(I) Material number	MATNAM(A) Material name							MCP(I) Phase change flag. Zero implies no phase changes	Card M (F16.4.5.2) Maximum MAXMAT
	NOTE: This card is only present for materials with change of phase capabilities. There can only be MAXCP such materials.								Card PC (F16.4.5.3)

Table F16.4.6 (continued)

Entry 1	Entry 2	Entry 3	Entry 4	Entry 5	Entry 6	Entry 7	Entry 8	Entry 9	Card
XGRID									
RG(R) Smallest X or R gross lattice line dimension	RG(R) Next X or R gross lattice line dimension	RG(R) Next X or R gross lattice line dimension	NOTE: Maximum of MAXGGL.						Card L1
NDRG(I) Number of divisions between corresponding X or R gross line and the following line	NDRG(I)	NDRG(I)	NOTE: Must have one less entry than in L1 cards. Maximum of MAXRFG.						Card N1 (F16.4.6.3)
ZGRID									
ZG(R)	NOTE: Same as L1 cards except for Z or ϕ direction. Maximum of MAXGGL.								Card L3 (F16.4.7.2)
NDZG(I)	NOTE: Same as N1 cards except for Z or ϕ direction. Maximum of MAXZFG.								Card N3 (F16.4.7.3)
ENCLOSURES									
NENCLT(I) Enclosure number	NENBCT(I) Boundary condition number describing heat transfer. Mechanism in enclosure	NSRF(I) Number of surfaces defining enclosure							Card EN1 (F16.4.8.2) Maximum MAXENC
NSUURFT(I) Surface number comprising enclosure.	NSURFT(I) Next surface number.	NSURFT(I) Next surface number.							Card EN2 (F16.4.8.3)
SURFACES									
NENSRT(I) Surface number	NEMST(I) Surface emissivity function number	ENRRIN(R) Smaller X or R surface dimension	ENRROT(R) Larger X or R surface dimension	ENTHLT(R) Smaller Y or θ surface dimension	ENTHRT(R) Larger Y or θ surface dimension	ENZZBK(R) Smaller Z or ϕ surface dimension	ENZZFK(R) Larger Z or θ surface dimension		Card SU (F16.4.9.2) Maximum MAXSRF
EMISSIVITIES									
NEMVYT(I) Emissivity number	EMIS(R) Emissivity if constant	NEMR(I) X- or R-dependent function number	NEMTH(I) Y- or θ-dependent function number	NEMZ(I) Z- or ϕ-dependent function number					Card EM (F16.4.10.2) Maximum MAXEMS

Table F16.4.6 (continued)

Entry 1	Entry 2	Entry 3	Entry 4	Entry 5	Entry 6	Entry 7	Entry 8	Entry 9	Card
ANALYTICAL FUNCTIONS									
NANALT(I) Analytical function number	NPARM(I) Number of coeffi-cients on A2 cards. Maximum MAXPAR	NOTE: A1 and A2 establish the mth analytical function in the form $F_m(V) = A_{m,1} + A_{m,2}V + A_{m,3}V^2 + A_{m,4}\cos(A_{m,5}V) + A_{m,6}\exp(A_{m,7}V) + A_{m,8}\sin(A_{m,9}V) + A_{m,10}\ln(A_{m,11}V)$.							Card A1 (F16.4.11.2) Maximum MAXANA
NPRM(I) Coefficient index I	A(NPRM)(R) Coefficient $A_{m,I}$	NPRM(I) Coefficient index i	A(NPRM)(R) Coefficient $A_{m,i}$	NOTE: To indicate a user-supplied function, the second entry on Card A1 is zero and Card A2 is omitted.					Card A2 (F16.4.11.3)
TABULAR FUNCTIONS									
NTABL(I) Tabular function number	NTBPRS(I) Number of pairs in table. Maximum MAXPRS	NOTE: The code uses linear interpolation to obtain the value of this function.							Card T1 (F16.4.12.2) Maximum MAXTBL
ARG(1)(R) Independent variable	VAL(1)(R) Corresponding function value	ARG(2)(R) independent variable	VAL(2)(R) Corresponding function value	NOTE: Maximum of MAXPRS pairs.					Card T2 (F16.4.12.3)
% Data Block Termination Card									(F16.4.13)

F16.5 OUTPUT DESCRIPTION FOR OCULAR

F16.5.1 General

The output generated by the OCULAR program is best illustrated by an example, thus the reader is referred to Appendix F16.A for an example of the actual output. The code automatically lists the input data card images, tabulates the input data with descriptive headings, and lists model information generated by the code from the input data. If the problem runs to completion, the code prints the emitting node number, the absorbing node number, the exchange (configuration) factor between the emitting and the absorbing node and the sum of the factors for each emitting node. The sum of the factors should equal the emissivity of the emitting node. If the difference due to discretization error is unacceptable, the mesh should be refined and the problem rerun.

In addition to the exchange factor results, the code writes the data necessary for input to HEATING. These data consist of an emitting (base) node number, the number of exchange factor connections made to this node, the boundary condition number describing the heat transfer mechanism present in the enclosure, a reciprocity flag, the absorbing node numbers, and the connector value. The connector is defined as the product of the emitting node's area and the exchange factor between the two nodes. In addition to including this information as standard output, the code writes a binary data set to unit No. 12 to be read as input by HEATING. Any errors encountered in the input data which cause problem termination are included as program output.

F16.5.2 Input Return

If the input data are read in card image form, OCULAR automatically lists the card images with the card columns indicated every tenth card and with each card image numbered. Since this capability serves to document the input data exactly as these data were supplied to the code, this feature assists the user in the location of input data errors identified by the program.

The standard output contains a summary of the capabilities of the current version of OCULAR, along with a heading identifying the version of the code, the date and time of the run, and the computer on which the job was executed. This summary is followed by tables indicating the maximum number of parameters that may be utilized during the run. These values are obtained from the BLOCK DATA subprogram and any data entered through the OPTION data block. The values of parameters from the first three data cards are then tabulated along with the input/output logical unit numbers available for the program. The code then tabulates data it generates from the input data. These data include the fine lattice lines along each axis, the orientation of the enclosure surfaces, the total number of nodes in the problem, and the enclosure nodes. Additionally, the code writes a summary of the dimensional characteristics of the enclosure nodes.

F16.5.3 Exchange Factors

The user-oriented output from OCULAR lists the emitting node, the absorbing node, and the exchange factor between the two nodes. Additionally, the code prints the sum of the exchange factors from each emitting node to all absorbing nodes. This sum should equal the emissivity of the emitting node. If the difference due to discretization error is unacceptable, the mesh should be refined and the problem should be rerun. OCULAR reports the total number of connectors produced. This value may be used as the value for MAXCON in the HEATING data set.

F16.5.4 HEATING Input

OCULAR output data are written on a binary data set to unit 94 for input into HEATING as unformatted input through the CONNECTOR data block. This data set contains the emitting node number, the number of absorbing nodes to which the emitting node has connections, the boundary condition number describing the heat transfer mechanism present, a reciprocity flag, the absorbing node numbers, and the connector value. The connector is defined as the product of the exchange factor and the emitting node's area.

F16.6 REFERENCES

1. J. A. Wiebelt, *Engineering Radiation Heat Transfer*, Holt, Rinehart, and Winston, New York, 1966.

2. S. N. Rea, "Rapid Method for Determining Concentric Cylinder Radiation View Factors," *AIAA Journal 13*(8), 1122 (August 1975).

3. C. P. Minning, "Radiation Shape Factors between End Wall and Outer Wall of Concentric Tubular Enclosures," *AIAA Journal 17(12)*, 1407 (December 1979).

4. C. P. Minning, "Shape Factors between Coaxial Annular Disks Separated by a Solid Cylinder," *AIAA Journal 17(3)*, 318 (March 1979).

5. R. Siegel and J. R. Howell, *Thermal Radiation Heat Transfer*, McGraw-Hill, New York, 1981.

6. J. R. Howell, *A Catalog of Radiation Configuration Factors*, McGraw-Hill, New York, 1982.

7. P. T. Williams, "Sample Problems for RADFAC/HOTTEL/ TRUMP Radiation Calculations," Union Carbide Corp. Letter Report, Union Carbide Corp., Nucl. Div., Oak Ridge Gaseous Diffusion Plant, January 10, 1983.

8. G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Inc., 1977.

F16.A SAMPLE PROBLEM

The use of the OCULAR program is best illustrated by example. This sample problem is presented for instructive purposes. It is not intended to represent an actual engineering problem, but rather to introduce the novice user to the capabilities of the program. Graybody radiation exchange factors are computed for both an annular and cylindrical axisymmetric enclosure.

The problem of interest is illustrated in Fig. F16.A.1. The OCULAR input data set is presented in Fig. F16.A.2. The user-supplied routine EMSSVY employed in this model is presented in Fig. F16.A.3.

The model consists of three materials — iron, air, and stainless steel. The air is assumed not to participate in the radiant exchange. The surfaces of the steel and the iron are assumed to be treated to exhibit emissive characteristics dependent upon surface orientation. The emissivity for horizontally oriented surfaces present in the enclosure is referred to as the radial emissivity of a material. The emissivity for vertically oriented surfaces is referred to as the axial emissivity of a material. The emissivity data for iron are assumed constant. The emissivity of stainless steel varies as a function of axial position on the enclosure surface. These data are presented in Table F16.A.1 below.

Table F16.A.1 Hemispherical total emissive property data

Material	Iron	Stainless steel
Radial emissivity	0.3	—
Axial emissivity	0.8	0.1 + 0.01 z

Illustrated in Fig. F16.A.1 are the locations of regions, enclosures, and surfaces contained in the OCULAR input. Region numbers are contained in circles, enclosure numbers in squares, and surface numbers in triangles. Regions 1, 4, 5, and 7 contain material 1, defined to be iron. Regions 2 and 6 contain material 2, defined to be air. Regions 3 and 8 contain material 3, defined to be stainless steel.

Multidimensional radiant exchange is present in regions 2 and 6. These regions are defined to be enclosures 1 and 2, respectively.

Enclosure 1 is a cylindrical enclosure composed of four surfaces. Note that one of these surfaces describes the centerline. In the calculation of exchange factors, it is necessary for a complete enclosure to be input to OCULAR. The centerline surface is actually a pseudosurface that must be input in order to describe a complete enclosure, but which is eliminated from the problem solution by the program. The emissivity of the upper- and lower-ring surfaces is constant. The emissivity of the shell surface varies as a function of axial position on the shell. This positional-dependent emissivity is input using the user-supplied routine capability of OCULAR.

Enclosure 2 is an annular enclosure composed of six surfaces. The enclosure consists of a cylinder, lower ring, upper ring, and shell. The cylinder is comprised of three enclosure surfaces which exhibit the different emissive properties of the iron and steel composing the surface. The positional-dependent emissivity present on a portion of the cylinder is input using the analytical function

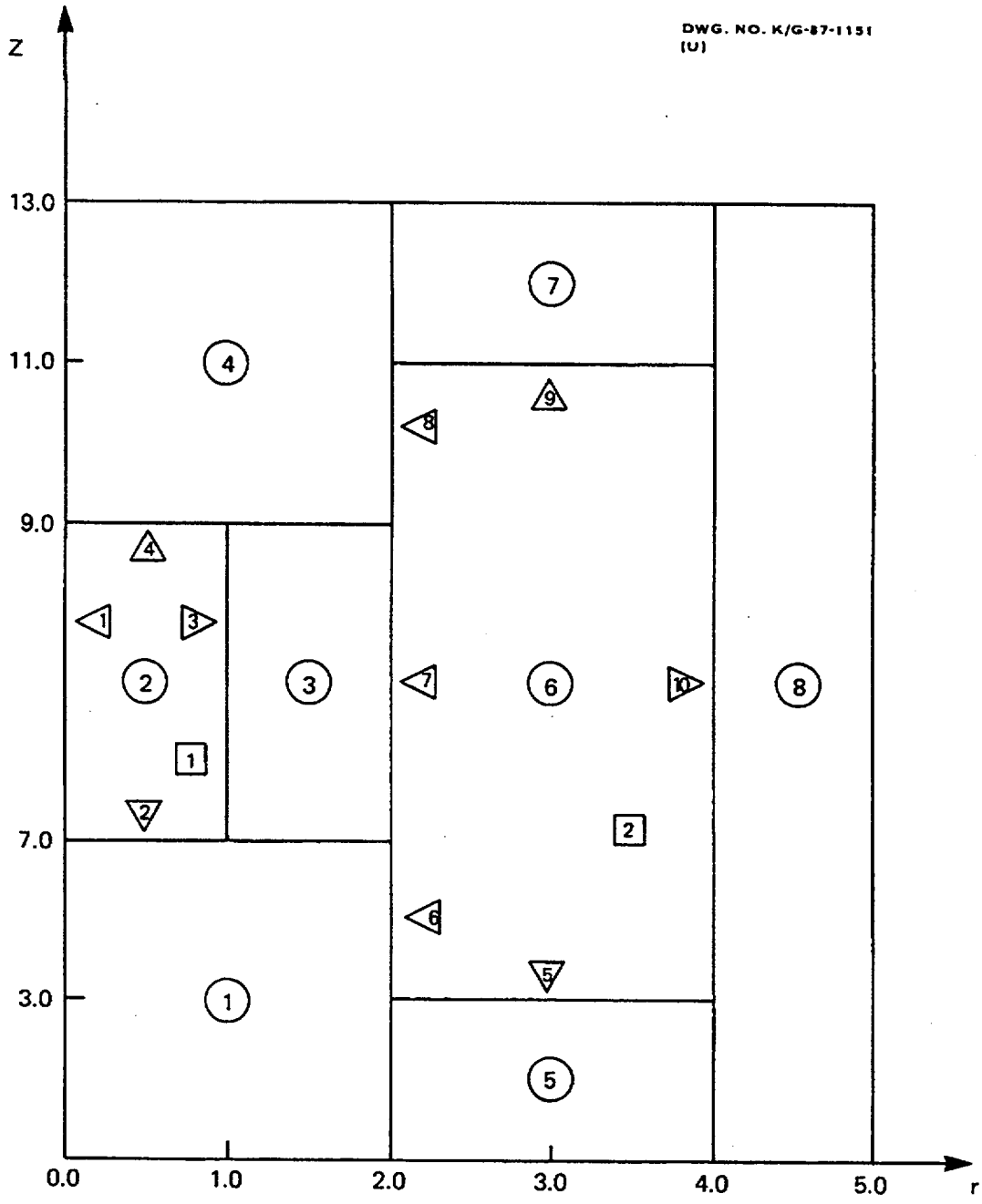


Figure F16.A.1 Sample problem model geometry

```
option
maxrfg=53,maxtfg=5,maxzfg=53,maxpbt=50,maxsrf=10,
@maxnea=38,maxreg=8,maxems=6,maxpts=200
ocular sample problem
500 3 .30000000d+02 0 1 0 0
region
1  1  0.0  2.0  0.0  0.0  0.0  7.0
0
2  2  0.0  1.0  0.0  0.0  7.0  9.0
0
3  3  1.0  2.0  0.0  0.0  7.0  9.0
0
4  1  0.0  2.0  0.0  0.0  9.0 13.0
0
5  1  2.0  4.0  0.0  0.0  0.0  3.0
0
6  2  2.0  4.0  0.0  0.0  3.0 11.0
0
7  1  2.0  4.0  0.0  0.0 11.0 13.0
0
8  3  4.0  5.0  0.0  0.0  0.0 13.0
0
materials
1  iron
2  air
3  stainlss
xgrid
0.0   1.0   2.0   4.0   5.0
    4     1     6     1
zgrid
0.0   3.0   7.0   9.0   11.0   13.0
    1     3     5     3      1
analytical functions
1  0
2  2
1  0.1   2  0.01
tabular functions
1  2
3.0  0.13   11.0  0.21
enclosures
1   1   4
1   2   3   4
2   1   6
5   6   7   8   9   10
surfaces
1   6     0.0  0.0  0.0  0.0  7.0  9.0
2   1     0.0  1.0  0.0  0.0  7.0  7.0
3   2     1.0  1.0  0.0  0.0  7.0  9.0
4   1     0.0  1.0  0.0  0.0  9.0  9.0
5   1     2.0  4.0  0.0  0.0  3.0  3.0
6   3     2.0  2.0  0.0  0.0  3.0  7.0
7   4     2.0  2.0  0.0  0.0  7.0  9.0
8   3     2.0  2.0  0.0  0.0  9.0 11.0
9   1     2.0  4.0  0.0  0.0 11.0 11.0
10  5     4.0  4.0  0.0  0.0  3.0 11.0
emissivity
1  0.3
2  1.0  0  0  1
3  0.8
4  1.0  0  0  2
5  1.0  0  0 -1
6  1.0
%
```

Figure F16.A.2 Sample problem OCULAR input

```
      subroutine emssvy(rvalue,r,th,z,value,number)
c*****************************************************************
c
c when it is referenced, this subroutine must be supplied by the user to
c evaluate the user-supplied function defining the emissivity
c this parameter may be positional-dependent
c the variables in the argument list are defined in the following table
c (y) indicates that the variable has been defined when the routine
c is called.  (n) indicates that it has not been defined
c
c    variable      type      definition
c    rvalue       real      value of the user-supplied function
c                           evaluating the emissivity property
c                           this value must be computed by the
c                           routine (n)
c    r            real      x or r coordinate of the enclosure node(y)
c    th           real      y or th coordinate of the enclosure node(y)
c    z            real      z or p coordinate of the enclosure node(y)
c    value        real      constant value of the parameter
c                           which appears on its respective input
c                           card (i) if it is nonzero
c                           if = zero, value=1.0 (y)
c                           name in calling sequence - emis(nemvyt)
c    number       integer   index of parameter being evaluated (y)
c                           name in calling sequence - nemvyt
c
c*****************************************************************
c
      double precision rvalue,r,th,z,value
      rvalue=value*(0.10d0+z/100.0d0)
      return
      end
```

Figure F16.A.3 User-supplied routine EMSSVY

capability of OCULAR. The emissivity of the lower and upper rings is constant. The emissivity of the shell is positional-dependent and is input using the tabular function capability.

In developing an input data set to produce exchange factors that will later be input to HEATING, it is necessary to define the boundary condition number that will be used by HEATING to describe the heat transfer mechanism which is present between nodes input in the CONNECTOR data block. In this sample problem, it is assumed that boundary condition 1, input as the second entry on the EN1 card, will be used to describe the radiation heat transfer mechanism present between nodes. As a side note, the Stefan-Boltzmann constant would generally be entered as the coefficient for radiation in the HEATING input data set for boundary condition 1.

The output for this sample problem is presented in Table F16.A.2. The printed output contains an echo of the input, summary tables of the input data, descriptions of each enclosure, and the surfaces composing it. Additional output is provided describing the location and emissivity of each enclosure surface nodal area. This output is provided for the purpose of validation of the model. It is recommended that the user run HEATING to receive a nodal map of the problem. Note that corner nodes appear twice in the enclosure node summary. The area associated with a corner node is composed of a vertical and horizontal area. Exchange factors for these areas are computed and reported separately. Reciprocity is used to reduce the number of connectors input to HEATING.

Table F16.A.2 OCULAR sample problem output

```
written by: c.b. bryan
contact:
      g.e. giles
      computational physics and engineering
      oak ridge national laboratory
      building 6011, mail stop number 6415
      post office box 2008
      oak ridge, tennessee 37831-6415
      phone - 423-574-8667

 definition of input/output units

 name    no   unit number definitions
 ibin    95   binary input data set (optional)
 iecho    5   initial input data set
 in      13   standard input data set
 io       6   standard output data set
 ireclg  80   record length in single word for binary d ata set on unit ibin (maximum of 80)
 ibout   12   unformatted connector data set

 print input card images with card columns indicated every 10th card--

 card
 no./col. 1.......10........20........30........40........50........60........70........80
     1    option
     2    maxrfg=53,maxtfg=5,maxzfg=53,maxpbt=50,maxsrf=10,
     2    @maxnea=38,maxreg=8,maxems=6,maxpts=200
     3    ocular sample problem
     4    500 3 .30000000D+02 0 1 0 0
     5    region
     6    1  1  0.0  2.0  0.0  0.0  0.0  7.0
     7    0
     8    2  2  0.0  1.0  0.0  0.0  7.0  9.0
     9    0
 no./col. 1.......10........20........30........40........50........60........70........80
    10    3  3  1.0  2.0  0.0  0.0  7.0  9.0
    11    0
    12    4  1  0.0  2.0  0.0  0.0  9.0 13.0
    13    0
    14    5  1  2.0  4.0  0.0  0.0  0.0  3.0
    15    0
    16    6  2  2.0  4.0  0.0  0.0  3.0 11.0
    17    0
    18    7  1  2.0  4.0  0.0  0.0 11.0 13.0
    19    0
 no./col. 1.......10........20........30........40........50........60........70........80
    20    8  3  4.0  5.0  0.0  0.0  0.0 13.0
    21    0
```

Table F16.A.2 (continued)

```
    22    materials
    23    1  iron
    24    2  air
    25    3  stainlss
    26    xgrid
    27    0.0   1.0   2.0   4.0   5.0
    28        4     1     6     1
    29    zgrid
no./col. 1.......10........20........30........40........50........60........70........80
    30    0.0   3.0   7.0   9.0   11.0   13.0
    31        1     3     5     3      1
    32    analytical functions
    33    1  0
    34    2  2
    35    1  0.1   2  0.01
    36    tabular functions
    37    1  2
    38    3.0  0.13   11.0  0.21
    39    enclosures
no./col. 1.......10........20........30........40........50........60........70........80
    40    1   1   4
    41    1   2   3   4
    42    2   1   6
    43    5   6   7   8   9   10
    44    surfaces
    45    1   6     0.0  0.0  0.0  0.0  7.0  9.0
    46    2   1     0.0  1.0  0.0  0.0  7.0  7.0
    47    3   2     1.0  1.0  0.0  0.0  7.0  9.0
    48    4   1     0.0  1.0  0.0  0.0  9.0  9.0
    49    5   1     2.0  4.0  0.0  0.0  3.0  3.0
no./col. 1.......10........20........30........40........50........60........70........80
    50    6   3     2.0  2.0  0.0  0.0  3.0  7.0
    51    7   4     2.0  2.0  0.0  0.0  7.0  9.0
    52    8   3     2.0  2.0  0.0  0.0  9.0 11.0
    53    9   1     2.0  4.0  0.0  0.0 11.0 11.0
    54    10  5     4.0  4.0  0.0  0.0  3.0 11.0
    55    emissivity
    56    1  0.3
    57    2  1.0  0  0  1
    58    3  0.8
    59    4  1.0  0  0  2
no./col. 1.......10........20........30........40........50........60........70........80
    60    5  1.0  0  0 -1
    61    6  1.0
    62    %
    63    %
    64    end
```

Table F16.A.2 (continued)

```
the variably dimensioned arrays require     82814 single words or  324k bytes of core.

                        release date:  05/12/93                    fortran 77 - first edition

code: ocular   05/12/93
date: 06/14/93
time: 14:53:53
jobnam: geg
computer: unix

 job description -- ocular sample problem
 geometry type number                                        3 (or rz  )
definition of input/output units
iecho     5   initial input data set
in       13   standard input data set
io        6   standard output data set
ibout    12   unformatted connector data set

number of parameters specified by the input data

  number of regions                                          8
  number of materials                                        3
  number of analytical functions                             2
  number of tabular functions                                1
  number of points in gross x or r lattice                   5
  number of points in gross y of t lattice                   1
  number of points in gross z or p lattice                   6
  number of enclosures in the model                          2
  number of surfaces                                        10
  number of emissivities                                     6

summary of region data
          ****************************** dimensions ********************************
region    left-x or     right-x or    lower-y or  upper-y or     rear-z or     front-z or
number     inner-r       outer-r      left-theta  right-theta    upper phi      lower phi
   1        .0000         2.0000         .0000       .0000         .0000         7.0000
   2        .0000         1.0000         .0000       .0000        7.0000         9.0000
   3       1.0000         2.0000         .0000       .0000        7.0000         9.0000
   4        .0000         2.0000         .0000       .0000        9.0000        13.0000
   5       2.0000         4.0000         .0000       .0000         .0000         3.0000
   6       2.0000         4.0000         .0000       .0000        3.0000        11.0000
   7       2.0000         4.0000         .0000       .0000       11.0000        13.0000
   8       4.0000         5.0000         .0000       .0000         .0000        13.0000
```

Table F16.A.2 (continued)

region number	material number	initial temp	heat generation	boundary numbers lf-x in-r	rt-x ot-r	lo-y lf-t	up-y rt-t	rr-z sm-p	ft-z lg-p
1	1	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0
3	3	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	2	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0
8	3	0	0	0	0	0	0	0	0

gross lattices and number of increments

r or x					
.000000	1.000000	2.000000	4.000000	5.000000	
4	1	6	1		
phi or z					
.000000	3.000000	7.000000	9.000000	11.000000	13.000000
1	3	5	3	1	

summary of material data

material number	material name
1	iron
2	air
3	stainlss

Table F16.A.2 (continued)

```
summary of analytic functions
     f(v) = a(1) + a(2)*v + a(3)*v**2 + a(4)*cos(a(5)*v) + a(6)*exp(a(7)*v) + a(8)*sin(a(9)*v) + a(10)*log(a(11)*v)
 no.            a(1)       a(2)       a(3)       a(4)       a(5)       a(6)       a(7)       a(8)       a(9)       a(10)      a(11)
           analytic function no.    1   is user supplied
  2          1.000D-01  1.000D-02  0.000D+00  0.000D+00  0.000D+00  0.000D+00  0.000D+00  0.000D+00  0.000D+00  0.000D+00
0.000D+00


summary of tabular functions
         table number -         1      number of pairs -       2
              argument                       value
           3.00000000D+00                 1.30000000D-01
           1.10000000D+01                 2.10000000D-01
```

summary of enclosure data

enclosure number	boundary condition describing the heat transfer mechanism in the enclosure	surface numbers comprising the ***** enclosure *****					
1	1	1	2	3	4		
2	1	5	6	7	8	9	10

summary of surface data

surface number	emissivity number	dimensions: left-x or inner-r	right-x or outer-r	lower-y or left-theta	upper-y or right-theta	rear-z or lower phi	front-z or upper phi
1	6	.0000	.0000	.0000	.0000	7.0000	9.0000
2	1	.0000	1.0000	.0000	.0000	7.0000	7.0000
3	2	1.0000	1.0000	.0000	.0000	7.0000	9.0000
4	1	.0000	1.0000	.0000	.0000	9.0000	9.0000
5	1	2.0000	4.0000	.0000	.0000	3.0000	3.0000
6	3	2.0000	2.0000	.0000	.0000	3.0000	7.0000
7	4	2.0000	2.0000	.0000	.0000	7.0000	9.0000
8	3	2.0000	2.0000	.0000	.0000	9.0000	11.0000
9	1	2.0000	4.0000	.0000	.0000	11.0000	11.0000
10	5	4.0000	4.0000	.0000	.0000	3.0000	11.0000

summary of emissivity data

number	emissivity	position-dependent function numbers: x or r	y or th	z or p
1	3.00000D-01	0	0	0
2	1.00000D+00	0	0	1
3	8.00000D-01	0	0	0
4	1.00000D+00	0	0	2
5	1.00000D+00	0	0	-1
6	1.00000D+00	0	0	0

summary of fine lattice lines
x or r fine lattice lines

1	.000000	2	.250000	3	.500000	4	.750000	5	1.000000
6	2.000000	7	2.333333	8	2.666667	9	3.000000	10	3.333333
11	3.666667	12	4.000000	13	5.000000				

Table F16.A.2 (continued)

```
z or phi fine lattice lines

         1      .000000          2     3.000000          3     4.333333          4     5.666667          5     7.000000
         6     7.400000          7     7.800000          8     8.200000          9     8.600000         10     9.000000
        11     9.666667         12    10.333333         13    11.000000         14    13.000000

***** job description *****ocular sample 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
***** number of warnings encountered in the input data for this case was --    0

summary of surface orientations
surface number     1 is a z (or phi)  surface
surface number     2 is an x (or r)  surface
surface number     3 is a z (or phi)  surface
surface number     4 is an x (or r)  surface
surface number     5 is an x (or r)  surface
surface number     6 is a z (or phi)  surface
surface number     7 is a z (or phi)  surface
surface number     8 is a z (or phi)  surface
surface number     9 is an x (or r)  surface
surface number    10 is a z (or phi)  surface

summary of enclosure surface connections
enclosure number    1
surface connected to surface
     1                 2
     1                 4
     2                 1
     2                 3
     3                 2
     3                 4
     4                 1
     4                 3
enclosure number    2
surface connected to surface
     5                 6
     5                10
     6                 5
     6                 7
     7                 6
     7                 8
```

```
     8                7
     8                9
     9                8
     9               10
    10                5
    10                9

this model contains  182 nodes.

summary of enclosure nodes
enclosure number    1

base surface nodes
 node      nodal         radial       axial         area boundaries         emissivity
number     area         location     location     minimum r     maximum r
   53      .049087      .000000      7.000000      .000000       .125000      .300000
   54      .392699      .250000      7.000000      .125000       .375000      .300000
   55      .785398      .500000      7.000000      .375000       .625000      .300000
   56     1.178097      .750000      7.000000      .625000       .875000      .300000
   57      .736311     1.000000      7.000000      .875000      1.000000      .300000

upper surface nodes
 node      nodal         radial       axial         area boundaries         emissivity
number     area         location     location     minimum r     maximum r
  118      .049087      .000000      9.000000      .000000       .125000      .300000
  119      .392699      .250000      9.000000      .125000       .375000      .300000
  120      .785398      .500000      9.000000      .375000       .625000      .300000
  121     1.178097      .750000      9.000000      .625000       .875000      .300000
  122      .736311     1.000000      9.000000      .875000      1.000000      .300000

shell surface nodes
 node      nodal         radial       axial         area boundaries         emissivity
number     area         location     location     minimum z     maximum z
   57     1.256637     1.000000      7.000000      7.000000      7.200000      .170000
   70     2.513274     1.000000      7.400000      7.200000      7.600000      .174000
   83     2.513274     1.000000      7.800000      7.600000      8.000000      .178000
   96     2.513274     1.000000      8.200000      8.000000      8.400000      .182000
  109     2.513274     1.000000      8.600000      8.400000      8.800000      .186000
  122     1.256637     1.000000      9.000000      8.800000      9.000000      .190000
```

Table F16.A.2 (continued)

ocular 05/12/93 ocular sample problem
geg 14:53:53
06/14/93

********* summary of exchange factor data**********

enclosure number 1

heat transfer mechanism specified by boundary condition 1.

node i	exchange factor sum	exchange factors from node i to node j											
53	3.00000D-01	53	9.23053D-04	54	7.39448D-03	55	1.48043D-02	56	2.18573D-02	57	1.28627D-02	57	1.33311D-02
		70	3.64666D-02	83	3.85296D-02	96	3.53164D-02	109	3.21274D-02	118	1.20961D-03	119	9.56428D-03
		120	1.85481D-02	121	2.63995D-02	122	1.55690D-02	122	1.50966D-02				
54	3.00000D-01	53	9.24310D-04	54	7.41038D-03	55	1.48703D-02	56	2.20481D-02	57	1.30308D-02	57	1.37262D-02
		70	3.74176D-02	83	3.83694D-02	96	3.48563D-02	109	3.17331D-02	118	1.19609D-03	119	9.46269D-03
		120	1.83733D-02	121	2.61863D-02	122	1.54590D-02	122	1.49361D-02				
55	3.00000D-01	53	9.25272D-04	54	7.43515D-03	55	1.50234D-02	56	2.25959D-02	57	1.36075D-02	57	1.52132D-02
		70	3.99174D-02	83	3.75008D-02	96	3.35553D-02	109	3.06825D-02	118	1.16139D-03	119	9.19931D-03
		120	1.79094D-02	121	2.56077D-02	122	1.51578D-02	122	1.45080D-02				
56	3.00000D-01	53	9.10720D-04	54	7.34937D-03	55	1.50639D-02	56	2.36185D-02	57	1.54437D-02	57	2.08592D-02
		70	4.18863D-02	83	3.47391D-02	96	3.12242D-02	109	2.89106D-02	118	1.10532D-03	119	8.76705D-03
		120	1.71228D-02	121	2.45951D-02	122	1.46240D-02	122	1.37802D-02				
57	3.00000D-01	53	8.57513D-04	54	6.94975D-03	55	1.45146D-02	56	2.47099D-02	57	1.93835D-02	57	3.43904D-02
		70	3.61126D-02	83	3.10345D-02	96	2.87806D-02	109	2.70657D-02	118	1.04752D-03	119	8.31693D-03
		120	1.62865D-02	121	2.34985D-02	122	1.40415D-02	122	1.30099D-02				
57	1.70000D-01	53	5.20744D-04	54	4.28943D-03	55	9.50825D-03	56	1.95555D-02	57	2.01506D-02	57	1.15187D-02
		70	1.95657D-02	83	1.72759D-02	96	1.55350D-02	109	1.41830D-02	118	5.30528D-04	119	4.19713D-03
		120	8.14231D-03	121	1.15655D-02	122	6.79416D-03	122	6.66747D-03				
70	1.74000D-01	53	7.12237D-04	54	5.84650D-03	55	1.24742D-02	56	1.96342D-02	57	1.05799D-02	57	9.78287D-03
		70	2.06916D-02	83	1.93472D-02	96	1.74429D-02	109	1.58448D-02	118	5.88848D-04	119	4.65083D-03
		120	8.98113D-03	121	1.26548D-02	122	7.37094D-03	122	7.39715D-03				
83	1.78000D-01	53	7.52531D-04	54	5.99522D-03	55	1.17190D-02	56	1.62839D-02	57	9.09214D-03	57	8.63794D-03
		70	1.93472D-02	83	2.09892D-02	96	1.99087D-02	109	1.81812D-02	118	6.74683D-04	119	5.32467D-03
		120	1.02374D-02	121	1.42442D-02	122	8.16696D-03	122	8.44497D-03				
96	1.82000D-01	53	6.89774D-04	54	5.44630D-03	55	1.04860D-02	56	1.46363D-02	57	8.43181D-03	57	7.76749D-03
		70	1.74429D-02	83	1.99087D-02	96	2.18772D-02	109	2.10227D-02	118	7.67291D-04	119	6.11008D-03
		120	1.19272D-02	121	1.65200D-02	122	9.17595D-03	122	9.79021D-03				
109	1.86000D-01	53	6.27488D-04	54	4.95830D-03	55	9.58827D-03	56	1.35519D-02	57	7.92939D-03	57	7.09151D-03
		70	1.58448D-02	83	1.81812D-02	96	2.10227D-02	109	2.34588D-02	118	7.57301D-04	119	6.21441D-03
		120	1.32471D-02	121	2.08043D-02	122	1.11484D-02	122	1.15742D-02				
118	3.00000D-01	53	1.20961D-03	54	9.56869D-03	55	1.85823D-02	56	2.65277D-02	57	1.57128D-02	57	1.35815D-02
		70	3.01490D-02	83	3.45438D-02	96	3.92853D-02	109	3.87738D-02	118	9.19318D-04	119	7.36004D-03
		120	1.47094D-02	121	2.16398D-02	122	1.26710D-02	122	1.47660D-02				

```
  119   3.00000D-01     53  1.19554D-03   54  9.46269D-03   55  1.83986D-02   56  2.63011D-02   57  1.55942D-02   57  1.34308D-02
                        70  2.97653D-02   83  3.40779D-02   96  3.91045D-02  109  3.97722D-02  118  9.20004D-04  119  7.37130D-03
                       120  1.47656D-02  121  2.18145D-02  122  1.28276D-02  122  1.51980D-02
  120   3.00000D-01     53  1.15926D-03   54  9.18663D-03   55  1.79094D-02   56  2.56842D-02   57  1.52686D-02   57  1.30277D-02
                        70  2.87396D-02   83  3.27597D-02   96  3.81671D-02  109  4.23908D-02  118  9.19337D-04  119  7.38281D-03
                       120  1.48907D-02  121  2.23146D-02  122  1.33684D-02  122  1.68312D-02
  121   3.00000D-01     53  1.09998D-03   54  8.72878D-03   55  1.70718D-02   56  2.45951D-02   57  1.46866D-02   57  1.23365D-02
                        70  2.69968D-02   83  3.03877D-02   96  3.52428D-02  109  4.43825D-02  118  9.01658D-04  119  7.27150D-03
                       120  1.48764D-02  121  2.32343D-02  122  1.51114D-02  122  2.30761D-02
  122   3.00000D-01     53  1.03793D-03   54  8.24481D-03   55  1.61683D-02   56  2.33983D-02   57  1.40415D-02   57  1.15954D-02
                        70  2.51595D-02   83  2.78766D-02   96  3.13206D-02  109  3.80533D-02  118  8.44730D-04  119  6.84139D-03
                       120  1.42597D-02  121  2.41782D-02  122  1.88814D-02  122  3.80984D-02
  122   1.90000D-01     53  5.89713D-04   54  4.66753D-03   55  9.06752D-03   56  1.29189D-02   57  7.62297D-03   57  6.66747D-03
                        70  1.47943D-02   83  1.68899D-02   96  1.95804D-02  109  2.31484D-02  118  5.76796D-04  119  4.74938D-03
                       120  1.05195D-02  121  2.16339D-02  122  2.23233D-02  122  1.42500D-02

                                  ********** summary of heating6 input data***********
connectors
    53   16    1    1
     53  4.531026D-05   54  3.629756D-04   55  7.267068D-04
@    56  1.072917D-03   57  6.313958D-04   57  6.543868D-04
@    70  1.790048D-03   83  1.891318D-03   96  1.733591D-03
@   109  1.577049D-03  118  5.937658D-05  119  4.694857D-04
@   120  9.104778D-04  121  1.295882D-03  122  7.642423D-04
@   122  7.410549D-04
    54   15    1    1
     54  2.910049D-03   55  5.839551D-03   56  8.658273D-03
@    57  5.117173D-03   57  5.390261D-03   70  1.469387D-02
@    83  1.506764D-02   96  1.368804D-02  109  1.246157D-02
@   118  4.697019D-04  119  3.715988D-03  120  7.215159D-03
@   121  1.028335D-02  122  6.070740D-03  122  5.865385D-03
    55   14    1    1
     55  1.179933D-02   56  1.774677D-02   57  1.068729D-02
@    57  1.194842D-02   70  3.135108D-02   83  2.945307D-02
@    96  2.635427D-02  109  2.409794D-02  118  9.121566D-04
@   119  7.225123D-03  120  1.406601D-02  121  2.011225D-02
@   122  1.190488D-02  122  1.139458D-02
    56   13    1    1
     56  2.782484D-02   57  1.819420D-02   57  2.457412D-02
@    70  4.934619D-02   83  4.092598D-02   96  3.678509D-02
@   109  3.405955D-02  118  1.302174D-03  119  1.032843D-02
@   120  2.017231D-02  121  2.897546D-02  122  1.722845D-02
@   122  1.623442D-02
    57   12    1    1
     57  1.427229D-02   57  2.532205D-02   70  2.659006D-02
```

Table F16.A.2 (continued)

```
@   83  2.285104D-02   96  2.119146D-02  109  1.992874D-02
@  118  7.713006D-04  119  6.123844D-03  120  1.199194D-02
@  121  1.730223D-02  122  1.033893D-02  122  9.579306D-03
   57   11    1    1
    57  1.447486D-02   70  2.458703D-02   83  2.170950D-02
@   96  1.952184D-02  109  1.782290D-02  118  6.666817D-04
@  119  5.274271D-03  120  1.023193D-02  121  1.453365D-02
@  122  8.537797D-03  122  8.378592D-03
   70   10    1    1
    70  5.200361D-02   83  4.862491D-02   96  4.383877D-02
@  109  3.982222D-02  118  1.479938D-03  119  1.168880D-02
@  120  2.257205D-02  121  3.180486D-02  122  1.852519D-02
@  122  1.859107D-02
   83    9    1    1
    83  5.275164D-02   96  5.003600D-02  109  4.569421D-02
@  118  1.695663D-03  119  1.338235D-02  120  2.572941D-02
@  121  3.579969D-02  122  2.052582D-02  122  2.122453D-02
   96    8    1    1
    96  5.498346D-02  109  5.283571D-02  118  1.928413D-03
@  119  1.535631D-02  120  2.997639D-02  121  4.151940D-02
@  122  2.306167D-02  122  2.460547D-02
  109    7    1    1
   109  5.895851D-02  118  1.903305D-03  119  1.561851D-02
@  120  3.329362D-02  121  5.228694D-02  122  2.801909D-02
@  122  2.908912D-02
  118    6    1    1
   118  4.512690D-05  119  3.612849D-04  120  7.220458D-04
@  121  1.062240D-03  122  6.219841D-04  122  7.248237D-04
  119    5    1    1
   119  2.894704D-03  120  5.798449D-03  121  8.566536D-03
@  122  5.037392D-03  122  5.968242D-03
  120    4    1    1
   120  1.169515D-02  121  1.752582D-02  122  1.049954D-02
@  122  1.321916D-02
  121    3    1    1
   121  2.737229D-02  122  1.780265D-02  122  2.718592D-02
  122    2    1    1
   122  1.390261D-02  122  2.805226D-02
  122    1    1    1
   122  1.790710D-02
```

Table F16.A.2 (continued)

summary of enclosure nodes
enclosure number 2
cylinder surface nodes

node number	nodal area	radial location	axial location	area boundaries minimum z	area boundaries maximum z	emissivity
19	8.377580	2.000000	3.000000	3.000000	3.666667	.800000
32	16.755161	2.000000	4.333333	3.666667	5.000000	.800000
45	16.755161	2.000000	5.666667	5.000000	6.333333	.800000
58	10.890855	2.000000	7.000000	6.333333	7.200000	.655077
71	5.026548	2.000000	7.400000	7.200000	7.600000	.174000
84	5.026548	2.000000	7.800000	7.600000	8.000000	.178000
97	5.026548	2.000000	8.200000	8.000000	8.400000	.182000
110	5.026548	2.000000	8.600000	8.400000	8.800000	.186000
123	6.702064	2.000000	9.000000	8.800000	9.333333	.570500
136	8.377580	2.000000	9.666667	9.333333	10.000000	.800000
149	8.377580	2.000000	10.333333	10.000000	10.666667	.800000
162	4.188790	2.000000	11.000000	10.666667	11.000000	.800000

base surface nodes

node number	nodal area	radial location	axial location	area boundaries minimum r	area boundaries maximum r	emissivity
19	2.181662	2.000000	3.000000	2.000000	2.166667	.300000
20	4.886922	2.333333	3.000000	2.166667	2.500000	.300000
21	5.585054	2.666667	3.000000	2.500000	2.833333	.300000
22	6.283185	3.000000	3.000000	2.833333	3.166667	.300000
23	6.981317	3.333333	3.000000	3.166667	3.500000	.300000
24	7.679449	3.666667	3.000000	3.500000	3.833333	.300000
25	4.101524	4.000000	3.000000	3.833333	4.000000	.300000

upper surface nodes

node number	nodal area	radial location	axial location	area boundaries minimum r	area boundaries maximum r	emissivity
162	2.181662	2.000000	11.000000	2.000000	2.166667	.300000
163	4.886922	2.333333	11.000000	2.166667	2.500000	.300000
164	5.585054	2.666667	11.000000	2.500000	2.833333	.300000
165	6.283185	3.000000	11.000000	2.833333	3.166667	.300000
166	6.981317	3.333333	11.000000	3.166667	3.500000	.300000
167	7.679449	3.666667	11.000000	3.500000	3.833333	.300000
168	4.101524	4.000000	11.000000	3.833333	4.000000	.300000

shell surface nodes

node number	nodal area	radial location	axial location	area boundaries minimum z	area boundaries maximum z	emissivity
25	16.755161	4.000000	3.000000	3.000000	3.666667	.130000
38	33.510322	4.000000	4.333333	3.666667	5.000000	.143333

Table F16.A.2 (continued)

51	33.510322	4.000000	5.666667	5.000000	6.333333	.156667
64	21.781709	4.000000	7.000000	6.333333	7.200000	.165333
77	10.053096	4.000000	7.400000	7.200000	7.600000	.174000
90	10.053096	4.000000	7.800000	7.600000	8.000000	.178000
103	10.053096	4.000000	8.200000	8.000000	8.400000	.182000
116	10.053096	4.000000	8.600000	8.400000	8.800000	.186000
129	13.404129	4.000000	9.000000	8.800000	9.333333	.191333
142	16.755161	4.000000	9.666667	9.333333	10.000000	.196667
155	16.755161	4.000000	10.333333	10.000000	10.666667	.203333
168	8.377580	4.000000	11.000000	10.666667	11.000000	.210000

Table F16.A.2 (continued)

ocular 05/12/93 ocular sample problem
geg 14:53:53
06/14/93

********* summary of exchange factor data**********

enclosure number 2

heat transfer mechanism specified by boundary condition 1.

node i	exchange factor sum	node j	factor	node j	factor	node j	factor	node j	factor	node j	factor	node j	factor
19	8.00000D-01	19	8.70522D-02	19	2.78361D-02	20	3.98851D-02	21	2.73511D-02	22	2.21879D-02	23	2.10817D-02
		24	2.24168D-02	25	1.22969D-02	25	3.26076D-02	32	1.18061D-01	38	5.78237D-02	45	8.68303D-02
		51	3.60056D-02	58	3.44527D-02	64	1.72022D-02	71	3.57946D-03	77	7.23448D-03	84	3.32275D-03
		90	6.83011D-03	97	3.10136D-03	103	6.44277D-03	110	2.91356D-03	116	6.05997D-03	123	1.09198D-02
		129	7.54867D-03	136	1.74604D-02	142	8.71161D-03	149	1.61760D-02	155	8.38124D-03	162	7.51869D-03
		162	1.68490D-03	163	4.34955D-03	164	5.38748D-03	165	6.28354D-03	166	7.05037D-03	167	7.67711D-03
		168	4.02458D-03	168	4.25056D-03								
19	2.99999D-01	19	1.06891D-01	19	1.55742D-03	20	3.01703D-03	21	3.08172D-03	22	3.34040D-03	23	3.67553D-03
		24	3.96680D-03	25	2.05108D-03	25	4.77560D-03	32	3.10090D-02	38	1.45713D-02	45	2.55864D-02
		51	1.27131D-02	58	1.16327D-02	64	6.78746D-03	71	1.28217D-03	77	2.92737D-03	84	1.22578D-03
		90	2.79107D-03	97	1.17330D-03	103	2.65507D-03	110	1.12652D-03	116	2.51769D-03	123	4.31618D-03
		129	3.16669D-03	136	7.07235D-03	142	3.70209D-03	149	6.69157D-03	155	3.61477D-03	162	3.11367D-03
		162	6.96418D-04	163	1.87926D-03	164	2.37196D-03	165	2.78767D-03	166	3.14056D-03	167	3.43117D-03
		168	1.80469D-03	168	1.85509D-03								
20	3.00000D-01	19	6.83745D-02	19	1.34689D-03	20	2.87684D-03	21	3.19491D-03	22	3.60494D-03	23	4.03826D-03
		24	4.39435D-03	25	2.27967D-03	25	5.33106D-03	32	4.90653D-02	38	1.63003D-02	45	3.04631D-02
		51	1.39042D-02	58	1.32711D-02	64	7.47968D-03	71	1.45281D-03	77	3.24795D-03	84	1.38686D-03
		90	3.10849D-03	97	1.32697D-03	103	2.96693D-03	110	1.27450D-03	116	2.82175D-03	123	4.88786D-03
		129	3.56014D-03	136	8.02470D-03	142	4.17683D-03	149	7.62381D-03	155	4.08954D-03	162	3.58317D-03
		162	8.34328D-04	163	2.19161D-03	164	2.72816D-03	165	3.18338D-03	166	3.57101D-03	167	3.89118D-03
		168	2.04401D-03	168	2.09889D-03								
21	3.00000D-01	19	4.10266D-02	19	1.20380D-03	20	2.79555D-03	21	3.30818D-03	22	3.85236D-03	23	4.40064D-03
		24	4.88005D-03	25	2.57571D-03	25	6.29246D-03	32	6.03600D-02	38	1.80326D-02	45	3.54693D-02
		51	1.43439D-02	58	1.46144D-02	64	7.73607D-03	71	1.57748D-03	77	3.38235D-03	84	1.49859D-03
		90	3.24980D-03	97	1.42950D-03	103	3.11222D-03	110	1.37047D-03	116	2.96835D-03	123	5.25048D-03
		129	3.75572D-03	136	8.61860D-03	142	4.42068D-03	149	8.20018D-03	155	4.34180D-03	162	3.87381D-03
		162	9.17361D-04	163	2.37625D-03	164	2.93732D-03	165	3.41599D-03	166	3.82547D-03	167	4.16572D-03
		168	2.18834D-03	168	2.23186D-03								
22	3.00000D-01	19	2.95839D-02	19	1.15986D-03	20	2.80384D-03	21	3.42432D-03	22	4.07490D-03	23	4.76297D-03
		24	5.46525D-03	25	2.99040D-03	25	7.91875D-03	32	6.13902D-02	38	1.93342D-02	45	3.88166D-02
		51	1.40591D-02	58	1.54287D-02	64	7.64927D-03	71	1.64553D-03	77	3.37330D-03	84	1.55653D-03
		90	3.25576D-03	97	1.48067D-03	103	3.12962D-03	110	1.41717D-03	116	2.99416D-03	123	5.42425D-03
		129	3.79985D-03	136	8.90221D-03	142	4.48828D-03	149	8.47814D-03	155	4.42442D-03	162	4.01658D-03
		162	9.57240D-04	163	2.46143D-03	164	3.03233D-03	165	3.52197D-03	166	3.94315D-03	167	4.29598D-03
		168	2.25873D-03	168	2.28030D-03								

Table F16.A.2 (continued)

23	3.00000D-01	19	2.52980D-02	19	1.14860D-03	20	2.82678D-03	21	3.52051D-03	22	4.28668D-03	23	5.19333D-03
		24	6.33814D-03	25	3.69679D-03	25	1.09675D-02	32	5.92289D-02	38	1.95613D-02	45	4.03240D-02
		51	1.31993D-02	58	1.57431D-02	64	7.34022D-03	71	1.66697D-03	77	3.27383D-03	84	1.57306D-03
		90	3.17720D-03	97	1.49444D-03	103	3.06780D-03	110	1.42960D-03	116	2.94587D-03	123	5.47224D-03
		129	3.75219D-03	136	8.98829D-03	142	4.45086D-03	149	8.57357D-03	155	4.40786D-03	162	4.07105D-03
		162	9.72890D-04	163	2.49065D-03	164	3.06301D-03	165	3.55667D-03	166	3.98416D-03	167	4.34604D-03
		168	2.28842D-03	168	2.28005D-03								
24	3.00000D-01	19	2.44547D-02	19	1.12693D-03	20	2.79640D-03	21	3.54913D-03	22	4.47157D-03	23	5.76194D-03
		24	7.79804D-03	25	5.00029D-03	25	1.68746D-02	32	5.66883D-02	38	1.71876D-02	45	3.99044D-02
		51	1.18715D-02	58	1.55191D-02	64	6.86558D-03	71	1.64206D-03	77	3.10627D-03	84	1.55041D-03
		90	3.03470D-03	97	1.47440D-03	103	2.94615D-03	110	1.41227D-03	116	2.84189D-03	123	5.41552D-03
		129	3.63631D-03	136	8.91765D-03	142	4.33688D-03	149	8.53078D-03	155	4.32079D-03	162	4.06048D-03
		162	9.72080D-04	163	2.48185D-03	164	3.05011D-03	165	3.54339D-03	166	3.97416D-03	167	4.34369D-03
		168	2.29205D-03	168	2.24599D-03								
25	3.00000D-01	19	2.51170D-02	19	1.09100D-03	20	2.71620D-03	21	3.50735D-03	22	4.58105D-03	23	6.29242D-03
		24	9.36225D-03	25	6.46097D-03	25	2.36165D-02	32	5.49686D-02	38	1.31219D-02	45	3.84578D-02
		51	1.06874D-02	58	1.50249D-02	64	6.43700D-03	71	1.59639D-03	77	2.94910D-03	84	1.51117D-03
		90	2.89772D-03	97	1.44063D-03	103	2.82644D-03	110	1.38316D-03	116	2.73735D-03	123	5.31768D-03
		129	3.51698D-03	136	8.78445D-03	142	4.21537D-03	149	8.43021D-03	155	4.22284D-03	162	4.02096D-03
		162	9.63990D-04	163	2.45808D-03	164	3.02085D-03	165	3.51204D-03	166	3.94416D-03	167	4.31920D-03
		168	2.28369D-03	168	2.20507D-03								
25	1.30000D-01	19	1.63038D-02	19	6.21823D-04	20	1.55489D-03	21	2.09749D-03	22	2.96953D-03	23	4.56981D-03
		24	7.73421D-03	25	5.78112D-03	25	3.65743D-03	32	2.70671D-02	38	5.26631D-03	45	1.53540D-02
		51	4.19050D-03	58	5.46195D-03	64	2.37528D-03	71	5.60316D-04	77	1.05454D-03	84	5.20707D-04
		90	1.01807D-03	97	4.87947D-04	103	9.76820D-04	110	4.60843D-04	116	9.31217D-04	123	1.73898D-03
		129	1.17484D-03	136	2.80606D-03	142	1.37432D-03	149	2.63167D-03	155	1.33650D-03	162	1.24195D-03
		162	2.90380D-04	163	7.40117D-04	164	9.03325D-04	165	1.03893D-03	166	1.15023D-03	167	1.23648D-03
		168	6.42125D-04	168	6.78380D-04								
32	8.00000D-01	19	5.90305D-02	19	4.03763D-03	20	1.43107D-02	21	2.01200D-02	22	2.30213D-02	23	2.46787D-02
		24	2.59821D-02	25	1.34559D-02	25	2.70671D-02	32	1.23643D-01	38	6.53800D-02	45	1.03892D-01
		51	5.09373D-02	58	4.27310D-02	64	2.24513D-02	71	4.41326D-03	77	8.94369D-03	84	4.05726D-03
		90	8.24459D-03	97	3.74277D-03	103	7.63463D-03	110	3.47215D-03	116	7.07494D-03	123	1.28232D-02
		129	8.68581D-03	136	2.01357D-02	142	9.86154D-03	149	1.83285D-02	155	9.33496D-03	162	8.44981D-03
		162	1.84935D-03	163	4.73154D-03	164	5.84059D-03	165	6.79889D-03	166	7.60906D-03	167	8.24939D-03
		168	4.30198D-03	168	4.67816D-03								
38	1.43333D-01	19	1.44559D-02	19	9.48652D-04	20	2.37712D-03	21	3.00544D-03	22	3.62517D-03	23	4.07527D-03
		24	3.93882D-03	25	1.60607D-03	25	2.63315D-03	32	3.26900D-02	38	5.68122D-03	45	2.32129D-02
		51	5.07007D-03	58	7.63678D-03	64	2.91186D-03	71	7.39998D-04	77	1.29037D-03	84	6.69402D-04
		90	1.24098D-03	97	6.13723D-04	103	1.18438D-03	110	5.69265D-04	116	1.12178D-03	123	2.11057D-03
		129	1.40297D-03	136	3.34058D-03	142	1.62070D-03	149	3.07924D-03	155	1.55183D-03	162	1.44202D-03
		162	3.30724D-04	163	8.40496D-04	164	1.02133D-03	165	1.16784D-03	166	1.28313D-03	167	1.36542D-03
		168	7.01933D-04	168	7.76221D-04								
45	8.00000D-01	19	4.34152D-02	19	3.33157D-03	20	8.88508D-03	21	1.18231D-02	22	1.45562D-02	23	1.68017D-02
		24	1.82895D-02	25	9.41414D-03	25	1.53540D-02	32	1.03892D-01	38	4.64258D-02	45	1.12355D-01
		51	6.72965D-02	58	5.45964D-02	64	3.79485D-02	71	5.93951D-03	77	1.44539D-02	84	5.53380D-03

Table F16.A.2 (continued)

		90	1.26674D-02	97	5.11651D-03	103	1.11846D-02	110	4.71824D-03	116	9.95367D-03	123	1.71848D-02
		129	1.17632D-02	136	2.63579D-02	142	1.28424D-02	149	2.33249D-02	155	1.17443D-02	162	1.05492D-02
		162	2.17808D-03	163	5.56396D-03	164	6.89841D-03	165	8.06106D-03	166	9.02793D-03	167	9.74708D-03
		168	5.04430D-03	168	5.76034D-03								
51	1.56667D-01	19	9.00140D-03	19	8.27677D-04	20	2.02769D-03	21	2.39066D-03	22	2.63609D-03	23	2.74986D-03
		24	2.72056D-03	25	1.30809D-03	25	2.09525D-03	32	2.54687D-02	38	5.07007D-03	45	3.36482D-02
		51	6.17384D-03	58	1.43838D-02	64	4.03645D-03	71	1.34717D-03	77	1.83737D-03	84	1.16219D-03
		90	1.78075D-03	97	1.01626D-03	103	1.70239D-03	110	9.04139D-04	116	1.60713D-03	123	3.22126D-03
		129	1.99319D-03	136	4.89760D-03	142	2.26827D-03	149	4.36874D-03	155	2.12954D-03	162	2.01791D-03
		162	4.48587D-04	163	1.13061D-03	164	1.36394D-03	165	1.54688D-03	166	1.68160D-03	167	1.76390D-03
		168	8.93637D-04	168	1.04524D-03								
58	6.55077D-01	19	2.65021D-02	19	2.33028D-03	20	5.95496D-03	21	7.49454D-03	22	8.90119D-03	23	1.00918D-02
		24	1.09429D-02	25	5.65843D-03	25	8.40299D-03	32	6.57400D-02	38	2.34978D-02	45	8.39944D-02
		51	4.42580D-02	58	4.90929D-02	64	4.02780D-02	71	5.88105D-03	77	1.83510D-02	84	5.73145D-03
		90	1.65169D-02	97	5.47039D-03	103	1.43749D-02	110	5.14176D-03	116	1.23661D-02	123	1.88832D-02
		129	1.39762D-02	136	2.88232D-02	142	1.44956D-02	149	2.50569D-02	155	1.26887D-02	162	1.11249D-02
		162	2.14812D-03	163	5.51182D-03	164	6.91812D-03	165	8.16924D-03	166	9.20141D-03	167	9.92496D-03
		168	5.10281D-03	168	6.07806D-03								
64	1.65333D-01	19	6.61621D-03	19	6.79834D-04	20	1.67813D-03	21	1.98361D-03	22	2.20652D-03	23	2.35264D-03
		24	2.42056D-03	25	1.21210D-03	25	1.82714D-03	32	1.72702D-02	38	4.47979D-03	45	2.91911D-02
		51	6.20992D-03	58	2.01390D-02	64	5.17830D-03	71	2.28407D-03	77	2.53480D-03	84	2.04052D-03
		90	2.50377D-03	97	1.76911D-03	103	2.42263D-03	110	1.52560D-03	116	2.29945D-03	123	5.20145D-03
		129	2.84962D-03	136	7.51342D-03	142	3.21639D-03	149	6.42336D-03	155	2.97694D-03	162	2.91117D-03
		162	6.24126D-04	163	1.56171D-03	164	1.87441D-03	165	2.11229D-03	166	2.27472D-03	167	2.35380D-03
		168	1.17572D-03	168	1.43913D-03								
71	1.74000D-01	19	5.96577D-03	19	5.56499D-04	20	1.41246D-03	21	1.75275D-03	22	2.05692D-03	23	2.31523D-03
		24	2.50871D-03	25	1.30261D-03	25	1.86772D-03	32	1.47109D-02	38	4.93332D-03	45	1.97984D-02
		51	8.98110D-03	58	1.27423D-02	64	9.89764D-03	71	1.63214D-03	77	5.47119D-03	84	1.65488D-03
		90	5.41563D-03	97	1.63392D-03	103	4.96610D-03	110	1.57672D-03	116	4.33311D-03	123	5.90833D-03
		129	4.83374D-03	136	9.12194D-03	142	4.85083D-03	149	7.91324D-03	155	4.09965D-03	162	3.47685D-03
		162	6.37732D-04	163	1.64940D-03	164	2.10006D-03	165	2.50849D-03	166	2.84440D-03	167	3.07068D-03
		168	1.57205D-03	168	1.92666D-03								
77	1.74000D-01	19	6.02873D-03	19	6.35280D-04	20	1.57887D-03	21	1.87908D-03	22	2.10831D-03	23	2.27350D-03
		24	2.37285D-03	25	1.20319D-03	25	1.75756D-03	32	1.49062D-02	38	4.30122D-03	45	2.40898D-02
		51	6.12458D-03	58	1.98802D-02	64	5.49207D-03	71	2.73560D-03	77	2.99420D-03	84	2.69491D-03
		90	3.04060D-03	97	2.46900D-03	103	2.97935D-03	110	2.16352D-03	116	2.85374D-03	123	7.28454D-03
		129	3.55707D-03	136	1.01818D-02	142	4.02018D-03	149	8.41239D-03	155	3.70582D-03	162	3.74995D-03
		162	7.81361D-04	163	1.94694D-03	164	2.33040D-03	165	2.61507D-03	166	2.79644D-03	167	2.86325D-03
		168	1.41476D-03	168	1.77775D-03								
84	1.78000D-01	19	5.53792D-03	19	5.32025D-04	20	1.34833D-03	21	1.66510D-03	22	1.94566D-03	23	2.18480D-03
		24	2.36868D-03	25	1.23307D-03	25	1.73569D-03	32	1.35242D-02	38	4.46268D-03	45	1.84460D-02
		51	7.74794D-03	58	1.24181D-02	64	8.84227D-03	71	1.65488D-03	77	5.38983D-03	84	1.72477D-03
		90	5.79833D-03	97	1.74842D-03	103	5.70989D-03	110	1.72682D-03	116	5.20925D-03	123	6.60787D-03
		129	5.91005D-03	136	1.03716D-02	142	5.86767D-03	149	9.04484D-03	155	4.83747D-03	162	3.95197D-03
		162	6.96180D-04	163	1.81831D-03	164	2.34913D-03	165	2.83621D-03	166	3.23522D-03	167	3.49644D-03

Table F16.A.2 (continued)

		168	1.78454D-03	168	2.23782D-03								
90	1.78000D-01	19	5.69175D-03	19	6.05700D-04	20	1.51107D-03	21	1.80544D-03	22	2.03485D-03	23	2.20639D-03
		24	2.31818D-03	25	1.18223D-03	25	1.69678D-03	32	1.37410D-02	38	4.13658D-03	45	2.11124D-02
		51	5.93584D-03	58	1.78933D-02	64	5.42484D-03	71	2.70782D-03	77	3.04060D-03	84	2.89917D-03
		90	3.27295D-03	97	2.85395D-03	103	3.27695D-03	110	2.61524D-03	116	3.16567D-03	123	8.94802D-03
		129	3.97564D-03	136	1.23561D-02	142	4.51813D-03	149	9.95669D-03	155	4.16991D-03	162	4.37498D-03
		162	8.89288D-04	163	2.21009D-03	164	2.64119D-03	165	2.95447D-03	166	3.14050D-03	167	3.18516D-03
		168	1.55855D-03	168	1.99261D-03								
97	1.82000D-01	19	5.16894D-03	19	5.09243D-04	20	1.29011D-03	21	1.58833D-03	22	1.85083D-03	23	2.07561D-03
		24	2.25256D-03	25	1.17551D-03	25	1.62649D-03	32	1.24759D-02	38	4.09149D-03	45	1.70550D-02
		51	6.77505D-03	58	1.18525D-02	64	7.66616D-03	71	1.63392D-03	77	4.93799D-03	84	1.74842D-03
		90	5.70791D-03	97	1.82162D-03	103	6.10385D-03	110	1.84694D-03	116	5.98009D-03	123	7.25572D-03
		129	7.12200D-03	136	1.16653D-02	142	7.15981D-03	149	1.02954D-02	155	5.78937D-03	162	4.48121D-03
		162	7.53569D-04	163	2.00002D-03	164	2.63731D-03	165	3.22621D-03	166	3.70380D-03	167	4.00677D-03
		168	2.03824D-03	168	2.63078D-03								
103	1.82000D-01	19	5.36898D-03	19	5.76187D-04	20	1.44226D-03	21	1.72901D-03	22	1.95601D-03	23	2.13042D-03
		24	2.25053D-03	25	1.15315D-03	25	1.62803D-03	32	1.27244D-02	38	3.94795D-03	45	1.86409D-02
		51	5.67462D-03	58	1.55729D-02	64	5.24903D-03	71	2.48305D-03	77	2.97935D-03	84	2.85495D-03
		90	3.27695D-03	97	3.05192D-03	103	3.47232D-03	110	3.00208D-03	116	3.42725D-03	123	1.07646D-02
		129	4.34783D-03	136	1.50072D-02	142	4.98891D-03	149	1.18409D-02	155	4.62852D-03	162	5.11448D-03
		162	1.00517D-03	163	2.49309D-03	164	2.97767D-03	165	3.32159D-03	166	3.50590D-03	167	3.51110D-03
		168	1.69498D-03	168	2.20583D-03								
110	1.86000D-01	19	4.85593D-03	19	4.88942D-04	20	1.23909D-03	21	1.52275D-03	22	1.77147D-03	23	1.98556D-03
		24	2.15764D-03	25	1.12862D-03	25	1.53614D-03	32	1.15738D-02	38	3.79510D-03	45	1.57275D-02
		51	6.02760D-03	58	1.11405D-02	64	6.61095D-03	71	1.57672D-03	77	4.32705D-03	84	1.72682D-03
		90	5.23048D-03	97	1.84694D-03	103	6.00416D-03	110	1.92446D-03	116	6.38694D-03	123	7.79136D-03
		129	8.23985D-03	136	1.29223D-02	142	8.70235D-03	149	1.16311D-02	155	7.02140D-03	162	5.05618D-03
		162	8.07043D-04	163	2.20005D-03	164	2.98370D-03	165	3.70109D-03	166	4.26828D-03	167	4.61509D-03
		168	2.33939D-03	168	3.13568D-03								
116	1.86000D-01	19	5.04998D-03	19	5.46374D-04	20	1.37169D-03	21	1.64908D-03	22	1.87135D-03	23	2.04574D-03
		24	2.17089D-03	25	1.11680D-03	25	1.55203D-03	32	1.17916D-02	38	3.73927D-03	45	1.65894D-02
		51	5.35710D-03	58	1.33966D-02	64	4.98215D-03	71	2.16655D-03	77	2.85374D-03	84	2.60463D-03
		90	3.16567D-03	97	2.99005D-03	103	3.42725D-03	110	3.19347D-03	116	3.58147D-03	123	1.23857D-02
		129	4.65347D-03	136	1.80725D-02	142	5.41306D-03	149	1.41814D-02	155	5.07219D-03	162	6.01675D-03
		162	1.12659D-03	163	2.79347D-03	164	3.34458D-03	165	3.72982D-03	166	3.90966D-03	167	3.85066D-03
		168	1.82253D-03	168	2.41471D-03								
123	5.70500D-01	19	1.36497D-02	19	1.40501D-03	20	3.56407D-03	21	4.37540D-03	22	5.08523D-03	23	5.70025D-03
		24	6.20528D-03	25	3.25431D-03	25	4.34746D-03	32	3.20581D-02	38	1.05529D-02	45	4.29621D-02
		51	1.61063D-02	58	3.06851D-02	64	1.69047D-02	71	4.43125D-03	77	1.09268D-02	84	4.95590D-03
		90	1.34220D-02	97	5.44179D-03	103	1.61469D-02	110	5.84352D-03	116	1.85785D-02	123	2.45454D-02
		129	2.67561D-02	136	4.25056D-02	142	3.15385D-02	149	3.95615D-02	155	2.65259D-02	162	1.72525D-02
		162	2.58121D-03	163	7.46075D-03	164	1.05652D-02	165	1.32201D-02	166	1.51630D-02	167	1.62849D-02
		168	8.22030D-03	168	1.17165D-02								

Table F16.A.2 (continued)

129	1.91333D-01	19	4.71792D-03	19	5.15412D-04	20	1.29797D-03	21	1.56488D-03	22	1.78118D-03	23	1.95427D-03
		24	2.08330D-03	25	1.07616D-03	25	1.46855D-03	32	1.08573D-02	38	3.50742D-03	45	1.47040D-02
		51	4.98298D-03	58	1.13557D-02	64	4.63064D-03	71	1.81265D-03	77	2.66780D-03	84	2.21627D-03
		90	2.98173D-03	97	2.67075D-03	103	3.26087D-03	110	3.08994D-03	116	3.49011D-03	123	1.33780D-02
		129	4.87019D-03	136	2.17161D-02	142	5.87722D-03	149	1.76818D-02	155	5.60884D-03	162	7.40225D-03
		162	1.27034D-03	163	3.16412D-03	164	3.83534D-03	165	4.32069D-03	166	4.52089D-03	167	4.34444D-03
		168	1.98041D-03	168	2.67488D-03								
136	8.00000D-01	19	1.74604D-02	19	1.84176D-03	20	4.68108D-03	21	5.74573D-03	22	6.67666D-03	23	7.49024D-03
		24	8.17451D-03	25	4.30072D-03	25	5.61212D-03	32	4.02714D-02	38	1.33623D-02	45	5.27158D-02
		51	1.95904D-02	58	3.74701D-02	64	1.95349D-02	71	5.47316D-03	77	1.22181D-02	84	6.22294D-03
		90	1.48274D-02	97	6.99916D-03	103	1.80087D-02	110	7.75335D-03	116	2.16870D-02	123	3.40045D-02
		129	3.47457D-02	136	6.24705D-02	142	4.87128D-02	149	6.17292D-02	155	4.69294D-02	162	2.74062D-02
		162	3.86654D-03	163	1.32472D-02	164	1.94856D-02	165	2.32109D-02	166	2.52382D-02	167	2.63463D-02
		168	1.32618D-02	168	2.12273D-02								
142	1.96667D-01	19	4.35581D-03	19	4.82043D-04	20	1.21824D-03	21	1.47356D-03	22	1.68310D-03	23	1.85452D-03
		24	1.98774D-03	25	1.03189D-03	25	1.37432D-03	32	9.86154D-03	38	3.24140D-03	45	1.28424D-02
		51	4.53653D-03	58	9.42214D-03	64	4.18130D-03	71	1.45525D-03	77	2.41211D-03	84	1.76030D-03
		90	2.71088D-03	97	2.14794D-03	103	2.99335D-03	110	2.61071D-03	116	3.24784D-03	123	1.26154D-02
		129	4.70178D-03	136	2.43564D-02	142	6.29878D-03	149	2.26392D-02	155	6.35860D-03	162	9.67000D-03
		162	1.38526D-03	163	3.50840D-03	164	4.44087D-03	165	5.28410D-03	166	5.76602D-03	167	5.43093D-03
		168	2.25561D-03	168	3.07036D-03								
149	8.00000D-01	19	1.61760D-02	19	1.74260D-03	20	4.44722D-03	21	5.46679D-03	22	6.35860D-03	23	7.14464D-03
		24	7.81988D-03	25	4.12729D-03	25	5.26333D-03	32	3.66569D-02	38	1.23169D-02	45	4.66498D-02
		51	1.74750D-02	58	3.25740D-02	64	1.67007D-02	71	4.74794D-03	77	1.00949D-02	84	5.42691D-03
		90	1.19480D-02	97	6.17725D-03	103	1.42090D-02	110	6.97867D-03	116	1.70177D-02	123	3.16492D-02
		129	2.82909D-02	136	6.17292D-02	142	4.52785D-02	149	6.78329D-02	155	5.25606D-02	162	3.38592D-02
		162	6.51985D-03	163	2.75520D-02	164	2.94492D-02	165	2.69784D-02	166	2.57011D-02	167	2.59114D-02
		168	1.33102D-02	168	2.58570D-02								
155	2.03333D-01	19	4.19062D-03	19	4.70673D-04	20	1.19278D-03	21	1.44727D-03	22	1.65916D-03	23	1.83661D-03
		24	1.98036D-03	25	1.03372D-03	25	1.33650D-03	32	9.33496D-03	38	3.10366D-03	45	1.17443D-02
		51	4.25908D-03	58	8.24763D-03	64	3.87003D-03	71	1.22989D-03	77	2.22349D-03	84	1.45124D-03
		90	2.50195D-03	97	1.73681D-03	103	2.77711D-03	110	2.10642D-03	116	3.04332D-03	123	1.06103D-02
		129	4.48707D-03	136	2.34647D-02	142	6.35860D-03	149	2.62803D-02	155	7.43970D-03	162	1.21888D-02
		162	1.25224D-03	163	3.22313D-03	164	4.37601D-03	165	5.96030D-03	166	8.05991D-03	167	9.45539D-03
		168	3.43319D-03	168	3.96602D-03								
162	8.00000D-01	19	1.50374D-02	19	1.62170D-03	20	4.18036D-03	21	5.16508D-03	22	6.02487D-03	23	6.78508D-03
		24	7.44422D-03	25	3.93719D-03	25	4.96780D-03	32	3.37992D-02	38	1.15361D-02	45	4.21967D-02
		51	1.61432D-02	58	2.89247D-02	64	1.51381D-02	71	4.17222D-03	77	8.99988D-03	84	4.74236D-03
		90	1.05000D-02	97	5.37745D-03	103	1.22748D-02	110	6.06742D-03	116	1.44402D-02	123	2.76039D-02
		129	2.36872D-02	136	5.48125D-02	142	3.86800D-02	149	6.77185D-02	155	4.87553D-02	162	6.24002D-02
		162	4.74300D-02	163	4.49637D-02	164	2.27465D-02	165	1.79189D-02	166	1.77019D-02	167	1.94014D-02
		168	1.07333D-02	168	2.59705D-02								

Table F16.A.2 (continued)

162	2.99999D-01	19	6.47002D-03	19	6.96418D-04	20	1.86889D-03	21	2.34844D-03	22	2.75685D-03	23	3.11325D-03
		24	3.42172D-03	25	1.81230D-03	25	2.23012D-03	32	1.42030D-02	38	5.07992D-03	45	1.67276D-02
		51	6.89029D-03	58	1.07234D-02	64	6.23127D-03	71	1.46933D-03	77	3.60051D-03	84	1.60400D-03
		90	4.09784D-03	97	1.73622D-03	103	4.63181D-03	110	1.85943D-03	116	5.19132D-03	123	7.92948D-03
		129	7.80498D-03	136	1.48475D-02	142	1.06388D-02	149	2.50362D-02	155	9.61723D-03	162	9.10656D-02
		162	2.01504D-03	163	3.15066D-03	164	3.02635D-03	165	3.31753D-03	166	3.68030D-03	167	3.91771D-03
		168	1.89569D-03	168	3.29234D-03								
163	3.00000D-01	19	7.45638D-03	19	8.38955D-04	20	2.19161D-03	21	2.71571D-03	22	3.16470D-03	23	3.55807D-03
		24	3.90005D-03	25	2.06303D-03	25	2.53754D-03	32	1.62224D-02	38	5.76340D-03	45	1.90764D-02
		51	7.75274D-03	58	1.22835D-02	64	6.96074D-03	71	1.69653D-03	77	4.00514D-03	84	1.87026D-03
		90	4.54648D-03	97	2.05716D-03	103	5.12865D-03	110	2.26291D-03	116	5.74656D-03	123	1.02319D-02
		129	8.67872D-03	136	2.27095D-02	142	1.20288D-02	149	4.72321D-02	155	1.10507D-02	162	3.85403D-02
		162	1.40654D-03	163	2.92671D-03	164	3.27796D-03	165	3.72698D-03	166	4.17496D-03	167	4.44929D-03
		168	2.13877D-03	168	3.62778D-03								
164	3.00000D-01	19	8.08122D-03	19	9.26548D-04	20	2.38714D-03	21	2.93732D-03	22	3.41137D-03	23	3.82876D-03
		24	4.19390D-03	25	2.21844D-03	25	2.70998D-03	32	1.75218D-02	38	6.12798D-03	45	2.06952D-02
		51	8.18364D-03	58	1.34903D-02	64	7.31021D-03	71	1.89005D-03	77	4.19472D-03	84	2.11422D-03
		90	4.75415D-03	97	2.37358D-03	103	5.35981D-03	110	2.68533D-03	116	6.02025D-03	123	1.26782D-02
		129	9.20482D-03	136	2.92283D-02	142	1.33226D-02	149	4.41737D-02	155	1.31280D-02	162	1.70599D-02
		162	1.18217D-03	163	2.86822D-03	164	3.46439D-03	165	4.05466D-03	166	4.62289D-03	167	4.99189D-03
		168	2.40870D-03	168	4.19552D-03								
165	3.00000D-01	19	8.37805D-03	19	9.67941D-04	20	2.47596D-03	21	3.03643D-03	22	3.52197D-03	23	3.95185D-03
		24	4.33081D-03	25	2.29258D-03	25	2.77048D-03	32	1.81304D-02	38	6.22847D-03	45	2.14962D-02
		51	8.25001D-03	58	1.41600D-02	64	7.32259D-03	71	2.00680D-03	77	4.18412D-03	84	2.26896D-03
		90	4.72715D-03	97	2.58097D-03	103	5.31454D-03	110	2.96087D-03	116	5.96772D-03	123	1.41014D-02
		129	9.21748D-03	136	3.09478D-02	142	1.40909D-02	149	3.59712D-02	155	1.58941D-02	162	1.19459D-02
		162	1.15192D-03	163	2.89876D-03	164	3.60415D-03	165	4.31343D-03	166	5.02935D-03	167	5.57094D-03
		168	2.74583D-03	168	5.19188D-03								
166	3.00000D-01	19	8.46044D-03	19	9.81424D-04	20	2.49971D-03	21	3.06038D-03	22	3.54883D-03	23	3.98416D-03
		24	4.37157D-03	25	2.31719D-03	25	2.76056D-03	32	1.82617D-02	38	6.15902D-03	45	2.16670D-02
		51	8.07166D-03	58	1.43542D-02	64	7.09714D-03	71	2.04797D-03	77	4.02688D-03	84	2.32936D-03
		90	4.52232D-03	97	2.66673D-03	103	5.04849D-03	110	3.07316D-03	116	5.62992D-03	123	1.45565D-02
		129	8.68011D-03	136	3.02859D-02	142	1.38385D-02	149	3.08413D-02	155	1.93438D-02	162	1.06211D-02
		162	1.15009D-03	163	2.92247D-03	164	3.69832D-03	165	4.52641D-03	166	5.44414D-03	167	6.34049D-03
		168	3.34406D-03	168	7.46694D-03								
167	3.00000D-01	19	8.37503D-03	19	9.74765D-04	20	2.47621D-03	21	3.02961D-03	22	3.51489D-03	23	3.95095D-03
		24	4.34369D-03	25	2.30685D-03	25	2.69777D-03	32	1.79987D-02	38	5.95819D-03	45	2.12664D-02
		51	7.69702D-03	58	1.40754D-02	64	6.67622D-03	71	2.00990D-03	77	3.74826D-03	84	2.28858D-03
		90	4.16966D-03	97	2.62261D-03	103	4.59635D-03	110	3.02078D-03	116	5.04087D-03	123	1.42123D-02
		129	7.58301D-03	136	2.87414D-02	142	1.18493D-02	149	2.82670D-02	155	2.06299D-02	162	1.05826D-02
		162	1.11299D-03	163	2.83136D-03	164	3.63046D-03	165	4.55804D-03	166	5.76408D-03	167	7.60746D-03
		168	4.95112D-03	168	1.48404D-02								

Table F16.A.2 (continued)

```
 168   3.00000D-01     19  8.22042D-03   19  9.59942D-04   20  2.43542D-03   21  2.97987D-03   22  3.46018D-03   23  3.89519D-03
                       24  4.29150D-03   25  2.28369D-03   25  2.62315D-03   32  1.75740D-02   38  5.73495D-03   45  2.06065D-02
                       51  7.30121D-03   58  1.35496D-02   64  6.24385D-03   71  1.92660D-03   77  3.46766D-03   84  2.18702D-03
                       90  3.82011D-03   97  2.49792D-03  103  4.15451D-03  110  2.86699D-03  116  4.46714D-03  123  1.34323D-02
                      129  6.47216D-03  136  2.70878D-02  142  9.21439D-03  149  2.71868D-02  155  1.40250D-02  162  1.09617D-02
                      162  1.00835D-03  163  2.54832D-03  164  3.27993D-03  165  4.20638D-03  166  5.69202D-03  167  9.27018D-03
                      168  8.02297D-03  168  3.00443D-02
 168   2.10000D-01     19  4.25056D-03   19  4.83097D-04   20  1.22435D-03   21  1.48791D-03   22  1.71022D-03   23  1.90004D-03
                       24  2.05883D-03   25  1.07957D-03   25  1.35676D-03   32  9.35631D-03   38  3.10489D-03   45  1.15207D-02
                       51  4.18094D-03   58  7.90147D-03   64  3.74173D-03   71  1.15600D-03   77  2.13330D-03   84  1.34269D-03
                       90  2.39113D-03   97  1.57847D-03  103  2.64700D-03  110  1.88141D-03  116  2.89765D-03  123  9.37321D-03
                      129  4.27980D-03  136  2.12273D-02  142  6.14073D-03  149  2.58570D-02  155  7.93204D-03  162  1.29852D-02
                      162  8.57380D-04  163  2.11620D-03  164  2.79701D-03  165  3.89391D-03  166  6.22245D-03  167  1.36037D-02
                      168  1.47092D-02  168  6.61985D-03

                                          ********** summary of heating6 input data***********
connectors
    19   38    1    1
     19  7.292871D-01   19  2.331993D-01   20  3.341410D-01
@    21  2.291357D-01   22  1.858808D-01   23  1.766134D-01
@    24  1.877988D-01   25  1.030181D-01   25  2.731725D-01
@    32  9.890657D-01   38  4.844228D-01   45  7.274280D-01
@    51  3.016397D-01   58  2.886303D-01   64  1.441124D-01
@    71  2.998723D-02   77  6.060741D-02   84  2.783664D-02
@    90  5.721976D-02   97  2.598190D-02  103  5.397485D-02
@   110  2.440855D-02  116  5.076790D-02  123  9.148128D-02
@   129  6.323963D-02  136  1.462755D-01  142  7.298222D-02
@   149  1.355160D-01  155  7.021454D-02  162  6.298846D-02
@   162  1.411540D-02  163  3.643874D-02  164  4.513402D-02
@   165  5.264083D-02  166  5.906502D-02  167  6.431559D-02
@   168  3.371623D-02  168  3.560943D-02
    19   37    1    1
     19  3.397760D-03   20  6.582134D-03   21  6.723272D-03
@    22  7.287625D-03   23  8.018761D-03   24  8.654208D-03
@    25  4.474753D-03   25  1.041875D-02   32  6.765113D-02
@    38  3.178962D-02   45  5.582092D-02   51  2.773573D-02
@    58  2.537872D-02   64  1.480794D-02   71  2.797271D-03
@ @  77  6.386529D-03   84  2.674247D-03   90  6.089159D-03
@ @  97  2.559735D-03  103  5.792466D-03  110  2.457689D-03
@  116  5.492750D-03  123  9.416446D-03  129  6.908654D-03
@  136  1.542948D-02  142  8.076702D-03  149  1.459874D-02
@  155  7.886201D-03  162  6.792971D-03  162  1.519348D-03
@  163  4.099907D-03  164  5.174822D-03  165  6.081755D-03
```

Table F16.A.2 (continued)

```
@  166  6.851631D-03  167  7.485661D-03  168  3.937225D-03
@  168  4.047182D-03
    20   36    1    1
     20  1.405890D-02   21  1.561329D-02   22  1.761708D-02
@    23  1.973465D-02   24  2.147484D-02   25  1.114057D-02
@    25  2.605245D-02   32  2.397782D-01   38  7.965817D-02
@    45  1.488710D-01   51  6.794863D-02   58  6.485464D-02
@    64  3.655262D-02   71  7.099785D-03   77  1.587249D-02
@    84  6.777462D-03   90  1.519095D-02   97  6.484782D-03
@   103  1.449915D-02  110  6.228367D-03  116  1.378969D-02
@   123  2.388661D-02  129  1.739811D-02  136  3.921611D-02
@   142  2.041183D-02  149  3.725697D-02  155  1.998528D-02
@   162  1.751067D-02  162  4.077296D-03  163  1.071023D-02
@   164  1.333232D-02  165  1.555692D-02  166  1.745127D-02
@   167  1.901590D-02  168  9.988916D-03  168  1.025711D-02
    21   35    1    1
     21  1.847638D-02   22  2.151565D-02   23  2.457779D-02
@    24  2.725537D-02   25  1.438547D-02   25  3.514373D-02
@    32  3.371137D-01   38  1.007132D-01   45  1.980981D-01
@    51  8.011169D-02   58  8.162195D-02   64  4.320634D-02
@    71  8.810294D-03   77  1.889062D-02   84  8.369725D-03
@    90  1.815031D-02   97  7.983822D-03  103  1.738192D-02
@   110  7.654164D-03  116  1.657839D-02  123  2.932424D-02
@   129  2.097591D-02  136  4.813535D-02  142  2.468973D-02
@   149  4.579845D-02  155  2.424918D-02  162  2.163545D-02
@   162  5.123508D-03  163  1.327146D-02  164  1.640510D-02
@   165  1.907847D-02  166  2.136546D-02  167  2.326577D-02
@   168  1.222199D-02  168  1.246506D-02
    22   34    1    1
     22  2.560335D-02   23  2.992665D-02   24  3.433918D-02
@    25  1.878927D-02   25  4.975500D-02   32  3.857261D-01
@    38  1.214806D-01   45  2.438920D-01   51  8.833614D-02
@    58  9.694153D-02   64  4.806180D-02   71  1.033919D-02
@    77  2.119505D-02   84  9.779971D-03   90  2.045657D-02
@    97  9.303300D-03  103  1.966395D-02  110  8.904358D-03
@   116  1.881288D-02  123  3.408155D-02  129  2.387517D-02
@   136  5.593424D-02  142  2.820067D-02  149  5.326971D-02
@   155  2.779944D-02  162  2.523693D-02  162  6.014519D-03
@   163  1.546564D-02  164  1.905271D-02  165  2.212918D-02
@   166  2.477554D-02  167  2.699244D-02  168  1.419201D-02
@   168  1.432753D-02
```

Table F16.A.2 (continued)

```
    23   33    1    1
     23  3.625631D-02   24  4.424854D-02   25  2.580849D-02
@    25  7.656788D-02   32  4.134954D-01   38  1.365637D-01
@    45  2.815148D-01   51  9.214864D-02   58  1.099079D-01
@    64  5.124442D-02   71  1.163763D-02   77  2.285567D-02
@    84  1.098203D-02   90  2.218101D-02   97  1.043318D-02
@   103  2.141731D-02  110  9.980512D-03  116  2.056606D-02
@   123  3.820345D-02  129  2.619522D-02  136  6.275011D-02
@   142  3.107283D-02  149  5.985481D-02  155  3.077266D-02
@   162  2.842127D-02  162  6.792056D-03  163  1.738799D-02
@   164  2.138382D-02  165  2.483021D-02  166  2.781470D-02
@   167  3.034110D-02  168  1.597620D-02  168  1.591776D-02
    24   32    1    1
     24  5.988464D-02   25  3.839950D-02   25  1.295880D-01
@    32  4.353347D-01   38  1.319911D-01   45  3.064436D-01
@    51  9.116679D-02   58  1.191779D-01   64  5.272390D-02
@    71  1.261014D-02   77  2.385447D-02   84  1.190628D-02
@    90  2.330485D-02   97  1.132260D-02  103  2.262478D-02
@   110  1.084549D-02  116  2.182417D-02  123  4.158818D-02
@   129  2.792488D-02  136  6.848260D-02  142  3.330484D-02
@   149  6.551167D-02  155  3.318126D-02  162  3.118228D-02
@   162  7.465036D-03  163  1.905923D-02  164  2.342314D-02
@   165  2.721129D-02  166  3.051935D-02  167  3.335711D-02
@   168  1.760168D-02  168  1.724799D-02
    25   31    1    1
     25  2.649982D-02   25  9.686361D-02   32  2.254552D-01
@    38  5.381991D-02   45  1.577354D-01   51  4.383465D-02
@    58  6.162519D-02   64  2.640151D-02   71  6.547618D-03
@    77  1.209582D-02   84  6.198083D-03   90  1.188507D-02
@    97  5.908772D-03  103  1.159272D-02  110  5.673069D-03
@   116  1.122732D-02  123  2.181059D-02  129  1.442497D-02
@   136  3.602963D-02  142  1.728945D-02  149  3.457669D-02
@   155  1.732009D-02  162  1.649207D-02  162  3.953829D-03
@   163  1.008186D-02  164  1.239011D-02  165  1.440473D-02
@   166  1.617707D-02  167  1.771530D-02  168  9.366616D-03
@   168  9.044160D-03
    25   30    1    1
     25  6.128076D-02   32  4.535144D-01   38  8.823786D-02
@    45  2.572586D-01   51  7.021248D-02   58  9.151577D-02
@    64  3.979814D-02   71  9.388185D-03   77  1.766892D-02
@    84  8.724527D-03   90  1.705791D-02   97  8.175628D-03
@   103  1.636678D-02  110  7.721502D-03  116  1.560268D-02
@   123  2.913693D-02  129  1.968462D-02  136  4.701600D-02
@   142  2.302693D-02  149  4.409399D-02  155  2.239319D-02
```

Table F16.A.2 (continued)

```
@  162  2.080907D-02  162  4.865363D-03  163  1.240078D-02
@  164  1.513536D-02  165  1.740746D-02  166  1.927233D-02
@  167  2.071738D-02  168  1.075891D-02  168  1.136637D-02
    32   29    1    1
     32  2.071651D+00   38  1.095452D+00   45  1.740726D+00
@    51  8.534634D-01   58  7.159644D-01   64  3.761753D-01
@    71  7.394494D-02   77  1.498530D-01   84  6.798009D-02
@    90  1.381394D-01   97  6.271065D-02  103  1.279194D-01
@   110  5.817651D-02  116  1.185417D-01  123  2.148555D-01
@   129  1.455322D-01  136  3.373772D-01  142  1.652317D-01
@   149  3.070965D-01  155  1.564087D-01  162  1.415780D-01
@   162  3.098615D-02  163  7.927775D-02  164  9.786005D-02
@   165  1.139166D-01  166  1.274910D-01  167  1.382198D-01
@   168  7.208035D-02  168  7.838328D-02
    38   28    1    1
     38  1.903796D-01   45  7.778710D-01   51  1.698998D-01
@    58  2.559109D-01   64  9.757750D-02   71  2.479757D-02
@    77  4.324061D-02   84  2.243188D-02   90  4.158548D-02
@    97  2.056605D-02  103  3.968908D-02  110  1.907625D-02
@   116  3.759122D-02  123  7.072591D-02  129  4.701387D-02
@   136  1.119438D-01  142  5.431014D-02  149  1.031862D-01
@   155  5.200237D-02  162  4.832251D-02  162  1.108267D-02
@   163  2.816530D-02  164  3.422507D-02  165  3.913462D-02
@   166  4.299804D-02  167  4.575565D-02  168  2.352202D-02
@   168  2.601143D-02
    45   27    1    1
     45  1.882528D+00   51  1.127563D+00   58  9.147707D-01
@    64  6.358326D-01   71  9.951744D-02   77  2.421768D-01
@    84  9.271966D-02   90  2.122445D-01   97  8.572787D-02
@   103  1.873990D-01  110  7.905489D-02  116  1.667753D-01
@   123  2.879345D-01  129  1.970940D-01  136  4.416308D-01
@   142  2.151758D-01  149  3.908125D-01  155  1.967778D-01
@   162  1.767531D-01  162  3.649402D-02  163  9.322505D-02
@   164  1.155839D-01  165  1.350644D-01  166  1.512644D-01
@   167  1.633139D-01  168  8.451808D-02  168  9.651543D-02
    51   26    1    1
     51  2.068874D-01   58  4.820071D-01   64  1.352626D-01
@    71  4.514394D-02   77  6.157099D-02   84  3.894538D-02
@    90  5.967357D-02   97  3.405511D-02  103  5.704755D-02
@   110  3.029800D-02  116  5.385540D-02  123  1.079454D-01
@   129  6.679255D-02  136  1.641200D-01  142  7.601030D-02
@   149  1.463979D-01  155  7.136152D-02  162  6.762065D-02
@   162  1.503228D-02  163  3.788702D-02  164  4.570605D-02
```

Table F16.A.2 (continued)

```
@  165  5.183633D-02  166  5.635082D-02  167  5.910891D-02
@  168  2.994608D-02  168  3.502619D-02
   58   25    1    1
    58  5.346641D-01   64  4.386615D-01   71  6.404962D-02
@   77  1.998577D-01   84  6.242035D-02   90  1.798829D-01
@   97  5.957724D-02  103  1.565555D-01  110  5.599814D-02
@  116  1.346772D-01  123  2.056537D-01  129  1.522131D-01
@  136  3.139089D-01  142  1.578694D-01  149  2.728915D-01
@  155  1.381904D-01  162  1.211593D-01  162  2.339486D-02
@  163  6.002846D-02  164  7.534423D-02  165  8.896997D-02
@  166  1.002112D-01  167  1.080912D-01  168  5.557392D-02
@  168  6.619522D-02
   64   24    1    1
    64  1.127922D-01   71  4.975097D-02   77  5.521230D-02
@   84  4.444608D-02   90  5.453639D-02   97  3.853432D-02
@  103  5.276898D-02  110  3.323025D-02  116  5.008604D-02
@  123  1.132964D-01  129  6.206965D-02  136  1.636552D-01
@  142  7.005843D-02  149  1.399118D-01  155  6.484294D-02
@  162  6.341021D-02  162  1.359452D-02  163  3.401661D-02
@  164  4.082794D-02  165  4.600921D-02  166  4.954736D-02
@  167  5.126968D-02  168  2.560930D-02  168  3.134661D-02
   71   23    1    1
    71  8.204047D-03   77  2.750122D-02   84  8.318352D-03
@   90  2.722195D-02   97  8.212966D-03  103  2.496236D-02
@  110  7.925461D-03  116  2.178057D-02  123  2.969849D-02
@  129  2.429701D-02  136  4.585187D-02  142  2.438295D-02
@  149  3.977626D-02  155  2.060708D-02  162  1.747655D-02
@  162  3.205588D-03  163  8.290797D-03  164  1.055605D-02
@  165  1.260906D-02  166  1.429751D-02  167  1.543491D-02
@  168  7.901983D-03  168  9.684449D-03
   77   22    1    1
    77  3.010094D-02   84  2.709222D-02   90  3.056745D-02
@   97  2.482106D-02  103  2.995172D-02  110  2.175012D-02
@  116  2.868892D-02  123  7.323215D-02  129  3.575952D-02
@  136  1.023582D-01  142  4.041528D-02  149  8.457056D-02
@  155  3.725497D-02  162  3.769859D-02  162  7.855094D-03
@  163  1.957280D-02  164  2.342773D-02  165  2.628959D-02
@  166  2.811293D-02  167  2.878453D-02  168  1.422268D-02
@  168  1.787189D-02
```

Table F16.A.2 (continued)

```
    84   21    1    1
     84  8.669642D-03   90  2.914559D-02   97  8.788530D-03
@   103  2.870106D-02  110  8.679924D-03  116  2.618456D-02
@   123  3.321477D-02  129  2.970716D-02  136  5.213318D-02
@   142  2.949411D-02  149  4.546433D-02  155  2.431578D-02
@   162  1.986474D-02  162  3.499380D-03  163  9.139816D-03
@   164  1.180802D-02  165  1.425632D-02  166  1.626197D-02
@   167  1.757501D-02  168  8.970101D-03  168  1.124852D-02
    90   20    1    1
     90  3.290328D-02   97  2.869107D-02  103  3.294351D-02
@   110  2.629127D-02  116  3.182478D-02  123  8.995531D-02
@   129  3.996746D-02  136  1.242174D-01  142  4.542115D-02
@   149  1.000956D-01  155  4.192054D-02  162  4.398213D-02
@   162  8.940096D-03  163  2.221828D-02  164  2.655216D-02
@   165  2.970155D-02  166  3.157173D-02  167  3.202067D-02
@   168  1.566825D-02  168  2.003186D-02
    97   19    1    1
     97  9.156483D-03  103  3.068129D-02  110  9.283716D-03
@   116  3.005922D-02  123  3.647121D-02  129  3.579906D-02
@   136  5.863606D-02  142  3.598915D-02  149  5.175045D-02
@   155  2.910056D-02  162  2.252503D-02  162  3.787851D-03
@   163  1.005320D-02  164  1.325656D-02  165  1.621672D-02
@   166  1.861731D-02  167  2.014023D-02  168  1.024529D-02
@   168  1.322373D-02
   103   18    1    1
    103  3.490752D-02  110  3.018021D-02  116  3.445451D-02
@   123  1.082173D-01  129  4.370911D-02  136  1.508693D-01
@   142  5.015398D-02  149  1.190374D-01  155  4.653099D-02
@   162  5.141640D-02  162  1.010505D-02  163  2.506331D-02
@   164  2.993484D-02  165  3.339226D-02  166  3.524512D-02
@   167  3.529745D-02  168  1.703984D-02  168  2.217543D-02
   110   17    1    1
    110  9.673415D-03  116  3.210428D-02  123  3.916365D-02
@   129  4.141799D-02  136  6.495432D-02  142  4.374279D-02
@   149  5.846439D-02  155  3.529343D-02  162  2.541513D-02
@   162  4.056642D-03  163  1.105867D-02  164  1.499770D-02
@   165  1.860369D-02  166  2.145470D-02  167  2.319795D-02
@   168  1.175904D-02  168  1.576163D-02
```

Table F16.A.2 (continued)

```
  116   16    1    1
   116  3.600482D-02  123  1.245145D-01  129  4.678183D-02
@  136  1.816844D-01  142  5.441804D-02  149  1.425673D-01
@  155  5.099127D-02  162  6.048693D-02  162  1.132569D-02
@  163  2.808299D-02  164  3.362342D-02  165  3.749628D-02
@  166  3.930424D-02  167  3.871107D-02  168  1.832208D-02
@  168  2.427527D-02
  123   15    1    1
   123  1.645047D-01  129  1.793211D-01  136  2.848752D-01
@  142  2.113734D-01  149  2.651439D-01  155  1.777780D-01
@  162  1.156270D-01  162  1.729945D-02  163  5.000241D-02
@  164  7.080845D-02  165  8.860184D-02  166  1.016236D-01
@  167  1.091423D-01  168  5.509297D-02  168  7.852486D-02
  129   14    1    1
   129  6.528069D-02  136  2.910852D-01  142  7.877903D-02
@  149  2.370096D-01  155  7.518166D-02  162  9.922065D-02
@  162  1.702782D-02  163  4.241225D-02  164  5.140942D-02
@  165  5.791514D-02  166  6.059861D-02  167  5.823337D-02
@  168  2.654571D-02  168  3.585437D-02
  136   13    1    1
   136  5.233514D-01  142  4.080957D-01  149  5.171416D-01
@  155  3.931546D-01  162  2.295979D-01  162  3.239225D-02
@  163  1.109796D-01  164  1.632419D-01  165  1.944509D-01
@  166  2.114351D-01  167  2.207179D-01  168  1.111014D-01
@  168  1.778333D-01
  142   12    1    1
   142  1.055371D-01  149  3.793242D-01  155  1.065393D-01
@  162  1.620225D-01  162  2.321022D-02  163  5.878382D-02
@  164  7.440743D-02  165  8.853594D-02  166  9.661063D-02
@  167  9.099610D-02  168  3.779304D-02  168  5.144445D-02
  149   11    1    1
   149  5.682755D-01  155  4.403305D-01  162  2.836586D-01
@  162  5.462061D-02  163  2.308194D-01  164  2.467127D-01
@  165  2.260134D-01  166  2.153132D-01  167  2.170747D-01
@  168  1.115071D-01  168  2.166188D-01
  155   10    1    1
   155  1.246534D-01  162  2.042258D-01  162  2.098153D-02
@  163  5.400412D-02  164  7.332073D-02  165  9.986582D-02
@  166  1.350450D-01  167  1.584266D-01  168  5.752371D-02
@  168  6.645129D-02
  162    9    1    1
   162  2.613815D-01  162  1.986744D-01  163  1.883434D-01
@  164  9.528020D-02  165  7.505848D-02  166  7.414936D-02
@  167  8.126859D-02  168  4.495952D-02  168  1.087849D-01
```

Table F16.A.2 (continued)

```
  162    8    1    1
   162  4.396136D-03  163  6.873671D-03  164  6.602473D-03
@  165  7.237721D-03  166  8.029173D-03  167  8.547120D-03
@  168  4.135764D-03  168  7.182774D-03
  163    7    1    1
   163  1.430260D-02  164  1.601915D-02  165  1.821347D-02
@  166  2.040270D-02  167  2.174331D-02  168  1.045201D-02
@  168  1.772867D-02
  164    6    1    1
   164  1.934882D-02  165  2.264552D-02  166  2.581911D-02
@  167  2.787995D-02  168  1.345270D-02  168  2.343221D-02
  165    5    1    1
   165  2.710209D-02  166  3.160033D-02  167  3.500326D-02

@  168  1.725256D-02  168  3.262152D-02
  166    4    1    1
   166  3.800727D-02  167  4.426495D-02  168  2.334595D-02
@  168  5.212906D-02
  167    3    1    1
   167  5.842108D-02  168  3.802185D-02  168  1.139661D-01
  168    2    1    1
   168  3.290641D-02  168  1.232275D-01
  168    1    1    1
   168  5.545834D-02

***** job description ***** ocular sample problem
***** the calculations for this case have been completed.
***** number of connectors generated for this case was          -- 1700
***** number of warnings issued for this case was               --    0
***** number of errors issued for this case was                 --    0
```

NUREG/CR-0200
Revision 6
Volume 2, Section F17
ORNL/NUREG/CSD-2/V2/R6

Computational Physics and Engineering Division

KENO-VI: A GENERAL QUADRATIC VERSION OF THE KENO PROGRAM

D. F. Hollenbach
L. M. Petrie
N. F. Landers*

Date Published: March 2000

Prepared for the
U.S. Department of Energy
Office of Environmental, Safety, and Health
Budget Activity No. HP 01 00 000
and
U.S. Nuclear Regulatory Commission
Office of Nuclear Material Safety and Safeguards
Washington, DC 20555
Under Interagency Agreement DOE 1886-C000-9B
NRC FIN No. B-0009

Prepared by the
OAK RIDGE NATIONAL LABORATORY
managed by
LOCKHEED MARTIN ENERGY RESEARCH CORP.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-96OR22464

---

*Formerly with Oak Ridge National Laboratory.

ABSTRACT

KENO-VI is an extension of the KENO Monte Carlo criticality program. It was developed for use in the SCALE system. KENO-VI contains all the features currently in KENO V.a plus a more flexible geometry package. The geometry package in KENO-VI is capable of modeling any volume that can be constructed using quadratic equations. In addition, such features as geometry intersections, body rotation, hexagonal arrays, and array boundaries have been included to make the code more flexible.

A set of predefined geometry volumes have been incorporated into KENO-VI. Additional volumes can be constructed using the QUADRATIC geometry card that allows the user to specify any volume that can be modeled using quadratic equations. The ability to intersect the volumes makes it possible to exactly model such things as pipe intersections, which was previously impossible in KENO V.a The ability to rotate bodies means volumes no longer must be positioned parallel to a major axis. Hexagonal arrays were added to simplify the construction of arrays. The use of array boundaries makes it possible to fill a noncuboidal volume with an array, specifying the boundary where a particle leaves and enters the array.

The primary purpose of KENO-VI is to determine k-effective. Other calculated quantities include lifetime, generation time, energy-dependent leakages, energy- and region-dependent absorptions, fissions, fluxed, and fission densities.

KENO-VI retains the KENO V.a features such as flexible data input, the capability of supergrouping energy-dependent data, a P_n 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 restart capabilities.

This advanced user-oriented program features a more complex geometry package, simplified data input, and efficient use of computer storage. These features allow the user to readily solve large geometrically complex problems whose computer storage requirements and geometric complexity preclude solution by previous versions of KENO.

CONTENTS

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

ACKNOWLEDGMENTS

Many individuals have contributed significantly to the development of KENO-VI. Special recognition is given to G. E. Whitesides, formerly Director, Computing Applications Division, who was responsible for the concept and development of the original KENO code. He has also contributed significantly to some of the techniques utilized in KENO-VI. J. T. Thomas has offered 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 for reviewing this document. P. B. Fox provided the color plots. The efforts of L. F. Norris (retired) and W. C. Carter in preparing this document are gratefully acknowledged.

Special appreciation is expressed to the DOE Nuclear Criticality Methods Resource Center, who with the leadership of program managers Blake Brown, Bill Kinney, Herb Field, Bob Kratzke, Jerry Counts, Sam Rosenblum, and Burton Rothleder provided funding for the development of KENO-VI.

F17.1 INTRODUCTION

KENO-VI, a functional module in the SCALE system, is a multigroup Monte Carlo criticality program used to calculate the k_{eff} of a three-dimensional (3-D) system. KENO-VI is the latest version of the KENO Monte Carlo criticality safety code. KENO-VI contains all the capabilities of KENO V.a such as 2-D plotting, arrays, holes, and a very similar output. KENO-VI differs from KENO V.a in the following ways. It contains a much larger set of geometrical bodies, including cuboids, cylinders, spheres, cones, dodecahedrons, elliptical cylinders, ellipsoids, hoppers, parallelpipeds, planes, rhomboids, and wedges. The code's flexibility is increased by allowing the following features: intersecting geometry regions; hexagonal as well as cuboidal arrays; bodies and holes rotated to any angle and translated to any position; and a specified array boundary that contains only that portion of the array located inside the boundary. Users should be aware that the added geometry features in KENO-VI can result in significantly longer run times than KENO V.a. A KENO-VI problem that can be modeled in KENO V.a will typically run twice as long in KENO-VI as in KENO V.a. Thus the new version VI is not a replacement for the existing version V.a, but an additional version for more complex geometries that could not be modeled previously.

The KENO-VI 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 feature 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 one-dimensional (1-D) (reaction rate) cross-section IDs 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.

KENO-VI contains a more flexible geometry package than the one in KENO V.a. In KENO-VI, geometry regions are constructed and processed as sets of quadratic equations. A set of geometric shapes that include all those used in KENO V.a plus others and the ability to build more complex geometric shapes using sets of quadratic equations is available in KENO-VI. Unlike KENO V.a, KENO-VI allows intersections between geometry regions within a unit and the ability to specify an array boundary that intersects the array.

The primary difference between the KENO V.a and KENO-VI geometry input is the methodology used to represent the geometry/material regions in a unit. KENO-VI uses two geometry records to describe a region. The first record, called the GEOMETRY record, contains the geometry keyword, region boundary definitions, and any geometry modification data. The second record, called the MEDIA record, contains the media keyword, the material, hole, or array ID number, the bias ID number, and the region definition vector. Using geometry modification data, regions can be rotated and translated to any angle and position within a unit. KENO-VI requires that a global unit be specified in a problem, including single unit problems.

All the options in KENO V.a pertaining to arrays are present in KENO-VI, plus additional flexibility. In addition to cuboidal arrays, hexagonal arrays can be directly constructed in KENO-VI. Also, the ability to specify an array boundary that intersects the array makes it possible to construct a lattice in a cylinder using one array in KENO-VI instead of multiple arrays and holes that would be required in KENO-VI.

An important feature of KENO-VI 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-VI 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 semicontinuous 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-VI can extend the use of these albedos to include other energy group structures 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-VI 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.

F17.2 THEORY AND TECHNIQUES

F17.2.1 THE TRANSPORT EQUATION

The equation KENO-VI solves may be derived in the following manner, starting with the Boltzmann neutron transport equation which may be written as[1]

$$\frac{1}{v}\frac{\partial\Phi}{\partial t}(X,E,\Omega,t) + \Omega\cdot\nabla\Phi(X,E,\Omega,t) + \Sigma_t(X,E,\Omega,t)\Phi(X,E,\Omega,t) = S(X,E,\Omega,t)$$

$$+ \int_{E'}\int_{\Omega'}\Sigma_s(X,E'\rightarrow E,\Omega'\rightarrow\Omega,t)\Phi(X,E',\Omega',t)d\Omega'dE' \ , \qquad \text{(F17.2.1)}$$

where

$\Phi(X,E,\Omega,t)$ = neutron flux (neutrons/cm^2/s) 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,

$\Sigma_s(X,E'\rightarrow E,\Omega'\rightarrow\Omega,t)$ = macroscopic 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/s 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:

$$q(X,E,\Omega,t) = S(X,E,\Omega,t) + \int_{E'}\int_{\Omega'}\Sigma_s(X,E'\rightarrow E,\Omega'\rightarrow\Omega,t)\Phi(X,E',\Omega',t)d\Omega'dE' \ . \qquad \text{(F17.2.2)}$$

Combining Eqs. (F17.2.1) and (F17.2.2), assuming the media to be isotropic, ignoring the time dependence of the cross sections and converting the equation to multigroup form, yields

$$\frac{1}{v_g}\frac{\partial\Phi_g}{\partial t}(X,\Omega,t) + \Omega\cdot\nabla\Phi_g(X,\Omega,t) + \Sigma_{t_g}(X)\Phi_g(X,\Omega,t) = q_g(X,\Omega,t) \ , \qquad \text{(F17.2.3)}$$

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

$$\Sigma_{tg}(X) = \frac{\int_{\Delta E_g} \Sigma_t(X,E)\Phi(X,E,\Omega,t)dE}{\int_{\Delta E_g} \Phi(X,E,\Omega,t)dE} ,$$

where

ΔE_g defines group g,

$q_g(X,\Omega,t)$ 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. (F17.2.3), and defining

$$T(R) = \int_0^R \Sigma_{t_g}(X-R'\Omega)dR' ,$$

the following equation can be written:

$$\Phi_g(X,\Omega) = \int_0^\infty q_g(X-R\Omega,\Omega)e^{-T(R)}dR . \qquad \text{(F17.2.4)}$$

At this point, the problem becomes an eigenvalue problem. If there is no external source, the source may be defined as

$$q_g(X,\Omega) = \sum_{g'} \int d\Omega' \Phi_{g'}(X,\Omega')\Sigma_s(X,g'\rightarrow g,\Omega'\cdot\Omega) + \frac{1}{k}Q'_g(X,\Omega) , \qquad \text{(F17.2.5)}$$

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

$$\Sigma_s(X,g'\rightarrow g,\Omega'\cdot\Omega) = \frac{\int_{\Delta E_g}\int_{\Delta E_{g'}}\Sigma_s(X,E'\rightarrow E,\Omega'\cdot\Omega)\Phi(X,E',\Omega')dE'dE}{\int_{\Delta E_{g'}}\Phi(X,E',\Omega')dE'}, \qquad \text{(F17.2.6)}$$

where

ΔE_g is the energy range defining energy group g

$\Delta E_{g'}$ 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

$$Q'_g(X,\Omega) = \frac{1}{4\pi}\sum_{g'}\int_{\Omega'}d\Omega'\Phi_{g'}(X,\Omega')\chi(X,g'\rightarrow g)\upsilon_{g'}(X)\Sigma_{fg'}(X), \qquad \text{(F17.2.7)}$$

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. (F17.2.5) into Eq. (F17.2.4) yields the following equation:

$$\Phi_g(X,\Omega) = \int_0^\infty dR e^{-T(R)}$$

$$\left\{\frac{1}{k}Q'_g(X-R\Omega,\Omega) + \sum_{g'}\left[\int_{\Omega'}d\Omega'\Phi_{g'}(X-R\Omega,\Omega')\Sigma_s(X-R\Omega,g'\rightarrow g,\Omega'\cdot\Omega)\right]\right\}. \qquad \text{(F17.2.8)}$$

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. (F17.2.7), Eq. (F17.2.8) can be written as

$$\Phi_g(X,\Omega) = \int_0^\infty dR e^{-T(R)} \left\{ \sum_{g'} \frac{1}{k} \int_{\Omega'} \upsilon_{g'}(X-R\Omega)\Sigma_{fg'}(X-R\Omega)\chi(X-R\Omega,g'\rightarrow g)\Phi_{g'}(X-R\Omega,\Omega')\frac{d\Omega'}{4\pi} \right.$$

$$\left. + \sum_{g'} \int_{\Omega'} d\Omega' \Sigma_{tg'}(X-R\Omega,\Omega') \frac{\Sigma_s(X-R\Omega,g'\rightarrow g,\Omega'\cdot\Omega)}{\Sigma_{tg'}(X-R\Omega)} \right\} . \qquad \text{(F17.2.9)}$$

Writing Eq. (F17.2.9) 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-VI.

$$\frac{\upsilon_g(X)\Sigma_{fg}(X)}{\Sigma_{tg}(X)} \Sigma_{tg}(X)\Phi_{g,n}(X,\Omega) = \frac{\upsilon_g(X)\Sigma_{fg}(X)}{\Sigma_{tg}(X)} \Sigma_{tg}(X) \int_0^\infty dR e^{-T(R)}$$

$$\left\{ \frac{1}{k} \sum_{g'} \int_{\Omega'} \frac{\upsilon_{g'}(X-R\Omega)\Sigma_{fg'}(X-R\Omega)}{\Sigma_{tg'}(X-R\Omega)} \chi(X-R\Omega,g'\rightarrow g)\Sigma_{tg'}(X-R\Omega)\phi_{g',n-1}(X-R\Omega,\Omega')\frac{d\Omega'}{4\pi} \right.$$

$$\left. + \sum_{g'} \int_{\Omega'} \frac{\Sigma_s(X-R\Omega,g'\rightarrow g,\Omega'\cdot\Omega)}{\Sigma_{tg'}(X-R\Omega)} \Sigma_{tg'}(X-R\Omega)\Phi_{g',n}(X-R\Omega,\Omega')d\Omega' \right\} \qquad \text{(F17.2.10)}$$

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-VI solves Eq. (F17.2.10) 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-VI are chosen by selecting path lengths from the distribution

$$e^{-T(R)n},$$

which is the probability of transport from any position X – RΩ 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

$$\sum_{g'}\int_{\Omega'} \frac{\Sigma_s(X,g'\rightarrow g,\Omega'\cdot\Omega)}{\Sigma_{tg'}(X)} \Sigma_{t_{g'}}(X)\Phi_{g',n}(X,\Omega')d\Omega' .$$

The collision density in group g, per solid angle about Ω is

$$\Sigma_{tg}\int_0^{\infty} dR e^{-T(R)}\sum_{g'}\int_{\Omega'} \frac{\Sigma_s(X-R\Omega,g'\rightarrow g,\Omega'\cdot\Omega)}{\Sigma_{tg'}(X-R\Omega)} \Sigma_{tg'}(X-R\Omega)\Phi_{g',n}(X-R\Omega,\Omega')d\Omega' .$$

The total collision density times $\frac{\upsilon_g(X)\Sigma_{fg}(X)}{\Sigma_{tg}(X)}$ is the relationship from which KENO-VI picks the source points for the next generation.

F17.2.2 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; 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-VI 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-VI randomly selects the cosine of the scattering angle according to

(1) $|\bar{\mu}| < \frac{10^{-10}}{3}$: scattering distribution is isotropic,

(2) $|\bar{\mu}| \le 1/3$: $\mu = (\sqrt{1 + 6\zeta\bar{\mu} + (3\bar{\mu})^2} - 1)/3\bar{\mu}$,

or

(3) $|\bar{\mu}| > 1/3$: $\mu = \zeta(1 - |\bar{\mu}|) + \bar{\mu}$,

where ζ is a uniform random variable between −1 and +1, and $\bar{\mu}$ is the mean cosine of the scattering angle. Otherwise, KENO-VI 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.

$$u' = u\cos\psi - \sqrt{v^2+w^2}\sin\psi\cos\eta$$

$$v' = v\cos\psi + \frac{uv}{\sqrt{v^2+w^2}}\cos\eta\sin\psi - \frac{w}{\sqrt{v^2+w^2}}\sin\psi\sin\eta ,$$

$$w' = w\cos\psi + \frac{uw}{\sqrt{v^2+w^2}}\cos\eta\sin\psi + \frac{v}{\sqrt{v^2+w^2}}\sin\psi\sin\eta ,$$

where

$\sin\psi = \sqrt{1-\mu^2}$,

$\cos\psi = \mu$ = cosine of the scattering angle,

η = a random azimuthal angle between 0 and 2π.

F17.2.3 BIASING OR WEIGHTING

In order to minimize the statistical deviation of k_{eff} per unit computer time, KENO-VI 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 that 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 following values are used by KENO-VI:

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 F17.2.1, 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 F17.2.2, 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-VI 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. Even though 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-VI to allow the user to minimize the variance in k_{eff} 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, when 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-VI 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[6] have been prepared for several reflector materials commonly used in KENO-VI calculations. The use of biasing to minimize the variance in k_{eff} per unit computer time will usually increase the variance in other parameters such as leakage or absorption in the reflector.

F17.2.4 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_{eff} because of the relatively long time a history may spend in the reflector. A differential albedo technique[7] was developed for use with the KENO codes to eliminate tracking in the reflector. This technique 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_{eff}. 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-VI were generated by 1-D discrete-ordinates calculations for slab geometry, representing infinite slabs. Consequently, for

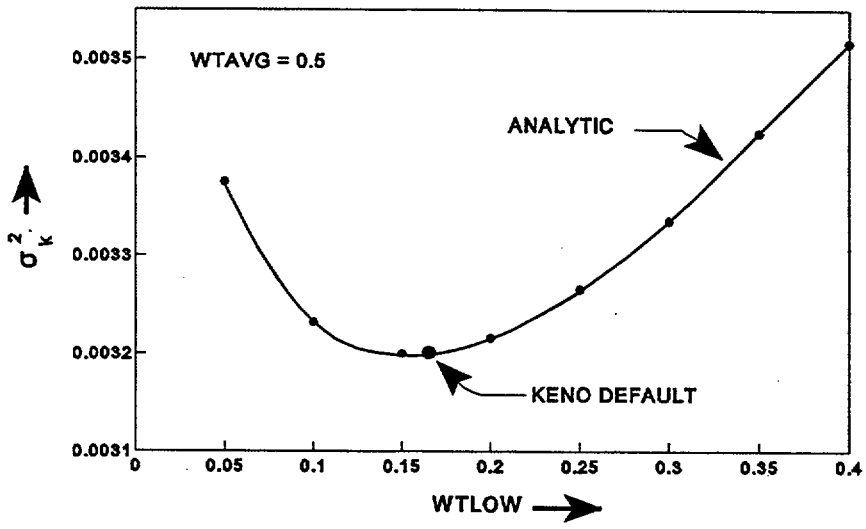


Figure F17.2.1 Analytic estimate of the relationship between WTLOW and the variance, σ^2_κ, when WTAVG is 0.5

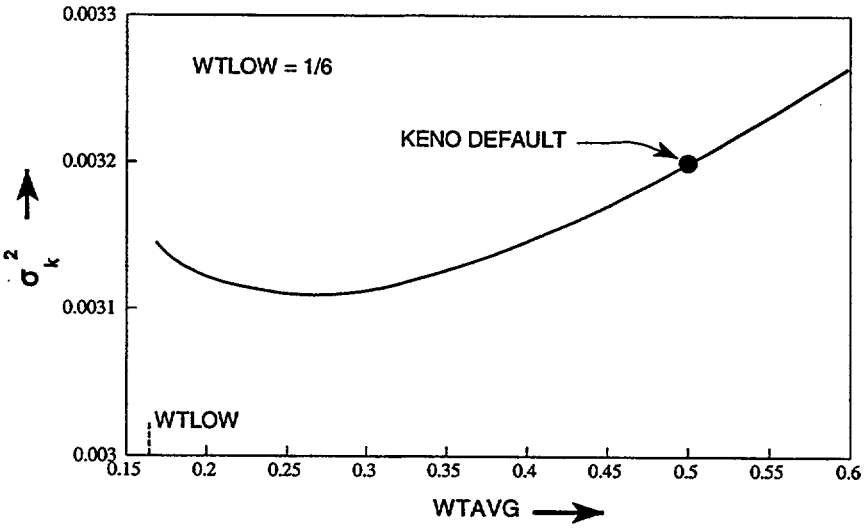


Figure F17.2.2 Analytic estimate of the relationship between WTAVG and the variance, σ^2_κ, when WTLOW is 1/6

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_{eff} 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-VI. In the past, their use was limited to problems utilizing cross sections having the Hansen-Roach 16-energy-group structure. KENO-VI extends the use of differential albedos to other energy group structures by allowing appropriate energy transfers. Lethargy boundary tables are created 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), as 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.

F17.2.5 SUPERGROUPING

An important feature of KENO-VI is the capability of supergrouping energy-dependent information—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-VI 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/Os. The more supergroups, the more I/Os 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-VI 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 are then brought into memory and tracking proceeds.

F17.2.6 RESTART

KENO-VI 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 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_{eff} values 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-VI 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 1, 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.

F17.2.7 GEOMETRY

KENO-VI geometry can model any geometric shape that can be described using quadratic equations. These geometric shapes are stacked together forming regions. The set of regions is then used to build units. A set of predefined shapes that include cones, cuboids (rectangular parallelepipeds), cylinders, dodecahedrons, ecylinders (elliptical cylinders), ellipsoids, hexprisms, hoppers, parallelepipeds, planes, rhombohedrons, spheres, and wedges is used to construct regions. In addition, the keyword quadratic is provided which allows additional shapes to be constructed by specifying the quadratic equations that describe the shape. These shapes can be rotated and translated to any orientation and position within their respective unit. Hemispheres and hemicylinders can be constructed using spheres and cylinders with a chord. Regions are rotated by providing the nonzero angles associated with the Euler X-convention.

A major improvement in KENO-VI is the ability to intersect regions. Region volumes are no longer calculated due to the complexity involved with intersecting regions. Each set of multiple geometry regions is called a unit. KENO-VI allows multiple sets of geometry regions (i.e., units), with each set having an independent coordinate system. A global unit must be specified for every problem, including single-unit problems. Units having cuboidal outer boundaries where the adjoining faces have the same dimensions can be stacked together in a 3-D rectangular parallelepiped called an array or lattice, just as children's blocks can be stacked. Unlike KENO V.a, units having hexagonal outer boundaries where the adjoining faces have the same dimensions can also be stacked together in an array. An array boundary must be specified that either coexists with the outermost edge of the array or that is entirely within the array. The array boundary can be any shape that is definable using quadratic equations. An array can be treated as a building block and be used as a unit within another array.

The use of holes in KENO-VI allows a unit to be emplaced within another unit. This feature allows the addition of a complex structure, previously defined as a unit, to be directly placed within another unit. However, a hole is not allowed to intersect other holes and must be completely contained within a specified region. In KENO-VI, the use of holes to place complex structures within a unit may actually reduce the CPU time needed for the problem.

Multiple arrays can be described in KENO-VI. A global array must be specified if one is desired. In KENO-VI the outermost boundary is always specified as the global-unit boundary. If the outermost boundary is to be the array boundary, a global unit must still be specified with the global-unit boundary coinciding with the array boundary. Unlike past versions of KENO, KENO-VI cannot run a single unit problem without specifying a global unit.

F17.2.8 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, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 1970.

4. T. J. Hoffman, *The Optimization of Russian Roulette Parameters for KENO V.a*, ORNL/TM-7539, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 1982.

5. L. M. Petrie and N. F. Cross, *KENO-IV - An Improved Monte Carlo Criticality Program*, ORNL-4938, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 1975. Also see "KENO IV/S: An Improved Monte Carlo Criticality Program," Sect. F5 of *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation*, NUREG/CR-0200, Rev. 3.

6. J. R. Knight and L. M. Petrie, *16- and 123-Group Weighting Functions for KENO V.a*, ORNL/TM-4660, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 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).

F17.3 LOGICAL PROGRAM FLOW

The general flow of the KENO-VI program during the solution of a problem is given in this section. A formal flowchart 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 flowchart, a brief explanation of the purpose of that section of the program, and a brief description of each subroutine involved. The abbreviated flowcharts are drawn with KENO-VI subroutine names contained in boxes and library routines as bare names. An arrow in the flowchart indicates that the subroutine associated with the arrow will be treated in detail later in the section.

F17.3.1 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 (Figure F17.3.1).

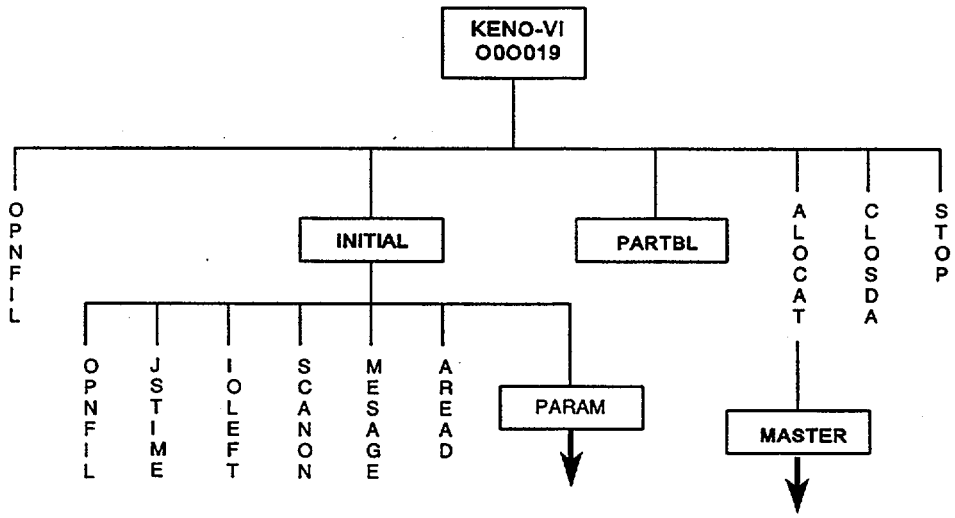


Figure F17.3.1 Flowchart of program initiation

KENOVI - O0O019 - These main programs set flags to specify the proper mode of data reading for the stand-alone KENO-VI 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.

OPNFIL - This library routine is called to initialize the input and output logical units.

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.

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/Os 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-VI), the date, the time execution was begun, and the job name. The library routine LISTQA is called to provide a program verification page for quality assurance purposes.

AREAD - This library routine is used to read alphanumeric data. It is used here to read the title 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. F17.3.2 for a more detailed description of PARAM.

PARTBL - This subroutine prints out a table containing the values of all input parameters.

ALOCAT - This subroutine is called with three 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. The third argument is prefixed by an asterisk and specifies the statement number to return to if there is not enough storage to run the problem.

MASTER - This subroutine is called by subroutine ALOCAT. It is the controlling subroutine for the bulk of the KENO-VI program flow. See Sect. F17.3.3 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.

STOP - This library routine is called to write a message and terminate execution when there is insufficient storage to run the problem.

F17.3.2 PARAMETER DATA

The function of this section of the program is to set default values for the parameters and to read the parameter input data (Figure F17.3.2).

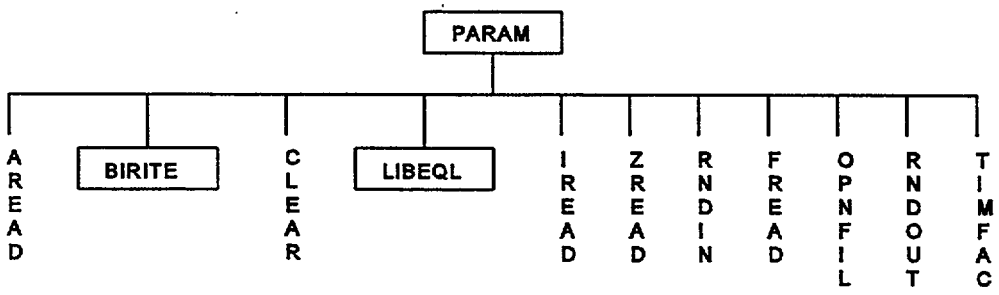


Figure F17.3.2 Flowchart of parameter reading

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.

OPNFIL - This library routine is called to initialize the logical unit for the restart data file.

BIRITE - This subroutine writes information to a buffer in CSAS format which could later be used for a restart problem.

LIBEQL - This subroutine reads the library number from ICE type input which contains the cross-section data to be used by the problem.

F17.3.3 OVERALL PROGRAM FLOW

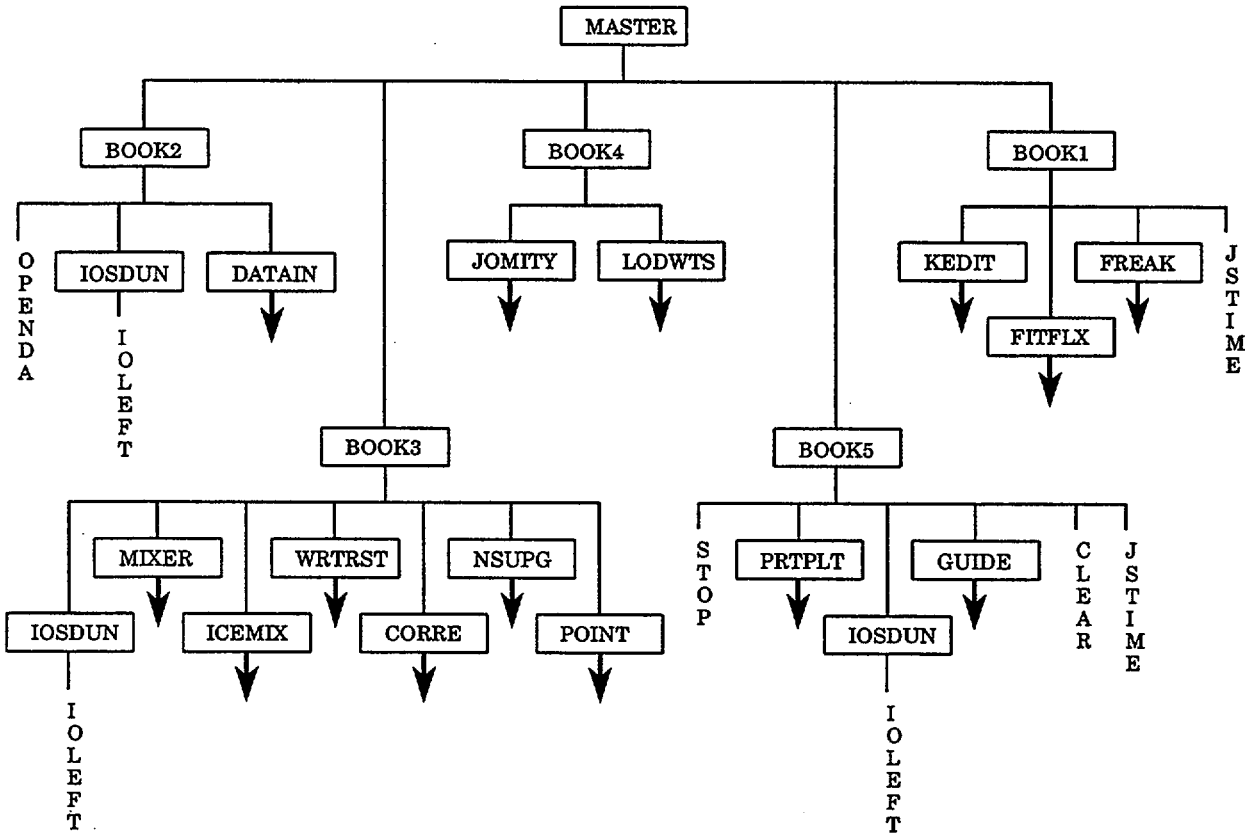


Figure F17.3.3 Flowchart of overall program flow

The purpose of this section of the program is to direct the primary flow of KENO-VI (Figure F17.3.3). This flowchart 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-VI. 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/Os used during various operations is calculated and printed. The following subroutines are called from MASTER as indicated:

Subroutine	Function	Condition
OPENDA	Initializes direct access	Always
BOOK2	Controls Inputting of Data	Always
BOOK3	Controls Data Mixing and Organization	Always
BOOK4	Controls Geometry Processing and biasing data	Always
BOOK5	Controls Tracking and Plots	Always
BOOK1	Controls Output	Always
IOSDUN	Initializes I/Os	Always
DATAIN	Reads input data	Always
MIXER	Mixes cross sections	If mixing table is read
ICEMIX	Reads ICE mixed library	If cross sections are not to be used from the restart data file
WRTRST	Writes restart information	If a unit is defined for writing restart information
CORRE	Generates albedo-cross-section information	If differential albedo data are used
NSUPG	Creates supergrouped data	Always
POINT	Calculates 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 are stored	Always
LODWTS	Loads biasing data from direct access into memory	Always
GUIDE	Controls tracking	Always
KEDIT	Edits calculated results	Always
FITFLX	Loads fluxes for printing	If fluxes are calculated
FREAK	Prints frequency distribution	Always
JSTIME	Timing	Always

OPENDA - This library routine initializes the direct-access files.

BOOK2 - This subroutine is called once to call the subroutines needed to read the input data.

BOOK3 - This subroutine is called once to call the subroutines that read, mix, or generate the required cross-section data, write restart information, create supergroup data if required, and calculate pointers.

BOOK4 - This subroutine is called once to call the subroutines that controls geometry data processing and reads required biasing data.

BOOK5 - This subroutine is called once to call the subroutines that print plots and control particle tracking.

BOOK1 - This subroutine is called once to call the subroutines that calculate results and print output.

IOSDUN - This subroutine is called several times to indicate the number of I/Os used for various operations.

DATAIN - This subroutine controls the reading of all input data except the title and parameters. It is explained in more detail in Sect. F17.3.4.

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. F17.3.5.

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-VI. This information is then written on the direct-access data file. See Sect. F17.3.6 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. F17.3.7.

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. F17.3.8 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. F17.3.9.

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 F17.3.10 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. F17.3.11.

PRTPLT - This subroutine is called to generate and print 2-D printer plots representing slices through the geometrical representation of the physical problem. See Sect. F17.3.12 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. F17.3.13 for a more detailed explanation.

KEDIT - This subroutine is responsible for editing the k_{eff} values and printing the various information calculated by KENO-VI. Section F17.3.14 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. F17.3.15.

FREAK - This subroutine is called to generate and print the frequency distribution of the k_{eff} 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.

F17.3.4 PROBLEM DESCRIPTION

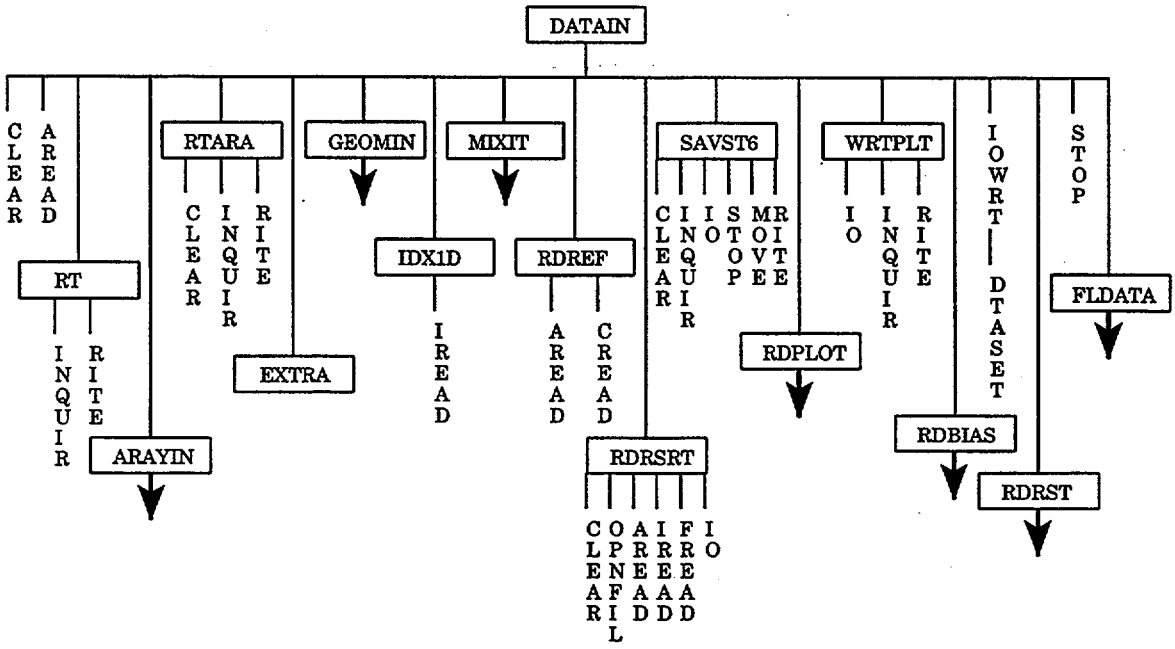


Figure F17.3.4 Flowchart of input data reading

This section of the program controls the reading of the input data (Figure F17.3.4), 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 and parameter data. It initializes COMMON /STDATA/, the common that contains the start data, and initializes the MT array that contains the IDs 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 Sect. F17.4. 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 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-VI, their names, data set names and the volume containing the data set. This table can be valuable for quality assurance applications. The library routine STOP is called to write error messages. 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. INQUIR is called to return the value of the next direct-access record after the geometry data and/or the extra 1-D cross sections are written on a direct-access device.

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 always called if array data are specified. See Sect. F17.3.4.1 for additional information. Data input for the array data block is described in detail in Sect. F17.4.5.

RTARA - This subroutine is called only if ARAYIN is called. CLEAR is called to zero the array before RTARA reads the array data from the scratch unit and writes them on the direct-access device. INQUIR is called to return the value of the next direct-access record after the array data are written on the direct-access device.

EXTRA - This dummy subroutine is provided to allow the user to input extra data that are not typically processed by KENO-VI. 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. F17.3.4.2 for additional information. The geometry region data block is described in detail in Sect. F17.4.4.

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 IDs and loads them into the MT array. The data reading is accomplished using the library routine IREAD. Section F17.4.9 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 F17.3.4.3 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. The library routine, CREAD, is used to read the albedo names. Some preliminary data checks are made to detect invalid face code names and incompatible boundary conditions. Section F17.4.6 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, CLEAR, is called to zero the array that will contain the start data. 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. The library routine, INQUIR, returns the value of the next direct-access record after writing the start data on the direct-access device. Data input for defining the initial source distribution is described in Sect. F17.4.8.

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. F17.4.8 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. The library routine, INQUIR, returns the value of the next direct-access record after writing the start type 6 data on the direct-access device.

RDPLOT - This subroutine is called to read the plot data block that is used to generate printer plots. Section F17.3.4.4 explains the processing of the plot data in more detail. Section F17.4.11 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 them in memory. RITE writes the data on the direct-access data file. INQUIR returns the value of the next direct-access record.

RDBIAS - This subroutine is called to read the biasing or weighting data to be used in the problem. See Sect. F17.3.4.5 for more information. The biasing input data block is described in detail in Sect. F17.4.7.

IOWRT - This library routine 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. F17.3.4.6.

FLDATA - This subroutine is called to supply default data for arrays that were not entered as input. Section F17.3.4.7 contains a more-detailed account of the exact procedure.

F17.3.4.1 *Array Data*

This section explains the procedure involved in reading the array data used in the problem (Figure F17.3.5).

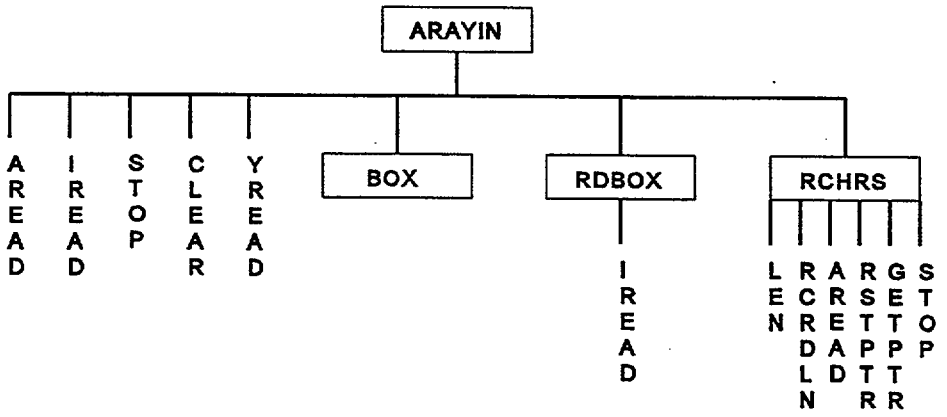


Figure F17.3.5 Flowchart for reading array data

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 unit orientation array. The unit orientation array data block for each array is read by YREAD for the FILL option and by RDBOX for the LOOP option. BOX then writes the array data on the scratch data file. CLEAR is called to zero the unit orientation array and the data reading is done using the library routines AREAD, IREAD, and YREAD. Section F17.4.5 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.

CLEAR - This library routine is called to zero the unit orientation array before it is loaded.

YREAD - This library routine is called to read the unit orientation data for the FILL option.

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. The intrinsic function LEN determines the length of the comment. 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).

F17.3.4.2 *Geometry Data*

This section of the program reads the geometry data (Figure F17.3.6).

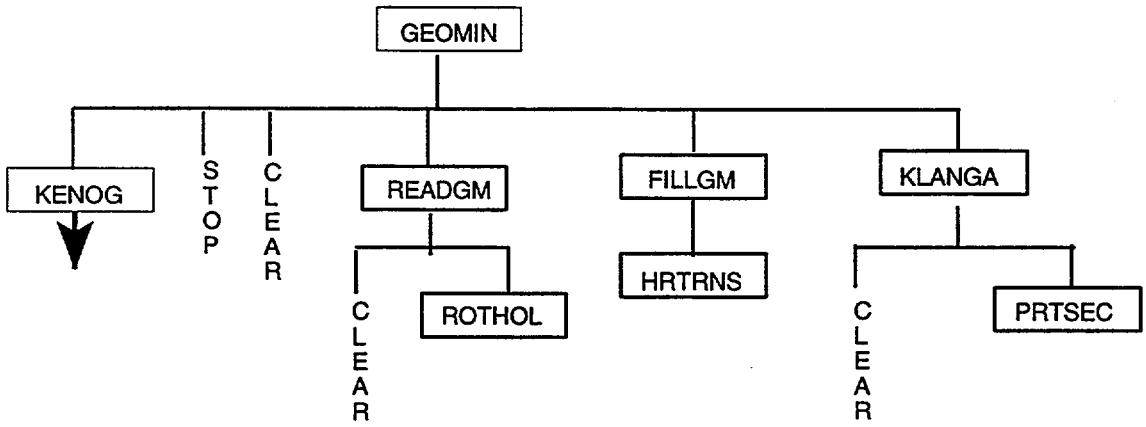


Figure F17.3.6 Flowchart for reading geometry and media keyword data

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. STOP is called to write an error message if more storage is needed. The data block read in this portion of the program is described in Sect. F17.4.4.

KENOG - This subroutine uses AREAD to read the geometry word and IREAD to read the unit number. KENOG then calls the appropriate subroutine, based on the geometry word previously read, to read in the associated geometry data. 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.

READGM - This subroutine reads the geometry data from the scratch data unit created in KENOG and loads them into the proper arrays.

KLANGA - This subroutine creates a geometry word sector array that defines how the geometry/material regions relate to the geometric shapes in each unit.

ROTHOL - This subroutine creates a rotation matrix for the holes if required.

PRTSEC - This subroutine writes the sector array to the file.

This section of the program contains the subroutines that read the geometry data for each geometric input shape, as well as holes and arrays, and writes the data to a scratch file.

ROTRNS - This subroutine translates and rotates the location and orientation of a geometric body based on the geometric keyword read in and the data relating to the body read from the subroutines ORIGIN and ROTATE.

SPHERE - This subroutine reads the radius of a sphere using DREAD and generates the coefficients of the quadratic equation that represents the surface of the sphere.

CYLNDR - This subroutine reads the radius and top and bottom surfaces of a cylinder using DREAD and generates the set of quadratic equations that represent the surface of the cylinder.

ECYL - This subroutine reads the x-radius, y-radius, and top and bottom surfaces of an elliptical cylinder using DREAD and generates the set of quadratic equations that represent the surface of the cylinder.

ELLIPS - This subroutine reads the x-radius, y-radius, and z-radius of an ellipse using DREAD and generates the quadratic equation that represents the surface of the ellipse.

CONE - This subroutine reads the top radius, bottom radius, top surface, and bottom surface using DREAD and generates the set of quadratic equations that represent the surface of the cone section.

FILLGM - This subroutine takes the geometry read in READGM and folds the HOLE data into the appropriate geometry data assays.

HRTRNS - This subroutine translates and rotates the equations of the geometry boundary of a hole.

CUBOID - This subroutine reads the +X, -X, +Y, -Y, +Z, and -Z faces of the cuboid using DREAD and generates the set of quadratic equations that represent the surface of the cuboid.

HXPRSM - This subroutine reads the inscribed radius, the top surface, and the bottom surface of a hexprism using DREAD and generates the set of quadratic equations that represent the surface of the hexprism.

WEDGE - This subroutine reads the x-coordinate of the triangular base, the x and y coordinates of the intersection of the other two sides, and the height of the wedge using DREAD and generates the set of quadratic equations that represent the surface of the wedge.

RHOMB - This subroutine reads the length of a side and the angle between the y-face and the y-axis of the rhomboid using DREAD and generates the set of quadratic equations that represent the surface of the rhomboid.

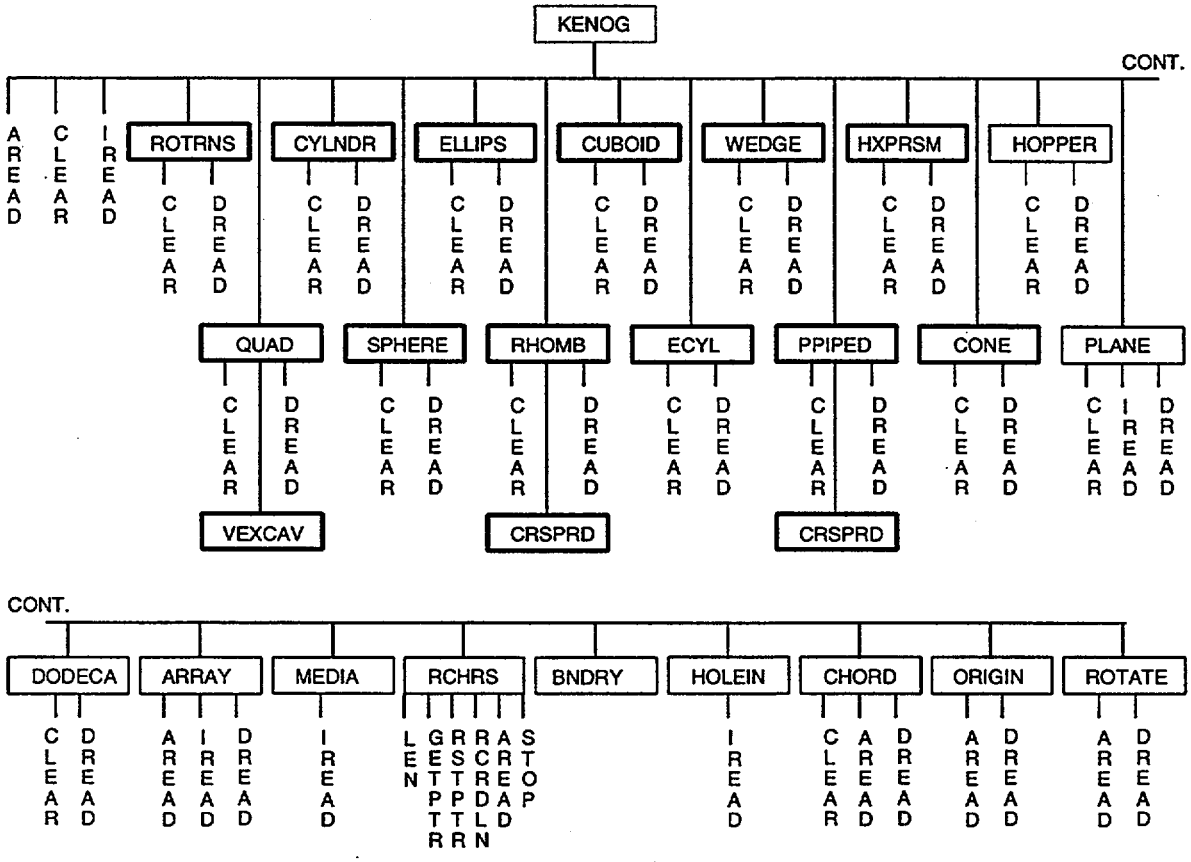


Figure F17.3.7 Flowchart for reading geometry boundary and media data

CRSPRD - This subroutine generates and returns a unit vector from the cross product of two input vectors.

PPIPED - This subroutine reads the x-distance, y-distance, z-distance, the angle between the y-distance and the y-axis, the angle between the z-distance and the z-axis, and the angle between the projection of the z-distance on the x-y plane and the x-axis using DREAD and generates the set of quadratic equations that represent the surface of the parallelepiped.

HOPPER - This subroutine reads the x half-length, y half-length and z surface for the top and the x half-length, y half-length, and z surface for the bottom using DREAD and generates the set of quadratic equations that represent the surface of the hopper.

PLANE - This subroutine reads the keywords and associated dimensions that represent a plane using AREAD and DREAD, respectively, and generates the quadratic equation that represents the planar surface.

QUAD - This subroutine reads the keywords and associated coefficients of a quadratic using AREAD and DREAD respectively.

VEXCAV - This function determines if the quadratic equation specified using QUAD is convex or concave.

DEDUCE - This subroutine reads the inscribed radius of the dodecahedron using DREAD and generates the quadratic equation that represents the surface of the dodecahedron.

ARRAY - This subroutine reads the array number and definition vector of the region containing the array using IREAD, the keyword for placing the array in the unit using AREAD, and the origin of the unit at the specified array position relative to the origin of the unit containing the array using IREAD and DREAD.

MEDIA - This subroutine reads the media number, the importance region, and the definition vector for the sector using IREAD.

BNDRY - This subroutine reads the definition vector that describes the boundary of a unit.

HOLEIN - This subroutine reads the unit number inserted in the hole and the definition vector of the region containing the hole using IREAD.

CHORD - This subroutine reads the chord orientation keyword using AREAD and the plane location using DREAD, then generates the quadratic equation that represents the surface of the plane and adds it to the set of equations that represents the geometric shape being developed.

ORIGIN - This subroutine reads the set of translation keywords using AREAD and the values associated with each keyword using DREAD. These values are then stored in common block /RTRN/.

F17.3.4.3 *Mixing Table Data*

This section deals with reading the mixing table data (Figure F17.3.8).

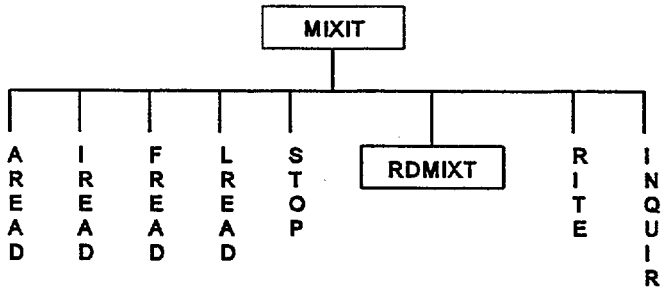


Figure F17.3.8 Flowchart for reading mixing table data

ROTATE - This subroutine reads the set of rotation keywords using AREAD and the values associated with each keyword using DREAD. These values are then stored in common block /RTRN/.

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 IDs. LREAD is used to determine if the next digit is a numeric digit. 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. RITE is called to write the mixing table information on the direct-access device and INQUIR is called to return the value of the next direct-access record.

LREAD - This library routine returns a value of "true" if the next character is a numeric digit. Otherwise, a value of "false" is returned.

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.

F17.3.4.4 *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 IDs used in the problem (Figure F17.3.9). The plot input data block is described in Sect. F17.4.11.

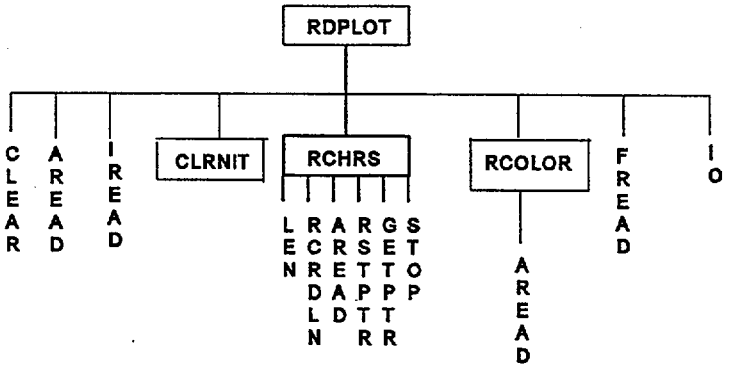


Figure F17.3.9 Flowchart for reading plot data

RDPLOT - This subroutine uses CLEAR to initialize the data arrays. 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. The intrinsic function LEN is used to determine the length of a character variable. 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.

CLRNIT - This subroutine initializes the color tables for used in creating color plots.

RCOLOR - This subroutine is used to redefine the red, green, and blue components of the plot color tables. AREAD is used to read the color table position and the new red, green, and blue values of that position.

F17.3.4.5 *Biasing Data*

This section of the program reads the biasing data used in the problem (Figure F17.3.10). The biasing input data block is described in Sect. F17.4.7.

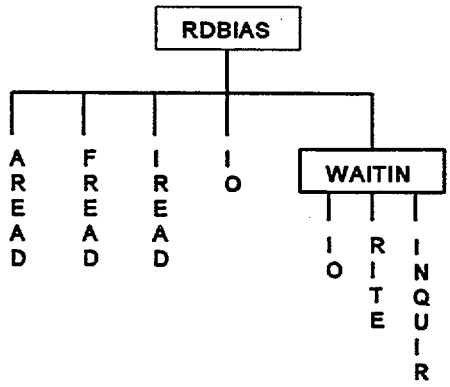


Figure F17.3.10 Flowchart for reading biasing data

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 by the user. 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. INQUIR is then called to return the value of the next direct-access record.

F17.3.4.6 *Restart Data*

This section of the program reads restart information from the restart data file (Figure F17.3.11).

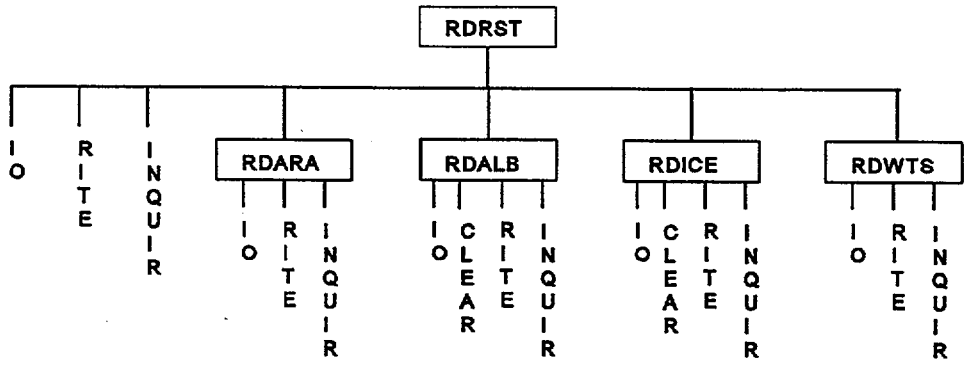


Figure F17.3.11 Flowchart for reading restart data

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 IDs 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. The library routine INQUIR is called to return the value of the next direct-access record.

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. INQUIR returns the value of the next direct-access record.

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. INQUIR returns the value of the next direct-access record.

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. INQUIR returns the value of the next direct-access record.

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. INQUIR returns the value of the next direct-access record.

F17.3.4.7 *Generate 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 (Figure F17.3.12).

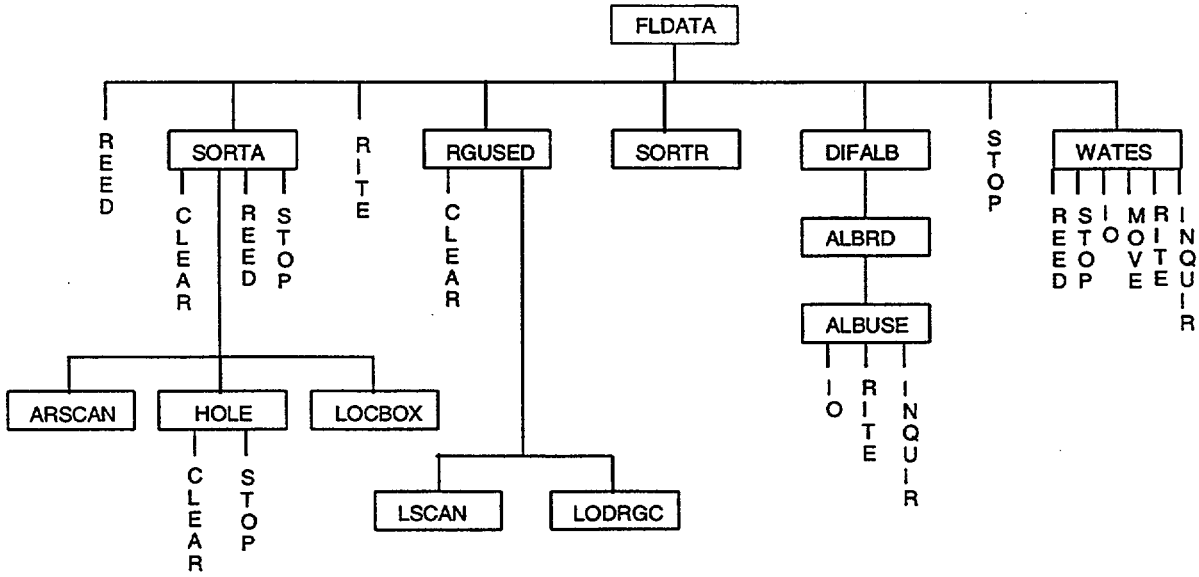


Figure F17.3.12 Flowchart for generating remaining data

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. The library routine, STOP, may be called to print an error message. 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.

ARSCAN - This function, which is called from sorta, is used to determine if array data have been entered if an array was specified.

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 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 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 is used in the unit orientation array. A value of false is returned if the unit 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. The library routine INQUIR is called after each call to RITE to return the value of the next direct-access record.

WATES - This subroutine reads the biasing input data block from the direct-access data file and reads the KENO-VI 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, 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. INQUIR returns the value of the next direct-access record after each call to RITE.

F17.3.5 CREATE A MIXED CROSS-SECTION DATA FILE

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 (Figure F17.3.13). This data file has the same format as an ICE mixed cross-section MORSE/KENO-VI library.[1]

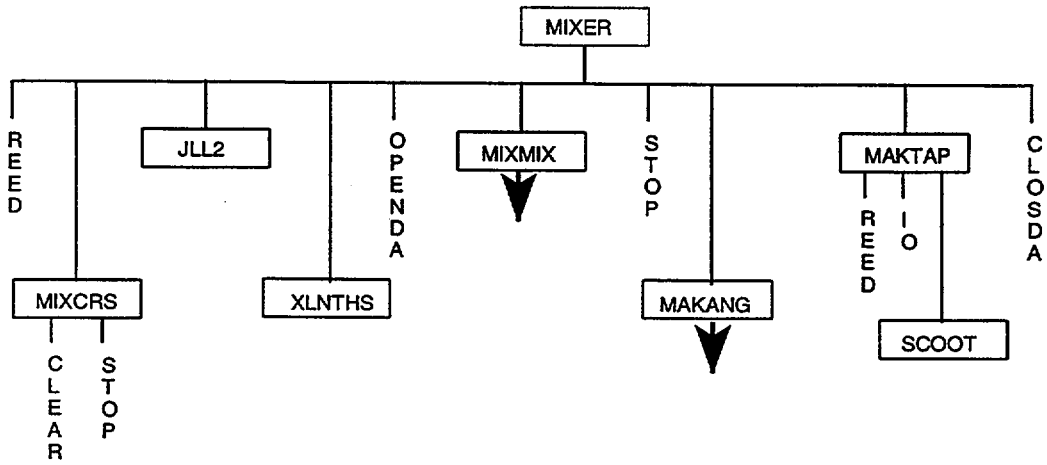


Figure F17.3.13 Flowchart for mixing cross sections

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. Calls to STOP are made to print error messages if more space is needed to mix the cross sections or to process the angles.

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. F17.3.5.1 for a more detailed description of subroutine MIXMIX.

MAKANG - This subroutine controls the generation of angles and probabilities. See Sect. F17.3.5.2 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 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.

F17.3.5.1 *Cross-Section Mixing*

This section of the program produces mixed cross sections (Figure F17.3.14).

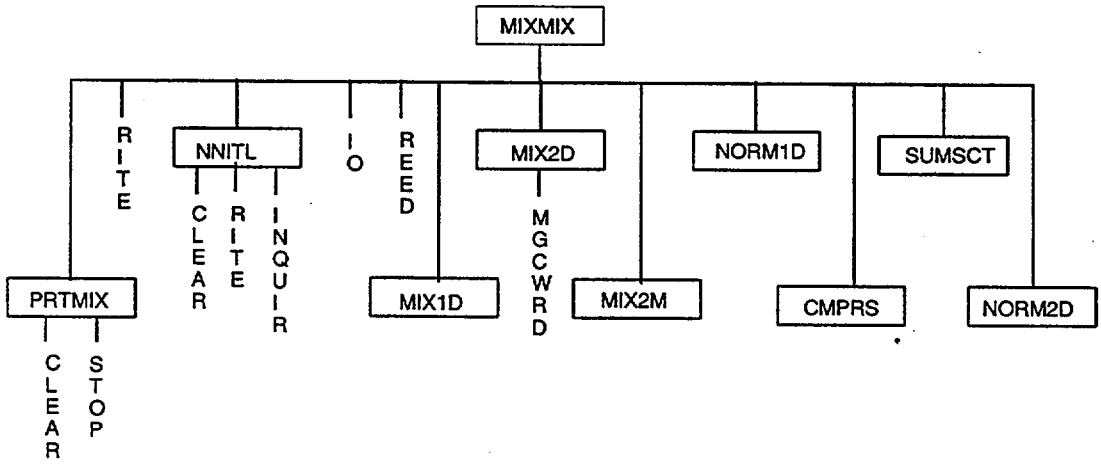


Figure F17.3.14 Flowchart of mixing operations

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. CLEAR is called to zero the array, RITE writes the array on the direct-access device, and INQUIR returns the value of the next direct-access record.

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 $\upsilon\Sigma_f\phi dE$ 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. The library routine, MGCWRD, is called to create the magic word for accessing the mixture cross sections. 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 $\upsilon\Sigma_f$ and the fission spectrum.

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 $(2_\ell + 1)$.

F17.3.5.2 *Generate Angles and Probabilities*

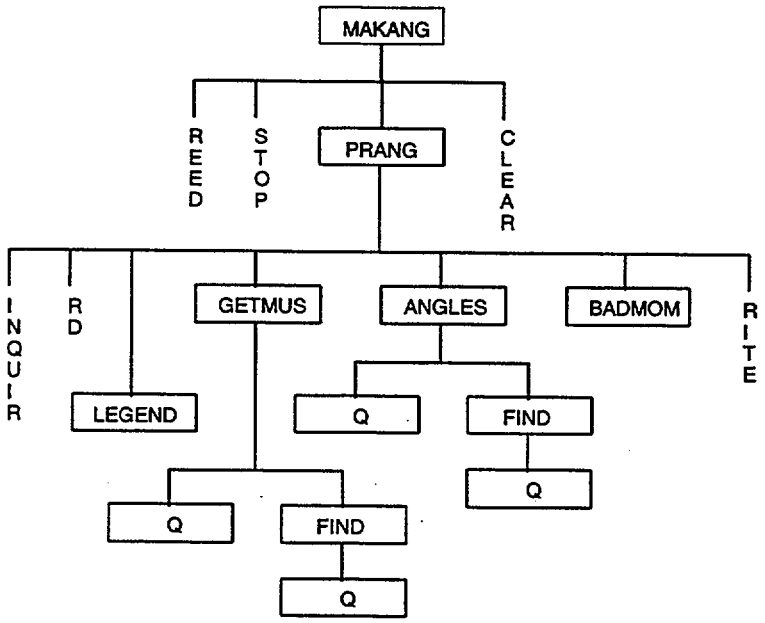


Figure F17.3.15 Flowchart of angle and probability generation

This portion of the program produces the angles and probabilities for each mixture.

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.

F17.3.6 WRITE CROSS SECTIONS ON DIRECT-ACCESS FILE

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 (Figure F17.3.16).

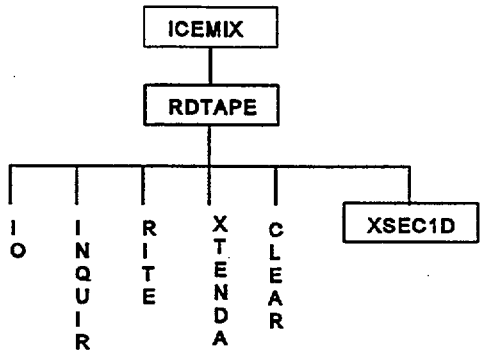


Figure F17.3.16 Flowchart of mixed cross-section processing

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-VI, and writes the data on the direct-access data file. The library routine SQRT is used in calculating the inverse velocities used by KENO-VI. 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-VI on the direct-access data file. INQUIR returns the value of the next direct-access record after each call to RITE. 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.

F17.3.7 WRITE INPUT DATA ON RESTART FILE

This section of the program is responsible for writing all data except the calculated results on the restart data file (Figure F17.3.17). 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. F17.4.3. The calculated results are written on the restart data file later in the program. The restart data file is used for restarting a problem.

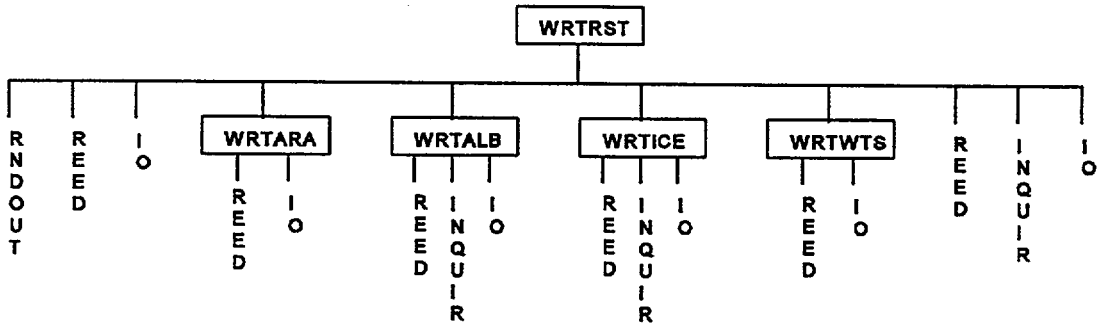


Figure F17.3.17 Flowchart for writing data on the restart data file

WRTRST - This subroutine writes the input data on the restart data file. RNDOUT is called to return the seed of the next random number. The problem title and parameter data are in the first record written on the restart data file. The array containing the IDs of the 1-D cross sections is then written. The geometry region data, mixing table data, extra 1-D data, biasing data, start data, plot data, and energy and inverse velocities are written. The unit orientation header record is written and WRTARA is called to write the unit orientation data. The albedo header record is written, and WRTALB is called to write the albedo data. The cross-section header record is written and WRTICE is called to write the cross sections. The header record for user-supplied weighting data is written, and WRTWTS is called to write the user-supplied weighting or biasing data on the restart data file.

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 called to load the direct-access pointers for the albedo data blocks. INQUIR is called to return the value of the next direct-access record. REED is used to read the data from the direct-access data file, and IO is used to write the data 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 the data on the restart data file.

WRTICE - This subroutine is called from WRTRST to write the cross-section data block on the restart data file. REED is called to load the direct-access pointers for the cross-section data blocks. INQUIR is called to return the value of the next direct-access record. The library routine REED is used to read the rest of 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.

F17.3.8 GENERATE ALBEDO CROSS-SECTION CORRESPONDENCE TABLES

This section of the program generates the correspondence tables that are necessary to correlate the energy group structures of the cross sections and albedos (Figure F17.3.18). It is invoked only if differential albedos are used in the problem.

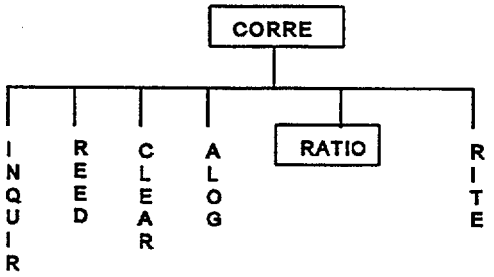


Figure F17.3.18 Flowchart for generating correspondence arrays

CORRE - This subroutine is called only if differential albedos are used. The library routine REED is called to read arrays. The library routine INQUIR is called to return the value of the next direct-access record. The library routine CLEAR is used to initialize 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.

F17.3.9 GENERATE SUPERGROUPED DATA

The function of this section of the program is to create supergrouped data from the energy-dependent input data (Figure F17.3.19). 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.

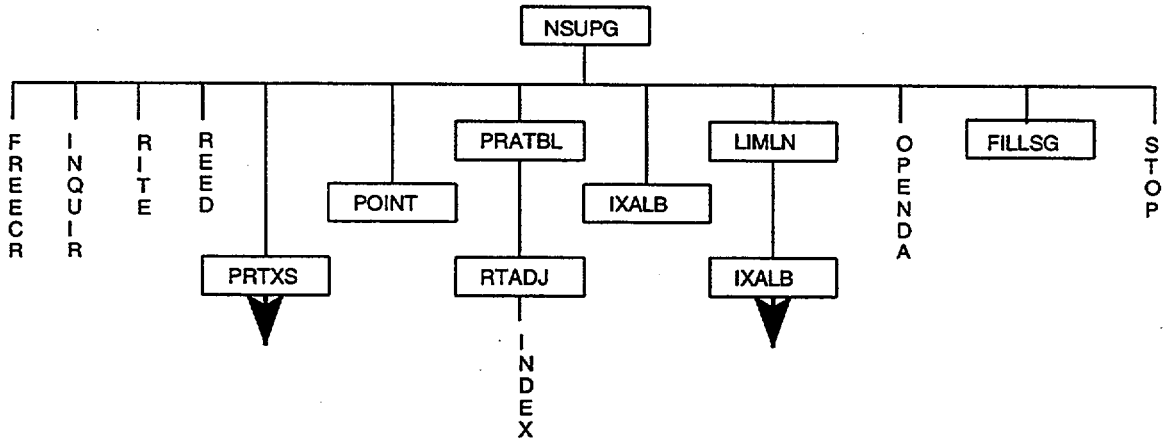


Figure F17.3.19 Flowchart of supergroup creation

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. The library routine INQUIR is used to return the value of the next direct-access record. PRTXS is called to print the cross-section information as described in Sect. F17.3.9.1. POINT is called to determine the storage requirements of the nonsupergrouped data and create pointers to access these data. PRATBL is called to print the first portion of the additional information table 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. F17.3.9.2. After the supergrouped data have been written on the direct-access device, the library routine FREECR is called to release the space that is not needed for the problem.

PRTXS - This subroutine prints the cross-section information as described in Sect. F17.3.9.1.

POINT - This subroutine determines the storage requirements of the nonsupergrouped data and creates pointers to access data within the nonsupergrouped storage array.

PRATBL - This subroutine prints the additional information table to the computer output.

RTADJ - This subroutine right adjusts the albedo boundary condition names which are read in as left-adjusted data. RTADJ utilizes the FORTRAN-supplied intrinsic function INDEX to locate the first blank character.

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. F17.3.9.2.

F17.3.9.1 *Print Macroscopic Cross Sections*

This portion of the program prints the macroscopic cross-section data for materials used in the problem description (Figure F17.3.20).

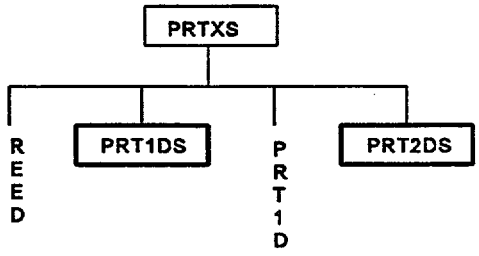


Figure F17.3.20 Flowchart for printing macroscopic cross sections

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, then 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 the library routine PRT1DS is called to print them. If the transfer arrays are to be printed, REED is called 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 2-D variable length array in a compact manner.

F17.3.9.2 *Write Supergroup Data File*

This portion of the program collects the group-dependent data by supergroup and writes them on the direct-access supergroup data file (Figure F17.3.21).

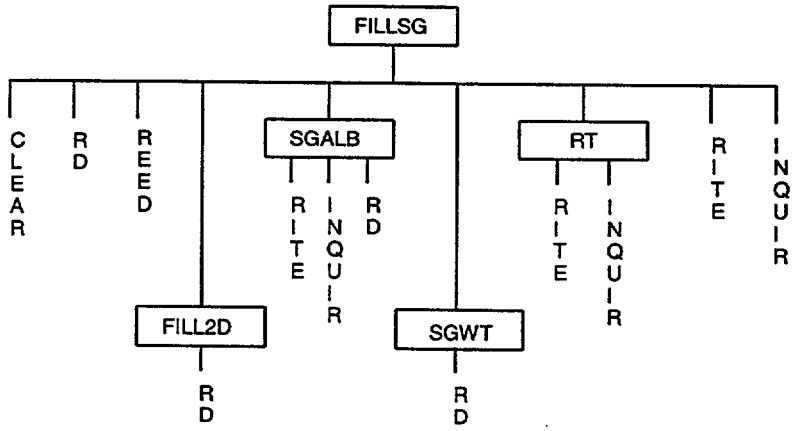


Figure F17.3.21 Flowchart for loading supergrouped data

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. INQUIR is called as necessary to return the value of the next direct-access record.

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. INQUIR is used to provide the value of the next direct-access record. 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 using the library routine, RITE. INQUIR is used to return the value of the next direct-access record, after the call to RITE.

RITE - This library routine is used to write the group-dependent data for the supergroup on the direct-access supergroup data file.

F17.3.10 PROCESS 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 (Figure F17.3.22).

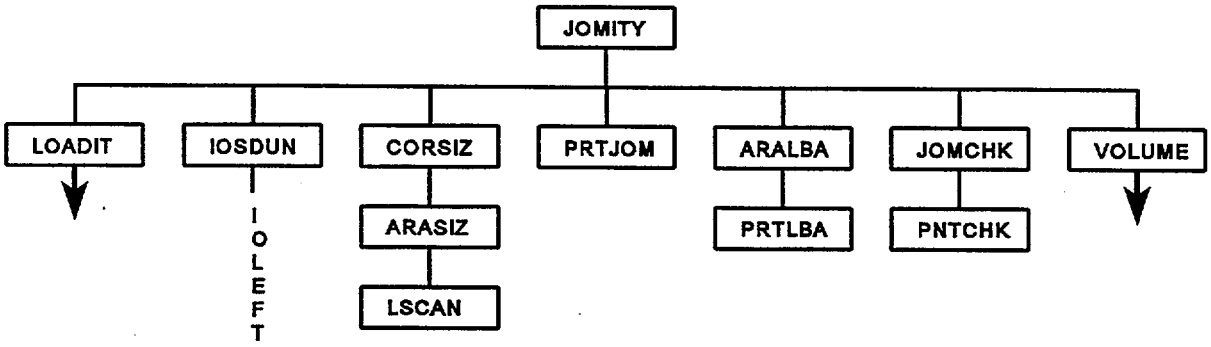


Figure F17.3.22 Flowchart for processing geometry

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 nonsupergrouped portion of the albedo data. Section F17.3.10.1 contains a more detailed description of the procedure.

IOSDUN - This subroutine returns the number of IOs used. The library routine IOLEFT returns the number of input/output requests left before the system cancels the job.

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 has been used in the array. ARASIZ also checks to ensure 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 is used in the unit orientation array. A value of false is returned if the unit was not used in the unit orientation array.

PRTJOM - This subroutine prints the geometry data used in the problem.

ARALBA - This subroutine calls 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 may print a warning message associated with one or more lattices.

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. F17.3.10.2 for additional details.

F17.3.10.1 *Load Data From the Direct-Access File*

This portion of the program loads data from the direct-access data file into permanent memory (Figure F17.3.23).

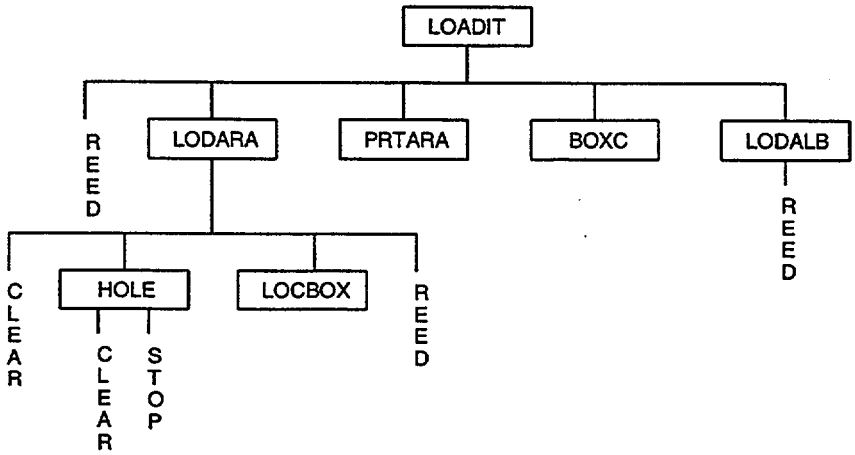


Figure F17.3.23 Flowchart for loading data from direct access

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 units 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 unit correspondence array, and LODALB is called to load the nonsupergrouped 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 ensure 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 at a given position in the unit orientation array.

PRTARA - This subroutine prints a table of the arrays used in the problem. The array number, the number of units in the x, y, and z directions, and the nesting level is printed for each array.

BOXC - This subroutine uses the number of units and the geometry region number corresponding to the first and last geometry region of each unit to generate the unit correspondence array which contains the unit number for each geometry region. This information 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.

F17.3.10.2 *Calculate 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 (Figure F17.3.24).

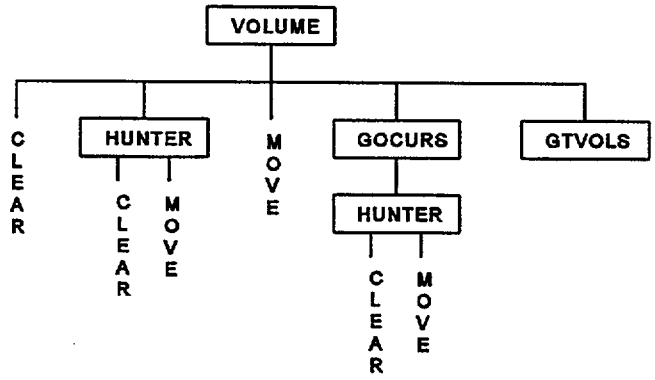


Figure F17.3.24 Flowchart for calculating volumes

VOLUME - This subroutine attempts to calculate the volume of each region for every unit that is used in the problem. It then attempts to calculate 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. CLEAR is called to initialize the array for the total volume of each mixture used in the problem prior to calculating and printing those totals.

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.

F17.3.11 LOAD BIASING OR WEIGHTING DATA

This portion of the program is executed only if the input parameter data contain PWT=YES, as described in Sect. F17.4.3. If biasing data are to be printed, the program loads and prints the average weight array (Figure F17.3.25).

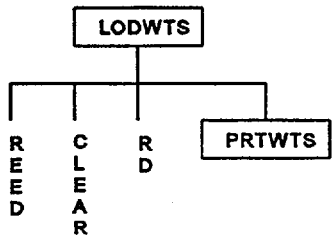


Figure F17.3.25 Flowchart for loading biasing data

LODWTS - This subroutine is responsible for printing the bias IDs vs the material IDs used in the problem and for loading and printing the biasing or weighting data. The library routine REED is called to load the bias IDs used in the problem from the direct-access device. 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 them. 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-averaged array for each biasing region in a compact fashion.

F17.3.12 GENERATE PLOT

This portion of the program generates character and color plots of a 2-D slice through the geometry (Figure F17.3.26). As many plots as are desired can be printed.

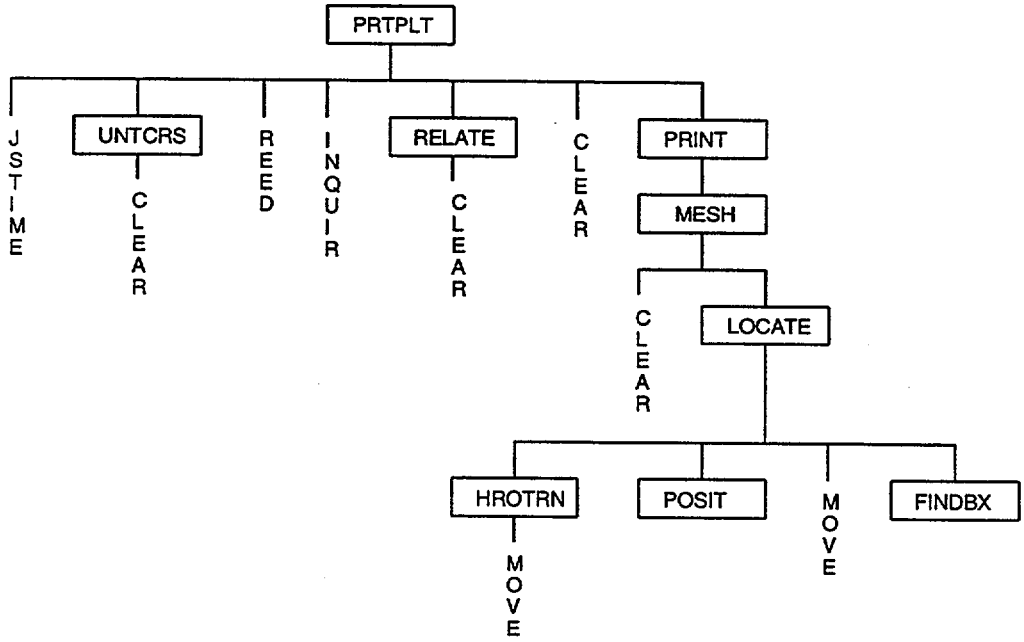


Figure F17.3.26 Flowchart for plots

PRTPLT - This subroutine controls the generation of the character or color plots. The library routine CLEAR is called to zero the necessary space. Subroutine UNTCRS is called to generate a unit correspondence array. REED is used to load the plot data from direct access. INQUIR is called to return the value of the next direct-access record. The plot title is printed, and subroutine RELATE is called to print a heading for the symbol map and print the symbol map. The plot coordinates, direction cosines and number of symbols across and down the plot as well as the step intervals, are printed. Then subroutine PRINT is called to generate the actual plot.

UNTCRS - This subroutine calls CLEAR to zero the unit correspondence array. It then loads the appropriate unit number in the appropriate position of the array.

RELATE - This subroutine calls CLEAR to zero the reverse correspondence array. It then prints the plot header telling whether a mixture, unit, or bias ID map is to be printed. It loads the array that correlates the symbols to be used in the plot with the mixture numbers, bias ID numbers, or unit

numbers that were used in the problem. The array is then printed to facilitate user interpretation of the plot.

PRINT - This subroutine determines the number of pages that will be needed to output the plot. Subroutine MESH is called to load the appropriate mixture numbers, unit numbers or bias ID numbers for each line of the plot. Then PRINT outputs the line of symbols/color 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 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 plot. Subroutine POSIT is called to determine the region that contains the specified mesh point. If array data are used, MOVE is called to load the current array data into the array stack. If the mesh point is located within a hole, HROTRN is called to determine the position within the unit contained in the hole that contains the mesh point. 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 that contains the mesh point. POSIT is used to determine the geometry region that contains the mesh point. If the mesh point is in a region that contains a hole, the coordinates of the mesh point are translated to the hole and POSIT is called again.

HROTRN - This subroutine rotates and translates the coordinates to locate the position of a specific point in a hole.

FINDBX - This subroutine locates the position in an array that contains a specified point.

LOCBOX - This function returns the unit at a given position in an array.

POSIT - This subroutine determines the region within a unit that contains a specified point.

F17.3.13 PROCESS HISTORIES BY SUPERGROUP

This section of the program is where the tracking of the individual histories is done, one supergroup at a time (Figure F17.3.27).

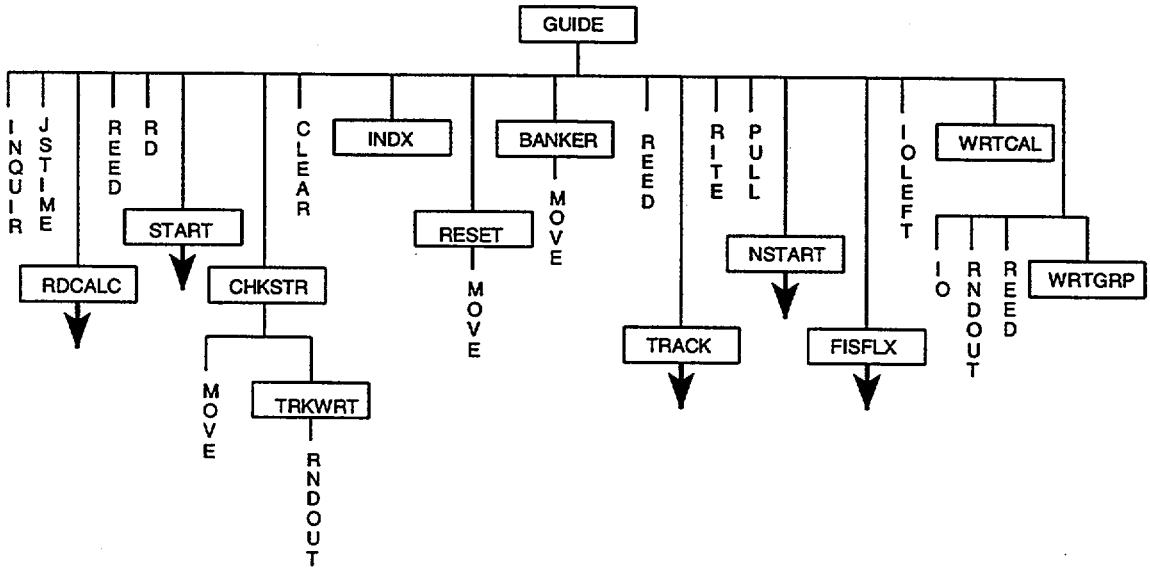


Figure F17.3.27 Flowchart of tracking routines

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/Os and time for the tracking procedure. INQUIR is used to return the value of the next direct-access record. SQRT is used in estimating the lower limit of the 99% confidence interval of the sample distribution of k_{eff}.

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_{eff}. 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/Os remain to ensure 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-VI. The following table is provided to assist in understanding the functions performed by the subroutines called by GUIDE:

Subroutine	Function	Condition
INQUIR	Set direct-access pointer	Always
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	Always
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

INQUIR - This library routine returns the value of the next direct-access record. It is called to obtain the pointer for the cross-section data.

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. F17.3.13.1 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. F17.3.13.2 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_{eff} 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. F17.4.3, 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_{eff} 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. F17.3.13.3 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 step 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. F17.3.13.4 for specific details.

FISFLX - This subroutine calculates statistics for k_{eff}, the matrix k_{eff}, fissions, absorptions, leakages, and fluxes. Section F17.3.13.5 contains additional details.

IOLEFT - This library routine is called to determine if sufficient I/Os 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. F17.4.3, 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_{eff} values 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_{eff} 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.

F17.3.13.1 *Load Calculated Restart Data*

This section of the program loads the calculated data such as k_{eff} values and fluxes from the restart data file if the starting generation number is greater than 1 (Figure F17.3.28).

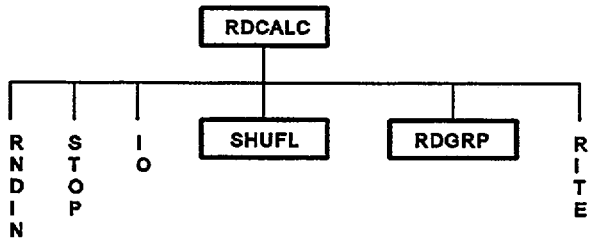


Figure F17.3.28 Flowchart for loading calculated restart data

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. STOP is called if more storage is needed to hold the neutron bank. IO is used to load data from the restart data file and SHUFL is called to sort and store the neutron bank. 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.

STOP - This library routine is called to write an error message and stop if the available storage is not large enough to hold the neutron bank.

IO - This library routine is used to load matrix k_{eff} information if it is to be calculated.

SHUFL - This subroutine is called to sort and store the neutron bank.

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.

F17.3.13.2 *Generate Initial Source Distribution*

This portion of the program is responsible for generating the initial source distribution (Figure F17.3.29) in accordance with information specified in the start data (see Sect. F17.4.8).

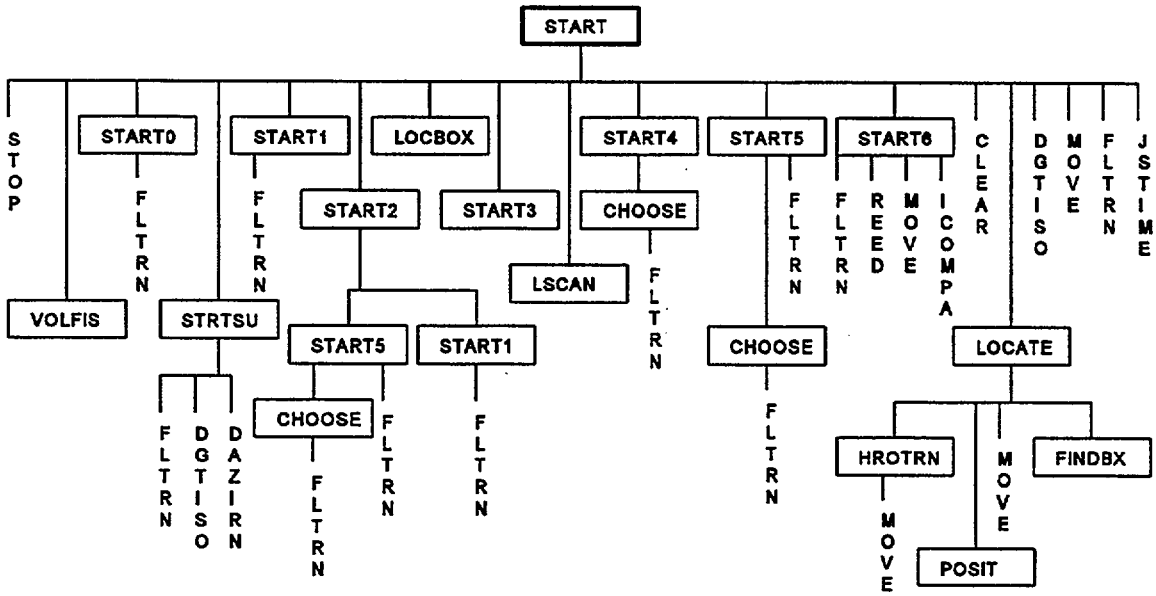


Figure F17.3.29 Flowchart for providing initial source distribution

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. STOP is called to print error messages if the start data are geometrically invalid or the fraction started as a spike is invalid. LSCAN is used to determine if the specified unit is in the global array. 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. F17.4.8). 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, DGTISO 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 error messages.

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.

LSCAN - This logical function returns a value of true if the specified unit is used in the problem.

START0 - This subroutine is called from START to generate a uniform initial source distribution in a cuboidal volume. The source distribution is generated by choosing points uniformly throughout the volume using the library routine FLTRN, and discarding points that do not occur in fissile material. See Sect. F17.4.8 and Table F17.4.6 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. DAZIRN, DGTISO, and FLTRN are used to choose points uniformly throughout a noncuboidal volume.

DAZIRN - This library routine provides the sine and cosine of a random azimuthal angle.

DGTISO - This library routine provides the direction cosines of an isotopically 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 DARSIN and FLTRN are used to provide the cosine distribution. See Table F17.4.6 for details of the various initial starting distributions.

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 F17.4.6. 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 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. REED is called to load the start type 6 data from the direct-access device. MOVE is called to move the start data into common. FLTRN is used to randomly supply additional starting points from existing ones. ICOMPA is used to determine if the starting point is identical to the previous one. See Table F17.4.6 for start data specifications.

LOCATE - This subroutine determines the geometry region that contains the specified starting point. HROTRN is used to determine the points location in a unit that is contained within a hole. FINDBX is used to determine the point's location in an array and the unit number at that location.

HROTRN - This subroutine locates the position of a specific point in a unit that is contained within a hole.

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 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.

F17.3.13.3 *Track Individual Histories*

This section of the program does the actual tracking of the individual histories (Figure F17.3.30).

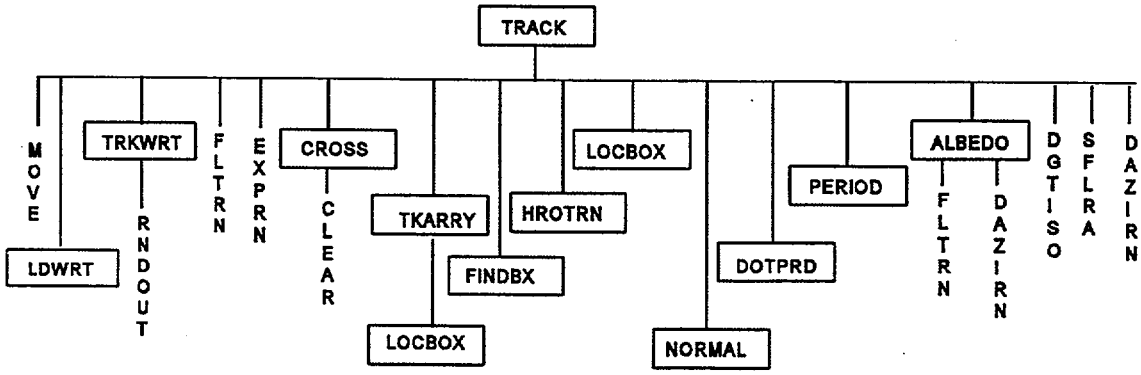


Figure F17.3.30 Flowchart for tracking individual histories

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. CROSS is called to determine if a boundary crossing has occurred. TKARRY is used to track a particle inside an array. FINDBX is used to locate the position of a particle in an array. HROTRN is called to transfer the coordinates of a particle from its current unit to the unit within the hole it is entering or vise versa. 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 DGTISO 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 an isotropic scattering. DAZIRN provides the sine and cosine of a random azimuthal angle for use in the anisotropic scattering treatment of a collision.

DOTPRD - This function returns the dot product of two direction vectors.

PERIOD - This subroutine processes periodic albedos on paired planes. Given a position and direct vector on one plane, it calculates the new position and direct vector on the paired plane.

NORMAL - This subroutine returns the normal to a surface at a given point on the surface.

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. F17.4.3). 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.

CROSS - This important subroutine is responsible for processing 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.

TKARRY - This subroutine is used to track a particle inside an array. It determines if a particle changes position within an array or crosses of the array.

FINDBX - This subroutine locates the position in an array that contains a specified point.

HROTRN - This subroutine is used to transfer the coordinate system of a particle when it enters or leaves a hole. Upon entering a hole the particle location is transferred to the coordinate system of the unit within the hole. Upon exiting a hole the particle location is transferred to the coordinate system of the unit surrounding the hole.

LOCBOX - This function returns the unit for a specified position in an array.

ALBEDO - This subroutine 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 1 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.

DGTISO - 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.

DAZIRN - 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. F17.3.31. 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 and in the figure by all capital letters. These descriptive names correspond to the functions depicted in the flowchart.

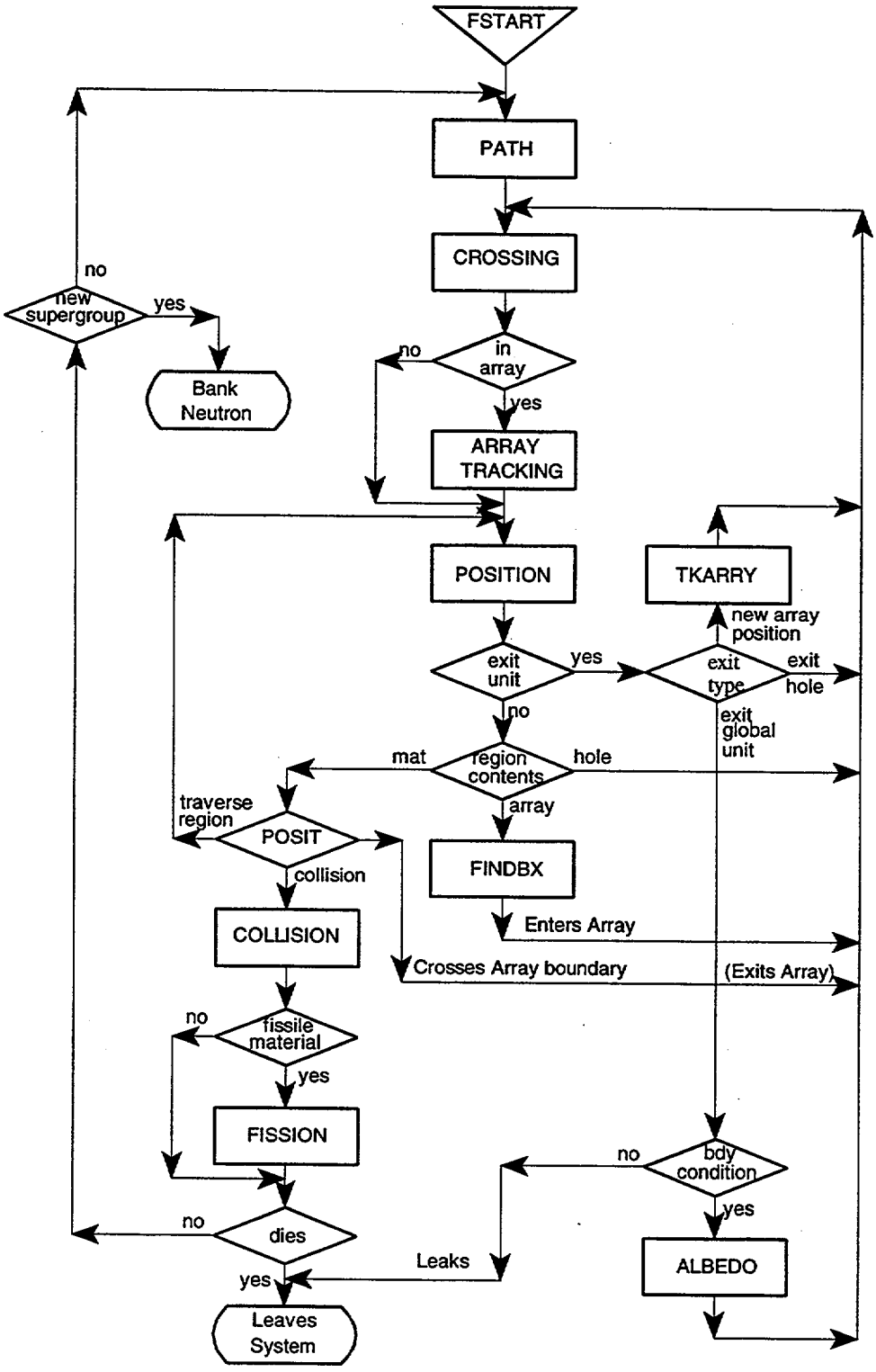


Figure F17.3.31 Logic flowchart for subroutine TRACK

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. The "CROSSING" portion of TRACK calculates the distance required for the neutron to cross each region boundary in the unit it is currently in. If the neutron is in a unit that is part of an array the "ARRAY TRACKING" portion of TRACK calculates the distance required for the neutron to exit the array boundary. The "POSITION" section of TRACK determines which region in the unit the neutron is in. If the neutron is outside its current unit three possibilities exist: (1) it could be changing to a new array position, (2) it could be exiting a hole, or (3) it could be exiting the global unit. If the neutron is changing array positions it calls subroutine TKARRY which transforms the coordinates of the neutron to those of the unit in the array position containing the neutron. If the neutron is exiting a hole or changing array position, the unit number is updated and it returns to "CROSSING." If the neutron is exiting the global unit it can enter an albedo boundary or leak from the system. If it leaks from the system, the history is terminated. If it enters an albedo boundary it enters the "ALBEDO" portion of TRACK which reflects the neutron back into the global unit.

If the "POSITION" section of TRACK determines the neutron is inside the current unit, the contents of the region containing the neutron is checked. If the region contains a hole, the unit number and coordinate system of the neutron are updated and the code returns to the "CROSSING." If the region contains an array, the subroutine FINDBX is called, which determines the neutron's position relative to the unit it is entering. The neutron then returns to the "CROSSING" section of TRACK.

If the region the neutron is in contains a material, "POSIT" determines if the neutron traverses the region or has a collision in the region. If the neutron traverses the region, it is returned to the "NEUTRON POSITION" section of TRACK; otherwise it enters the "COLLISION" section of TRACK which updates its weights, position, direction, and energy group. If the collision occurs in a region containing fissile material, it enters the "FISSION" section of TRACK. After a collision, the particle can die, move to a new supergroup, in which case it is banked for later, or return to the "PATH" section of TRACK and continue.

"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 "COLLISION" 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.

"CROSSING"- This portion of TRACK initializes all the pointers and variables for the unit currently containing the neutron and determines which quadratic equations in the QC array need to be processed. It then checks to see if the current unit is in an array, and, if so, goes to the "ARRAY TRACKING" section of TRACK. Subroutine CROSS is then called to calculate the CA array which contains crossing distances from the current neutron position to the boundary of each region in the unit.

"ARRAY TRACKING"- This portion of TRACK initializes all the pointers and variables for the unit currently containing the array and determines which quadratic equations in the QC array need to be processed. Subroutine CROSS is then called to calculate the CA array which contains crossing distances from the current neutron position to the boundary of each region in the unit. The distance from the current neutron position needed to cross the array boundary is determined.

"POSITION"- This portion of TRACK determines which region, if any, in the current unit contains the neutron. This section first determines which region in the unit contains the neutron, if any, and sets a flag to .true. if the neutron is inside the unit and .false. if it is outside the unit. Error messages are printed if the neutron is found to be in more than one region. These error messages may be caused by incorrectly specified regions (i.e., specifying the same volume in more than one region) or by not explicitly specifying which region contains the boundary between regions.

If the neutron is found to be inside the unit under consideration, the contents of the region are examined. The contents of the region containing the neutron will fall into one of three categories: (1) material, (2) hole, (3) or array. If the neutron is in a region containing material, it transfers to the "POSIT" section of the code. If the neutron is in a region containing a hole, a flag is set indicating an inward crossing into a hole and subroutine HOLROT is called, which translates and rotates the coordinates of the neutron to those of the unit in the hole. The neutron is then transferred to the "CROSSING" section of TRACK.

If the neutron is in a region containing an array, the array number is determined and the appropriate pointers and variables are initialized. If the neutron is already in an array, subroutine MOVE is used to store the current array location in the array stack. Subroutine FINDBX is then called to determine the array position of the neutron and transform the coordinates of the neutron into those of the unit in the array position it is crossing into. The neutron is then transferred to the "CROSSING" section of TRACK.

If the neutron is found not to be in the unit, there are four possibilities: (1) it has undergone an albedo reflection and roundoff error has put it outside the unit, (2) it has crossed out of the global unit, (3) it has crossed out of a hole, or (4) it has changed positions in an array. If the neutron crossed out of the global unit, it is transferred to the "ALBEDO" section of TRACK which determines if it leaks from the system or is reflected back into the system.

If the neutron has crossed out of a hole, a flag is set indicating an outward crossing out of a hole and subroutine HOLROT is called, which translates and rotates the coordinates of the neutron to those of the unit surrounding the hole. The neutron is then transferred to the "CROSSING" section of TRACK. If tracking information was asked for, subroutine TRKWRT is called to print pertinent information. The neutron is then transferred to the "CROSSING" section of TRACK.

If the neutron has changed position in an array, subroutine TKARRY is called to translate the neutron to the coordinate system of the unit in the new array position. If tracking information was asked for, subroutine TRKWRT is called to print pertinent information. The neutron is then transferred to the "CROSSING" section of TRACK.

"POSIT" - This portion of TRACK determines the material in the region, determines the distance the neutron needs to travel to exit the unit, and computes the amount of mean free paths needed to travel to exit the region. If tracking data were asked for, subroutine TRKWRT is called to print pertinent data. If the number of mean free paths needed to exit the region is greater than the number of mean-free paths assigned in "PATH," the neutron is transferred to the "COLLISION" section of TRACK. Otherwise the distance the neutron has traveled is updated, and the amount of mean-free paths already traveled is computed to exit the region. If tracking data were asked for, subroutine TRKWRT is called to print pertinent data. If the neutron is in an array, the neutron is checked to see if it crosses the array boundary. If it does not, the neutron is transferred to the "POSITION" section of TRACK. If the array boundary is crossed, the region number, position, unit number, and distance traveled are updated. If the array was nested, subroutine MOVE is called to restore the unit location in the surrounding array. If tracking information was indicated, subroutine TRKWRT is called to print pertinent information. The neutron is then transferred to the "CROSSING" section of TRACK.

"COLLISION" - When a history has a collision, the processing is done in this portion of TRACK. The number of mean-free paths used, distance traveled, and neutron position are updated. 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_{eff} values 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 ensure that the neutron bank has adequate space for another history. If it is full, a message is written and a counter incremented. If the counter reaches 50, the program is terminated and an error message is printed. 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 DAZIRN, 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 DGTISO. 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 ensure 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_{eff} 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_{eff}. 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.

"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.

F17.3.13.4 *Provide 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 (Figure F17.3.32).

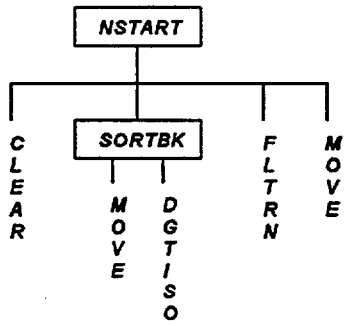


Figure F17.3.32 Flowchart for providing the next-generation source

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 DGTISO is used to provide direction cosines from an isotropic distribution. Then the neutron number, weight and age are initialized.

F17.3.13.5 *End-of-Generation Processing*

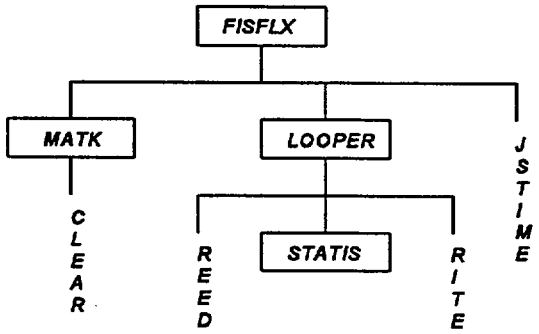


Figure F17.3.33 Flowchart for end of generation processing

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_{eff}, the running average value of k_{eff} and its deviation, and the matrix k_{eff} and its deviation are processed and printed. MATK is called to process the matrix k_{eff}(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_{eff} 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_{eff} 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.

F17.3.14 END OF PROBLEM PROCESSING

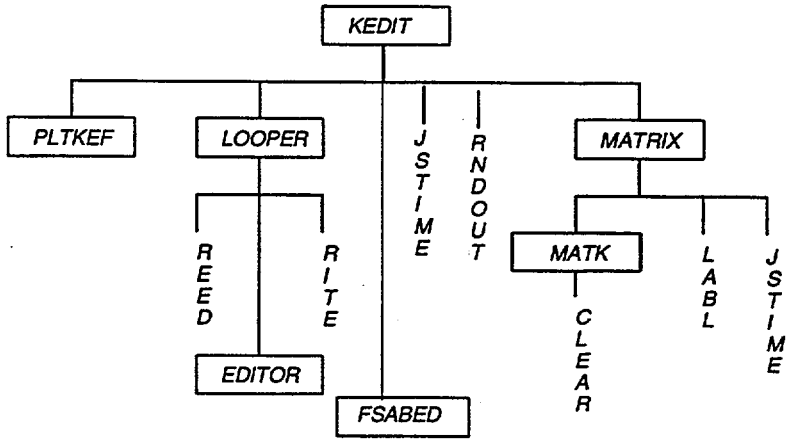


Figure F17.3.34 Flowchart for end-of-problem processing

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_{eff} 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_{eff} as a function of the number of generations. It also prints a plot of the average value of k_{eff} 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 total 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_{eff} 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_{eff} vs generation and a plot of the average value of k_{eff} vs generations skipped. The library routines MIN, MAX, and SQRT are used to generate the k_{eff} 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.

FSABED - This subroutine is called from KEDIT to calculate and print the total fissions and absorptions by region of the problem after all generations have been completed.

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_{eff}. It is called if one or more matrix options were specified in the parameter data (see Sect. F17.4.3). If matrix information was collected, MATK is called to calculate cofactor k_{eff} values. 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. F17.4.3). 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_{eff} values.

F17.3.15 PRINT FLUXES

This portion of the program is responsible for printing the fluxes at the completion of a problem (Figure F17.3.35).

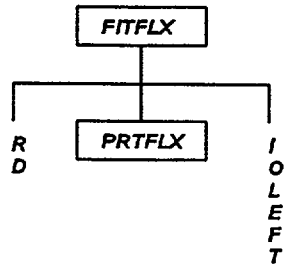


Figure F17.3.35 Flowchart for printing fluxes

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. The library routine IOLEFT is called to determine the number of input/output requests remaining before the system cancels the job. 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.

F17.3.16 REFERENCES

1. S. K. Fraley, *Users Guide for ICE-II*, ORNL/CSD/TM-9/R1, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., July 1977. Also see "ICE-S: Module to Mix Multigroup Cross Sections," Sect. F8 of *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation*, NUREG/CR-0200, Rev. 4 (ORNL/NUREG/CSD-2/R4), Vols. I, II, and III, February 1995). Available from Radiation Shielding Information Center, Oak Ridge National Laboratory as CCC-545.

2. "Appendix B, Generalized Gaussian Quadrature," *The MORSE Code - A Multigroup Neutron and Gamma-Ray Monte Carlo Transport Code*, ORNL-4585, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 1970. Also see "MORSE-SGC/S for the SCALE System," Sect. F9 of *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation*, NUREG/CR-0200, Rev. 4 (ORNL/NUREG/CSD-2/R4), Vols. I, II, and III, February 1995). Available from Radiation Shielding Information Center, Oak Ridge National Laboratory as CCC-545.

F17.4 KENO-VI DATA GUIDE

KENO-VI may be run "stand alone" or as a part of a SCALE criticality safety sequence. If KENO-VI 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 V.a format library, also called a Monte Carlo format cross-section library. If KENO-VI uses an AMPX working- format library, a mixing table data block must be entered. If an ICE mixed cross-section MORSE/KENO V.a library is used, a mixing table data block is not entered, and the mixtures specified in the KENO-VI geometry description must be consistent with the mixtures created in ICE. These are the entries in the 11$[MUD] array in the ICE input data.

If KENO-VI is run as part of a SCALE criticality safety sequence, the mixtures are defined in the CSAS data (Sect. C6.4) and a mixing table data block cannot be entered in KENO-VI. Furthermore, the mixture numbers used in the KENO-VI geometry description must correspond to those defined in the standard composition data of the CSAS input. A mixture number of 500 **must** be used in the KENO-VI geometry description in order to use a cell-weighted mixture. A cell-weighted mixture is available only in SCALE sequences that use XSDRN to perform a cell-weighting calculation.

F17.4.1 KENO-VI INPUT OUTLINE

The data input for KENO-VI is outlined below. Defaulted data for KENO-VI 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 F17.4.1. 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 F17.4.1 Types of input data

Type of data	First four 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[a]	MIXT or MIX
Plot[a]	PLOT or PLT or PICT

[a]MIX and PLT must include a trailing blank that is considered part of the keyword.

Two data records **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 80 characters including blanks). A title **must be entered.** See Sect. F17.4.3.

(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 a parameter is entered more than once, the last value is used. The words **END PARA** terminate the parameter data. See Sect. F17.4.3.

$(n_1)...(n_9)$ 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. F17.4.4.

(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 (3-D) 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. F17.4.5.

(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. F17.4.7.

(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. F17.4.6.

(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. F17.4.8.

(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-VI is run as a part of a SCALE criticality safety sequence. See Sect. F17.4.10.

(n_7) **READ X1DS extra 1-D cross-section IDs END X1DS**

Enter the IDs of any extra one-dimensional (1-D) 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. F17.4.9.

(n_8) **READ PLOT plot data END PLOT**

Enter the data needed to provide a two-dimensional (2-D) character or color plot of a slice through a specified portion of the 3-D geometrical representation of the problem. Plot data must begin with the words **READ PLOT** and terminate with the words **END PLOT**. See Sect. F17.4.11.

(n_9) **END DATA must be entered**

Terminate the data for the problem.

F17.4.2 PROCEDURE FOR DATA INPUT

This section is a brief list of the input data for KENO-VI. Additional information concerning KENO-VI data input may be found in Sect. F17.5. The first data record **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 DEFINITION VECTOR, GEOMETRY WORD, MIXTURE NUMBER, BIAS ID,* and *REGION DIMENSIONS,*

(2) albedo data such as *FACE CODES* and *ALBEDO NAMES,* and

(3) weighting data such as *BIAS ID NUMBERS,* etc.

F17.4.3 TITLE AND PARAMETER DATA

TITLE . . . A title must be entered.

title length is 80 characters, including blanks.

PARAMETER DATA . . . Enter only those parameters whose values you wish to change. The commonly changed parameters are *TME, GEN,* and *NPG.* Seldom changed parameters are *NBK, NFB, XNB, XFB, 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*.

SIG = *sigma* This is used to terminate a problem when a standard deviation of *sigma* has been reached. The problem will still terminate when the specified number of generations have been run, even if the standard deviation is still greater than *sigma*.

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 = 1000000. (This value 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. *BEG* must be 1 greater than the number of generations retrieved from the restart file.

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 the 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. F17.4.11.

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 *uP1D.* 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 2-D 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 = NO. 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 = NO

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 = NO

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 = NO

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 plots 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. F17.4.11. 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 tracking 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 a MORSE/KENO-VI format mixed cross-section library, 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 format cross-section 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 if *BEG* > 1

WRS = *wstrt* logical unit number for writing restart data, default = 0
A nonzero value must be entered if *RES* > 0.

EXAMPLE: READ PARAM NPG=203 FLX=YES END PARAM

F17.4.4 GEOMETRY DATA

GEOMETRY REGION DATA . . . geometric arrangements in KENO-VI are achieved in a manner similar to using a child's building blocks. Each building block is called a UNIT. An array or lattice is constructed by stacking these units. Once an array or lattice has been constructed, it can be placed in a unit by using the ARRAY specification.

Each UNIT in an array or lattice has its own coordinate system. All geometry data used in a problem are correlated to the absolute coordinate system by specifying a global unit. A global unit that encloses the entire system must be specified for every problem including a single unit problem. Units are constructed of combinations from over a dozen predefined shapes plus any other shapes that can be defined using quadratic equations. These geometric shapes are allowed to intersect each other, producing regions that are then described by specifying them as internal or external to the geometric shapes. The geometric shapes can be translated anywhere within the system and rotated to any orientation. Figure 17.4.1 shows some acceptable geometric configurations.

Special options are provided to enhance the basic geometry package. 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 within a geometry/material region (see Fig. F17.4.2). The region boundary must coincide exactly with the boundary of the unit placed in the hole. The hole option allows complex units to be constructed separately and then placed in an existing unit. Since a particle must check every region to determine its location within a unit, using holes to contain complex sections of a problem may decrease the CPU time needed for the problem. An arbitrary number of HOLES can be placed in a unit in combination with a series of surrounding regions.

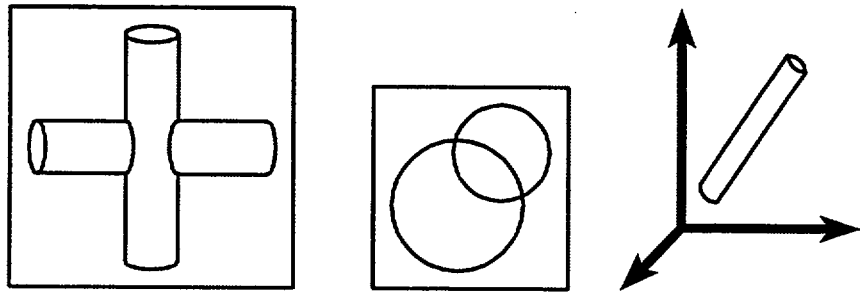


Figure F17.4.1 Examples of geometric configurations allowed in KENO-VI

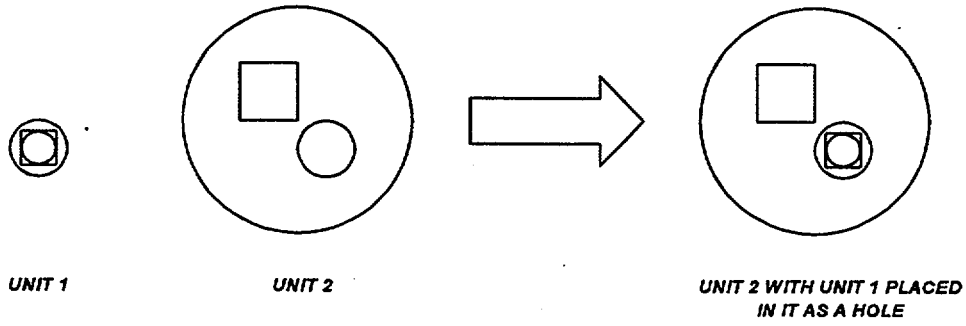


Figure F17.4.2 Example demonstrating HOLE capability in KENO-VI

Lattices or arrays are created by stacking UNITS. Units that have a rectangular parallelepiped outer region can be used to construct a rectangular array. Units that have a hexprism outer region can be used to construct a triangular-pitched matrix. In either case, the adjacent faces of adjacent units stacked together must match exactly. See Sect. F17.5.6.4 for additional clarification and Figs. F17.4.3 and 17.4.4 for typical examples.

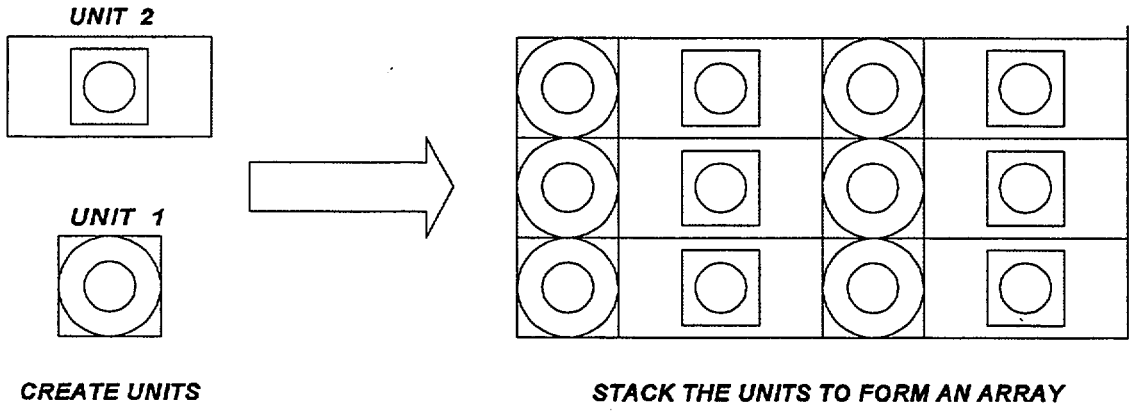


Figure F17.4.3 Example of rectangular-pitched array construction

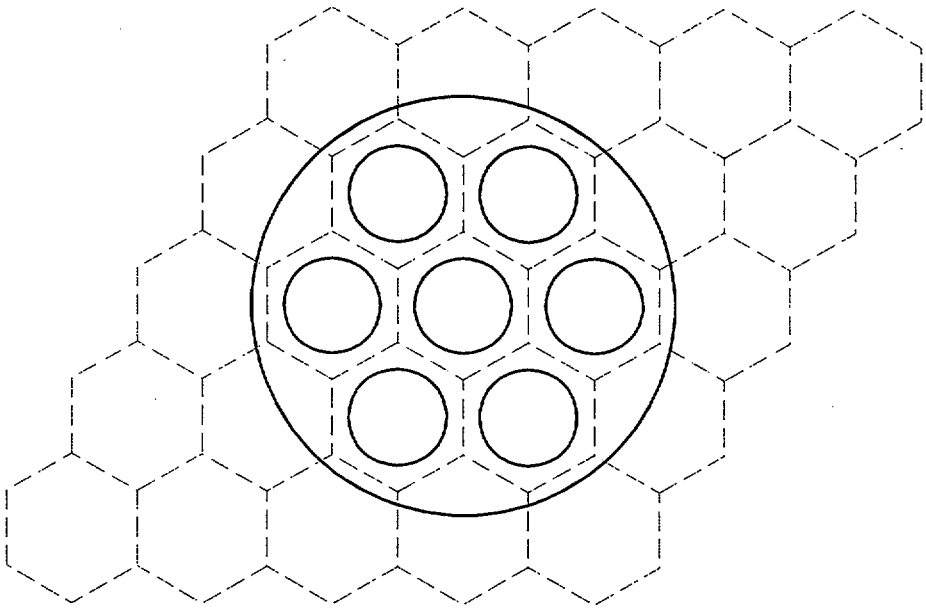


Figure F17.4.4 Example of triangular-pitched array construction

The ARRAY option is provided to allow placing an array or lattice within a unit. An array is placed in a unit by placing it directly into a geometry/material region. The region boundary containing the array must coincide with or be contained within the array boundary. The region boundary becomes the array boundary. A particle enters or leaves the array when the region boundary is crossed. Multiple array may be placed directly in a UNIT by placing them in separate regions. Arrays can also be placed within a unit 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. F17.4.5 for an example of an array composed of units containing holes and arrays.

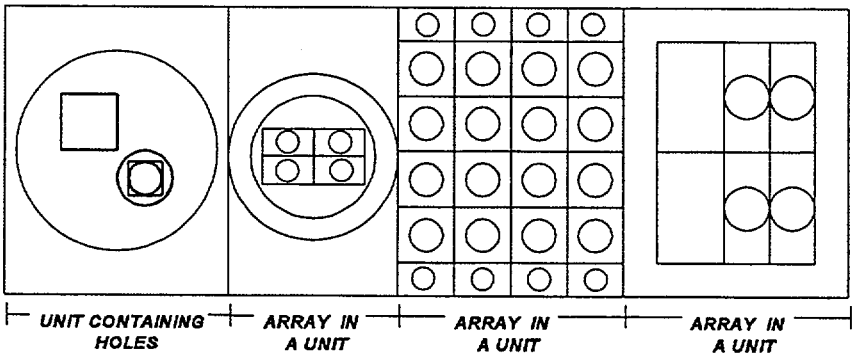


Figure F17.4.5 Example of an array composed of units containing arrays and holes

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 the user wishes to define must be entered. See Sect. F17.5.6 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 the following:

(a) UNIT INITIALIZATION
(b) GEOMETRY REGION DESCRIPTION
(c) 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) GLOBAL SPECIFICATION
fgeom
GLOBAL

The word GLOBAL, when entered, must be followed by one or more blanks, the word UNIT, and a UNIT ID NUMBER. Enter the geometry word GLOBAL immediately prior to the UNIT that defines the overall geometric boundaries of the problem. A global unit must be entered for each problem.

(a2) UNIT NUMBER DEFINITION

KEYWORD UNIT ID NUMBER

fgeom	*nbox*
UNIT	enter unit ID number (greater than zero)

The UNIT NUMBER definition assigns the specified unit number to the geometry data that define the unit.

(*b*) *GEOMETRY REGION DESCRIPTION* . . . This type of geometry data consists of geometry records, media records, a boundary record and their associated data. Free-form input is used to enter the data. Options R, *, $, and P from Table F17.4.2 can be used. The first record, called the GEOMETRY RECORD, is entered in the form:

GEOMETRY KEYWORD
fgeom

GEOMETRY RECORD LABEL (integer)
lbl

APPROPRIATE REGION DIMENSIONS (cm)
xx

GEOMETRY MODIFICATION KEYWORD
moddat

APPROPRIATE MODIFICATION INFORMATION
modinf

The second record, called the *MEDIA RECORD*, is entered in the form:

MEDIA KEYWORD
fmed

MIXTURE, HOLE, OR ARRAY ID
mat

BIAS ID, ONLY FOR MIXTURE
imp

REGION DEFINITION VECTOR, ONLY FOR MIXTURE OR ARRAY
i1 i2 i3 i4

The third record, called the BOUNDARY RECORD, is entered in the form:

BOUNDARY KEYWORD
bndry

REGION DEFINITION VECTOR
i1 i2 i3 i4

The geometry records contain information describing the shape and location of each geometric shape. The media records contain information describing how the region boundaries relate to the geometric

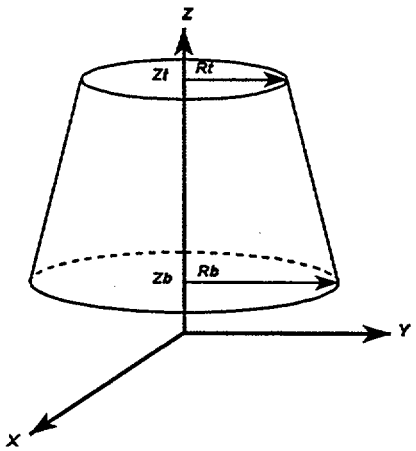
shapes and the contents of each region. The geometry and media records can be combined in any order with a unit. The region definition vector in the media and boundary records refer to the geometry record labels. In the region definition vector, a positive label means the media is inside the specified geometric shape and a negative label means the media is outside the specified geometric shape. Each unit must contain one BOUNDARY record which describes the unit boundary. Only that section of the unit geometry contained within the volume specified by the boundary record is considered part of the unit.

The *GEOMETRY KEYWORD fgeom* is followed by one or more blanks and must be one of the keywords below. A label and the *APPROPRIATE REGIONAL DIMENSIONS* follow the keywords as described below.

CONE, CUBOID, CYLINDER, DODECAHEDRON, ECYLINDER, ELLIPSOID, HEXPRISM, HOPPER, PARALLELEPIPED, PPIPED, PLANE, QUADRATIC, RHOMBOID, SPHERE, WEDGE

Note: *fgeom* may be no more than 12 characters long.

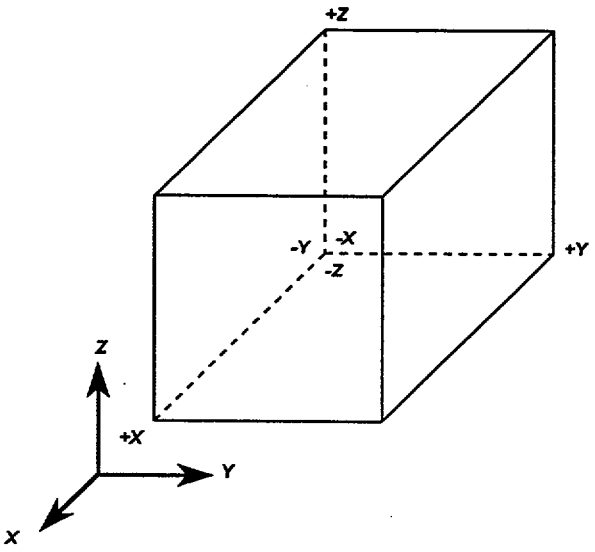
CONE specifies a body consisting of one nappe of a right circular cone. It is defined by specifying the top radius of the cone, Rt, the Z coordinate of the top face, Zt, the bottom radius of the cone, Rb, and the Z coordinate of the bottom face, Zb. Figure 17.4.6 shows the correct input sequence for a cone.



CONE LBL Rt Zt Rb Zt

Figure F17.4.6 Example of cone construction

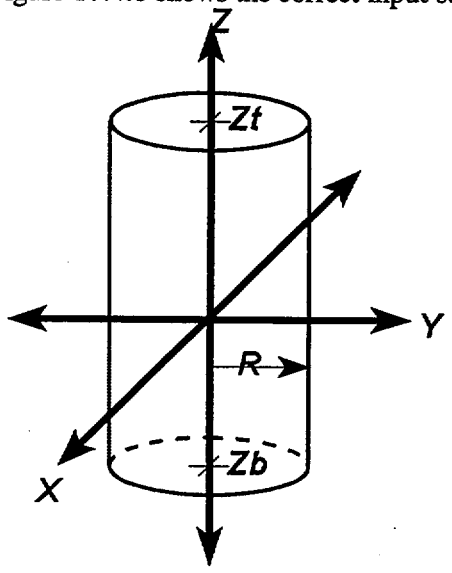
CUBOID specifies a rectangular parallelepiped. It is defined by specifying the +X dimension, -X dimension, +Y dimension, -Y dimension, +Z dimension, -Z dimension. It is perpendicular to the X, Y, and Z axes unless otherwise specified by the option geometry modification data. Figure 17.4.7 shows the correct input sequence for a cuboid.



CUBOID LBL +X -X +Y -Y +Z -Z

Figure F17.4.7 Example of cuboid construction

CYLINDER specifies a right circular cylinder. It is defined by specifying the radius of the cylinder, R, the Z coordinate of the top face, Zt, and the Z coordinate of the bottom face, Zb. Its centerline must lie on the Z axis, unless otherwise specified by the optional geometry modification data. Figure 17.4.8 shows the correct input sequence for a cylinder.



CYLINDER LBL R Zt Zb

Figure F17.4.8 Example of cylinder construction

DODECAHEDRON specifies a body whose surface consists of 12 rhombuses of the same size and shape. It is defined by specifying the radius of the inscribed sphere, R. It is centered on the origin in a fixed orientation unless otherwise specified by the optional geometry modification data. Figure 17.4.9 shows the correct input sequence for a dodecahedron.

ECYLINDER specifies a right cylinder with an elliptical cross-section. It is defined by specifying the semiradius along the x-axis, Rx, the semiradius along the y-axis, Ry, the Z coordinate of the top face, Zt, and the Z coordinate of the bottom face, Zb. Its centerline must lie on the Z axis, unless otherwise specified by the optional geometry modification data. Figure 17.4.10 shows the correct input sequence for an elliptical cylinder.

ELLIPSOID specifies a body whose cross-section slices parallel to each of the coordinate axes are ellipses. It is defined by specifying the semiradius along the x-axis, Rx, the semiradius along the y-axis, Ry, and the semiradius along the z-axis, Rz. It is centered about the origin, unless otherwise specified by the optional geometry modification data. Figure 17.4.11 shows the correct input sequence for an ellipsoid.

HEXPRISM specifies a body whose top and bottom faces are hexagons that have the same orientation and are perpendicular to the Z axis. It is defined by specifying the inscribed radius, R, the Z coordinate of the top face, Zt, and the Z coordinate of the bottom face, Zb. Figure 17.4.12 is an example input for a hexprism. See Fig. F17.4.4 for the default orientation used to stack hexprisms in an array.

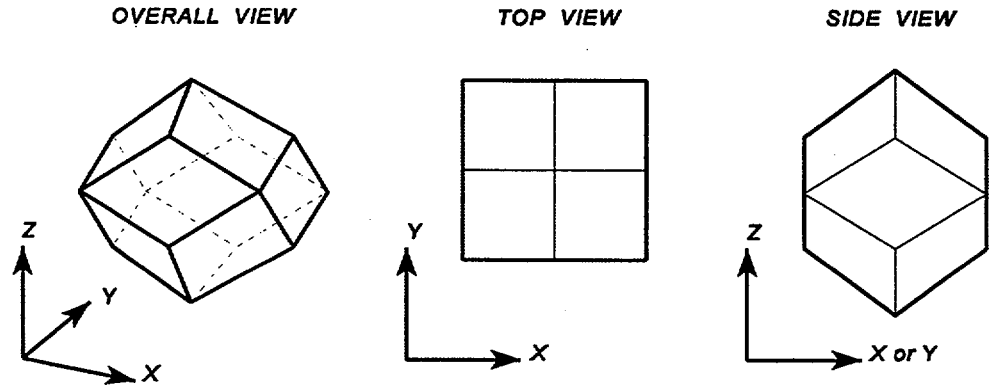
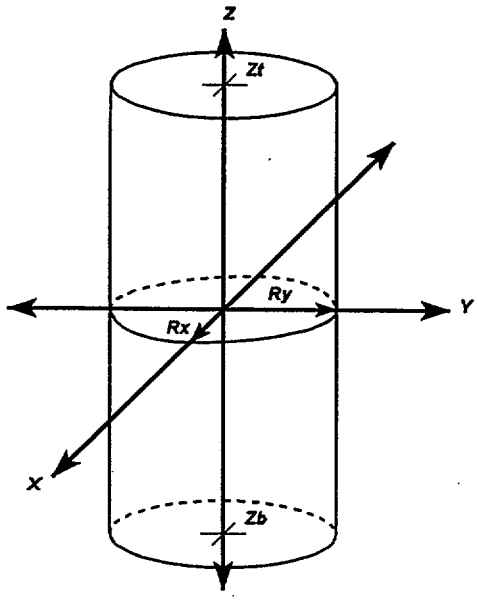
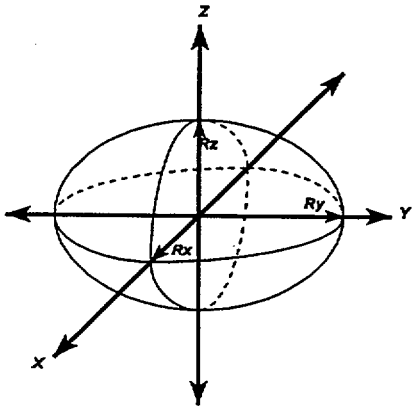


Figure F17.4.9 Example of dodecahedron construction



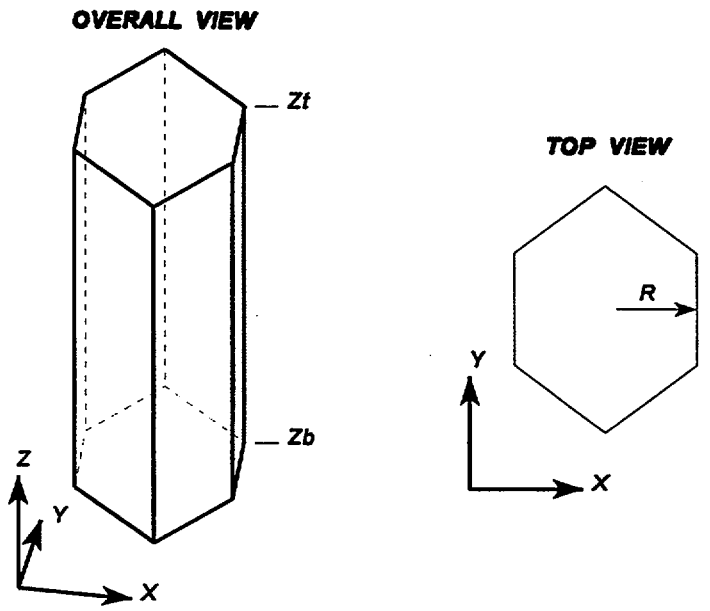
ECYLINDER LBL Rx Ry Zt Zb

Figure F17.4.10 Example of elliptical cylinder construction



ELLIPSOID LBL Rx Ry Rz

Figure F17.4.11 Example of ellipsoid construction



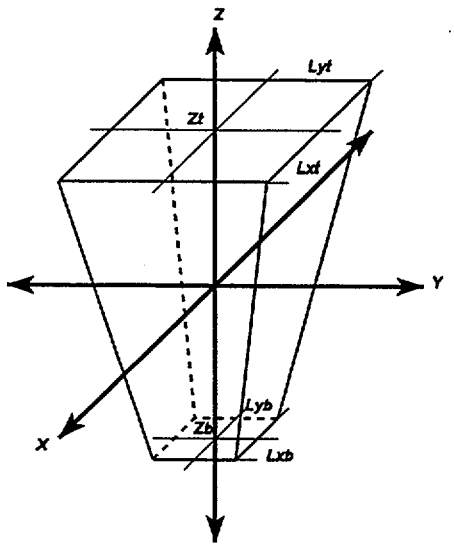
HEXPRISM LBL R Zt Zb

(R is the Inscribed Radius)

Figure F17.4.12 Example of hexprism construction

HOPPER specifies a body whose top and bottom faces are rectangular parallelepipeds centered about the z-axis and parallel to the x and y axes. It is defined by specifying the half-length of the top face along the x-axis, Lxt, the half-length of the top face along the y-axis, Lyt, the Z coordinate of the top face, Zt, the half-length of the bottom face along the x-axis, Lxb, the half-length of the bottom face along the y-axis, Lyb, and the Z coordinate of the bottom face, Zb. Its centerline must lie on the Z axis unless otherwise specified by the optional geometry modification data. Figure 17.4.13 shows the correct input sequence for a hopper.

PARALLELEPIPED or **PPIPED** is a body with six faces composed of parallelograms, whose opposing faces are parallel. It is defined by specifying the length of the faces in the x direction, XDIST, the length of the faces in the y direction, YDIST, the length of the faces in the z direction, ZDIST, the angle between the x-face and the y-axis, PSI, the angle between the y-face and the z-axis, THETA, and the angle between the projection of the top corner nearest the z-axis onto the x-y plane and the x-axis, PHI. The bottom face must lie on the x-y plane at Z = 0 with a corner at the origin unless otherwise specified by the optional geometry modification data. Figure 17.4.14 shows the correct input sequence for a parallelepiped. The angles psi, theta, and phi must be in the range 0 to 90°.



HOPPER LBL Lxt Lyt Zt Lxb Lyb Zb

Figure F17.4.13 Example of hopper construction

PLANE is a surface where any two points can be connected by a straight line entirely contained within plane that divides all space into two regions. The positive side of the plane is the side the normal points to or where the equation $aX + bY + cZ + d > 0$. It is defined by specifying the coefficients of the equation $aX + bY + cZ + d = 0$ using the keywords XPL=a, YPL=b, ZPL=c, and CON=d. Only the nonzero coefficients of the equation need to be specified. Figure 17.4.15 shows the correct input sequence for a plane.

QUADRATIC specifies a surface using a quadratic equation of the form:
$aX^2 + bY^2 + cZ^2 + dXY + eXZ + fYZ + gX + hY + iZ + j = 0$.
It is defined by specifying the coefficients of the above equation using the keywords AQU=a, BQU=b, CQU=c, DQU=d, EQU=e, FQU=f, GQU=g, HQU=h, IQU=i, and JQU=j. Only the nonzero coefficients of the equation need to be specified.

RHOMBOID is a body composed of six identical faces, each one a rhombus. It is defined by specifying the length of the edge of the base along the x-axis, DX and the angle between y edge of the base and the y-axis, psi. Its base is in the xy plane at $Z = 0$, with a corner at the origin unless otherwise specified by the optional geometry modification data. Figure 17.4.16 shows the correct input sequence for a rhomboid.

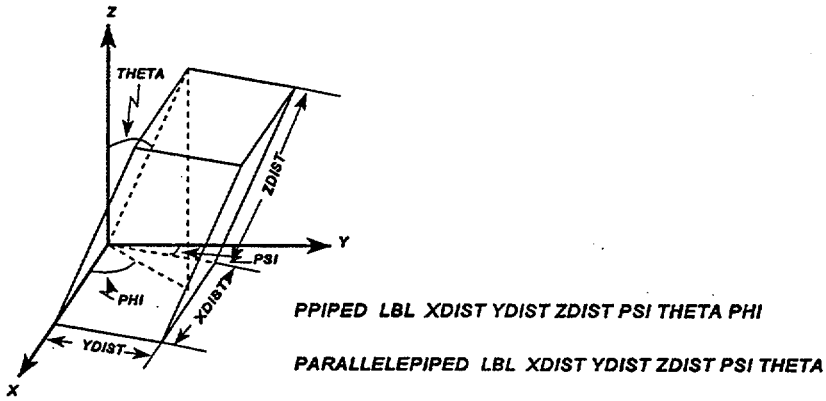


Figure F17.4.14 Example of parallelepiped construction

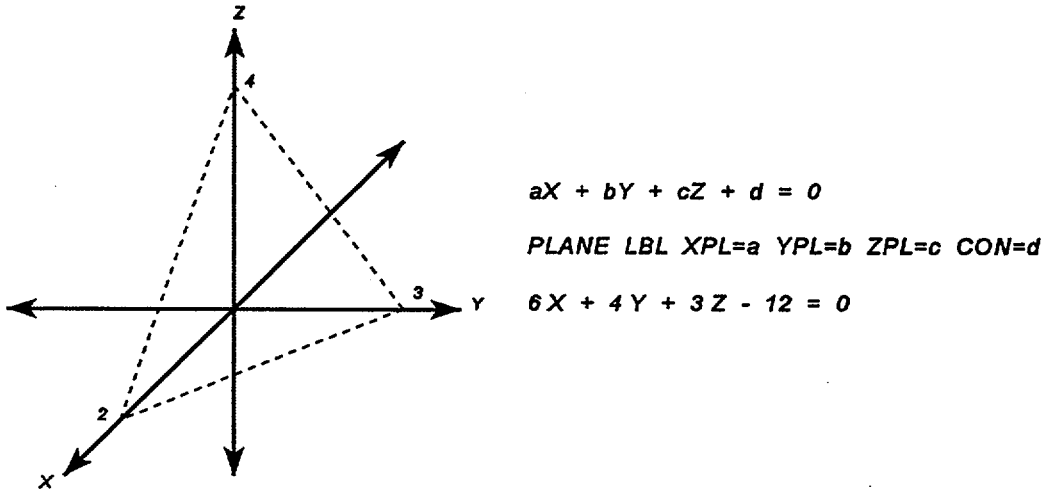
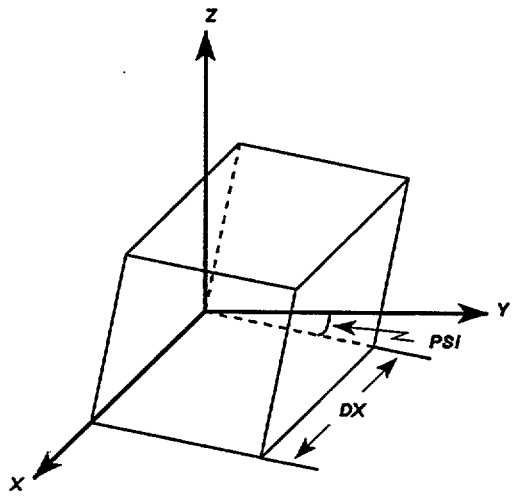


Figure F17.4.15 Example of plane construction



RHOMBOID LBL DX PSI

Figure F17.4.16 Example of rhomboid construction

SPHERE specifies a sphere. It is defined by specifying the radius, R. It is centered about the origin, unless otherwise specified by the optional geometry modification data. Figure 17.4.17 shows the correct input sequence for a sphere.

WEDGE is a right-triangular prism having five faces. The two ends are triangles, and the three sides are rectangles. It is defined by specifying the length of the base along the x-axis, XBASE, the x and y coordinate where the other two sides meet, XPT and YPT, and the length along the z-axis, XLNG. One side is in the XZ plane at Y = 0, and the bottom face is in the XY plane at Z = 0, with a corner at the origin unless otherwise specified by the optional geometry modification data. Figure 17.4.18 shows the correct input sequence for a wedge.

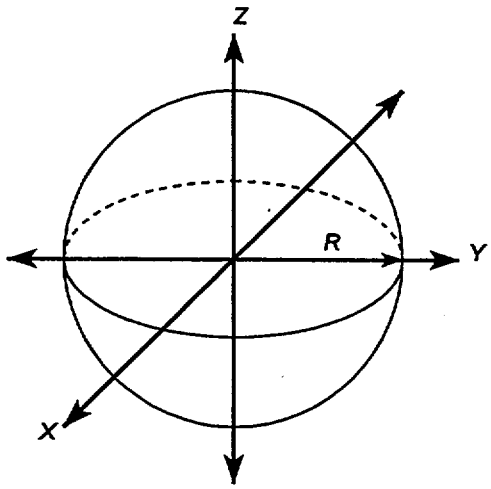


Figure F17.4.17 Example of sphere construction

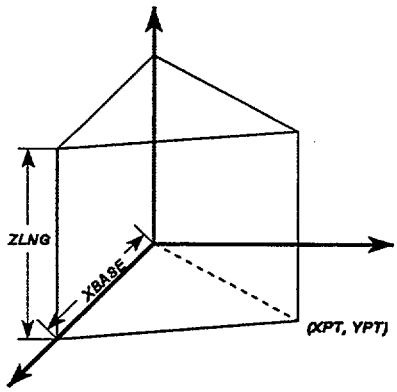


Figure F17.4.18 Example of wedge construction

The *GEOMETRY MODIFICATION KEYWORD, moddat,* is followed by one or more blanks and must be one of the keywords below. The geometry modification data follow the appropriate region dimensions on the geometry record. The modification data can be used in any order or combination with respect to a geometry record. The *APPROPRIATE MODIFICATION INFORMATION, modinf,* follows the keyword as described below.

CHORD, ORIGIN, ROTATE

CHORD is used to provide a place that truncates a geometry shape. It is used by following the keyword with a subordinate keyword, +X=, -X=, +Y=, -Y=, +Z=, or -Z=, which defines the direction of the chord and the intercept of the plane on the axis. A plus in the keyword indicates the positive side of the chord is the inside direction. A minus in the keyword indicates the negative side of the chord is the inside direction. Figure F17.4.19 provides two examples of the use of the CHORD option.

ORIGIN is used to translate the origin of the geometric shape to a specified location. It is used by following the keyword ORIGIN with the subordinate keywords, X=, Y=, and Z= followed respectively by the translated coordinates. The keyword and subordinate keywords may appear multiple times following the geometry or hole record. If an subordinate keyword appears more than once following the ORIGIN keyword the values are summed. Only nonzero coordinates and their respective keywords need to specified. Figure F17.4.20 provides an example of the use of the ORIGIN option.

ROTATE is used to rotate the geometric shape with respect to the origin of the coordinate system of the unit. It is used by following the keyword ROTATE with the subordinate keywords A1=, A2=, and A3= followed respectively by the rotation angle in degrees. The keyword and subordinate keywords may appear multiple times following the geometry or hole record. If an subordinate keyword appears more than once following the ROTATE keyword the values are summed. If ORIGIN and ROTATE data follow the same geometry or hole record, the origin is always rotated prior to translation regardless of what order the data appears. Only nonzero coordinates and their respective keywords need to be specified. Figure F17.4.21 provides an example of the use of the ROTATE option.

The *MEDIA KEYWORD, fmed,* is followed by one or more blanks and must be one of the keywords below.

MEDIA, HOLE, ARRAY

MEDIA is used to define the location of a mixture relative to the geometric shapes in the unit. The keyword, MEDIA, is followed by one or more blanks and the *MIXTURE ID mat*, the *BIAS ID imp*, and the *REGION DEFINITION VECTOR* (i1, i2, ... in). The keyword *BOUNDARY* must precede the media record for the outermost region in a unit. The outermost region must enclose the entire unit. Figure F17.4.22 shows the input for a set of three intersecting spheres in a cuboid.

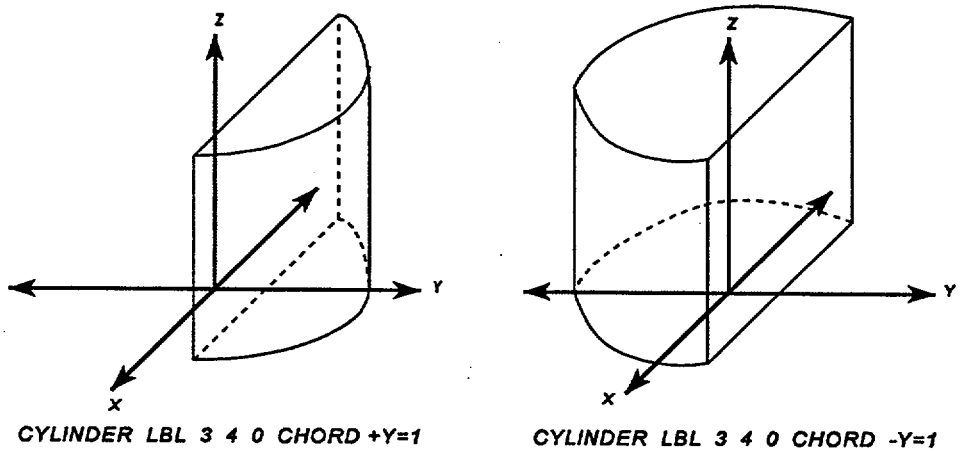


Figure F17.4.19 Examples of the CHORD option

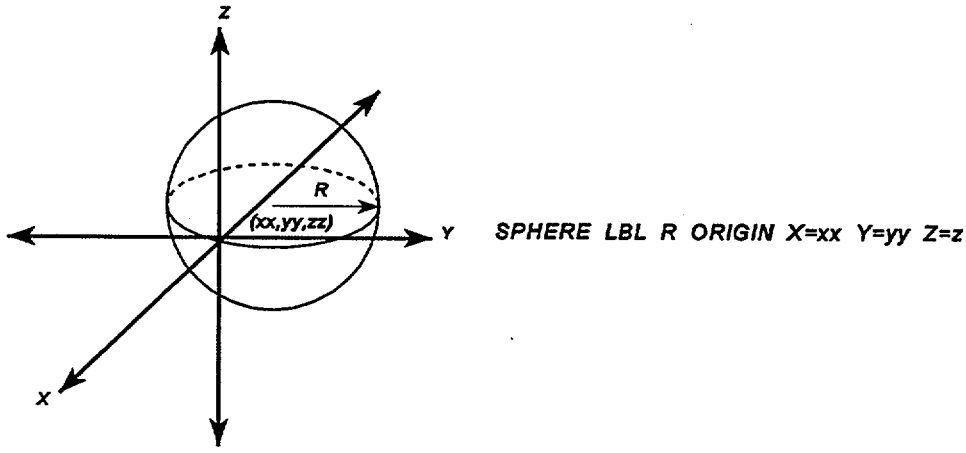
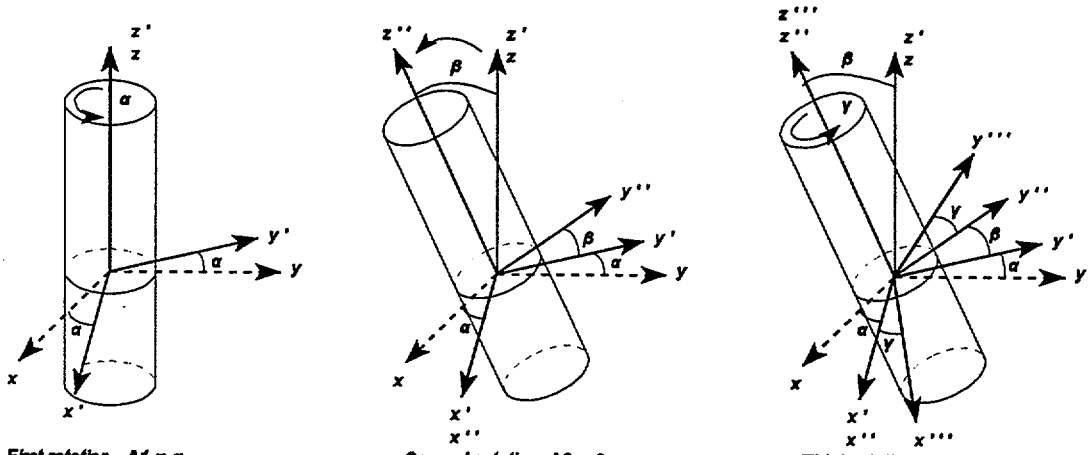
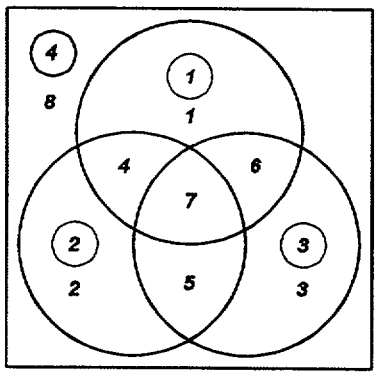


Figure F17.4.20 Example of the ORIGIN option



CYLINDER LBL R Zt Zb ROTATE A1=α A2=β A3=γ

Figure F17.4.21 Explanation of the ROTATE option




```
sphere 10 3 origin y=2
sphere 20 3 origin x=-2 y=-2
sphere 30 3 origin x=2  y=-2
cubold 40 6 -6 6 -6 6 -6
media 1 1 10 -20 -30
media 2 1 20 -10 -30
media 3 1 30 -10 -20
media 4 1 10  20 -30
media 5 1 20  30 -10
media 6 1 30  10 -20
media 7 1 10  20  30
media 8 1 40 -10 -20 -30
boundary 40
```

Figure F17.4.22 Example of the MEDIA record

HOLE is used to position a unit within a region in a surrounding unit relative to the geometric shapes in surrounding unit. The keyword, *HOLE*, is followed by the *HOLE ID mat*, and the *REGION DEFINITION VECTOR, i1, i2, ... , in* with one or more blanks separating the data. The unit being placed in the hole can be rotated and translated using the *APPROPRIATE MODIFICATION INFORMATION*. The boundary dimensions of the unit being placed in the hole must precisely match the boundary of the region containing the hole. The unit being placed in the hole is aligned in the region using ROTATE and ORIGIN. If a hole is nested in either an array position or another hole the boundary of the nested hole may touch but must not cross the boundary of the unit in the array of the other hole.

ARRAY is used to position an array within a region in a surrounding unit relative to the geometric shapes in the surrounding unit. The keyword is followed by one or more blanks and the *ARRAY ID mat* and the *REGION DEFINITION VECTOR, i1 i2 ... in*. The *REGION DEFINITION VECTOR* defines the boundary of the array. The subordinate keyword PLACE is then entered, followed by six numbers that precisely locate the array within the surrounding unit. The first three numbers consist of the location in the array of the unit selected to position the array. The next three numbers consist of the position in the surrounding unit that contains the origin of the selected unit.

The *MIXTURE, HOLE,* or *ARRAY ID mat* specifies the mixture, hole, or array that is to occupy the volume defined by the region if the keyword is MEDIA. This information is followed by one or more blanks.

The *BIAS ID imp* specifies the weights that are to be used in the volume defined by each media record. This is used only if the *MEDIA* keyword is specified. 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. F17.4.7. The *BIAS ID* number is followed by one or more blanks.

REGION DEFINITION VECTOR i1 i2 ... is a series of geometry record numbers used to define the location of a region that contains a material mixture, or array. A positive value indicates the region is located inside the space defined by the geometry record. A negative value indicates the region is located outside the space defined by the geometry record. As many geometry record numbers as are required to define the volume occupied by the region must be entered. Each media card represents a distinct region of the unit. Each region can only be represented by one media card. Care must be taken when regions share boundaries to specify which region contains the boundary.

The *BOUNDARY KEYWORD*, **BOUNDARY** is followed by one or more blanks and the region definition vector that defines the boundary of the unit. Each unit must have one and only one boundary record. The boundary record describes the volume contained within the unit. Only that section of the geometry contained within the volume specified by the unit boundary record is part of the unit.

(c) *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 nonblank 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 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.

Additional EXAMPLES of geometry input are given below:

1. Initiate input data for unit No. 6.

   UNIT 6

2. Initiate input data for the global unit which is unit No. 4.

   GLOBAL UNIT 4

3. Specify a hemisphere labeled 10 with a radius of 5.0 cm which contains only material in the sphere with z > 2.0 with the sphere centered at the origin, and its origin translated to x=1.0, y=1.5, and z=3.0.

   SPHERE 10 5.0 CHORD +Z=2.0 ORIGIN x=1.0 y=1.5 z=3.0

4. Specify a hemicylinder labeled 10 having a radius of 5.0 cm and a length extending from z=2.0 cm to z=7.0 cm. The hemisphere has been truncated parallel to the z axis at x=-3 such that material 1 does not exist between x=-3 and x=-5. Position the origin of the truncated hemicylinder at x=10 cm and y=15 cm with respect to the origin of the unit and rotate it so it is in the YZ plane centered at x=10 and at a 45° angle with the y plane.

   CYLINDER 10 5.0 7.0 2.0 CHORD +x=-3.0 ORIGIN x=10.0 y=15.0 ROTATE A2=-45

5. Position array 6 do that the origin of the unit in position 1, 2, 3 is located at position x=2, y=3, z=4 relative to the origin of the surrounding unit. Specifying the array boundary to be the region that is inside the geometry records labeled 1 and 2 and outside the geometry record labeled 3 used to describe the surrounding unit.

   ARRAY 6 1 2 -3 PLACE 1 2 3 2.0 3.0 4.0

6. Place unit 2 in the surrounding unit such that the origin unit 2 is at x=3, y=3.5, z=4 relative to the origin of the surrounding unit. A boundary based on the unit 2 boundary is automatically added to the set of equations for unit containing the hole.

   HOLE 2 ORIGIN x=3.0 y=3.5 z=4.0

F17.4.5 ARRAY DATA

ARRAY DEFINITION data . . . The array definition data block is used to define the size and type of an array and to position units (defined in the geometry data) in a 3-D lattice that represents the array being described. As many arrays as are necessary can be described in a problem, subject to computer storage limitations. Any multiple arrays can be placed in any unit either directly or indirectly using holes. There is no default global array. If a global array is desired it must be explicitly defined.

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 array type, (3) the number of units in each direction of the 3-D lattice, (4) the global array number, and (5) a comment to be printed at the beginning of the array in the printout. The array identification number (1), the number of units in each direction (3), and the associated *UNIT ORIENTATION DESCRIPTION* must be entered for each array that is used in the problem. The array type (3) is entered to specify if the array is cuboidal or hexagonal. If not entered the array is assumed to be cuboidal. The global array number (4), should be entered only if desired. A global array number is required to accumulate certain neutron physics information such as certain matrix information. If the global array number 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 3-D lattice and the *DELIMITER* is used to terminate the unit orientation data for the array. *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**

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, no default
GBL= *globl*	array number for the global array (if entered more than once the last value entered becomes the global array), no default.
TYP = *atyp*	type of array (square or hexagonal), default = square
NUX = *nbxmax*	number of units in the X direction of the array, no default
NUY = *nbymax*	number of units in the Y direction of the array, no default

NUZ = *nbzmax* number of units in the Z direction of the array, no default

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 units may be considered as consisting of a 3-D matrix of unit numbers, with the unit position increasing in the positive X, Y, and Z directions, respectively. Each set of mixed unit orientation data for the **LOOP** option consists of the following parameters, separated by one or more blanks.

ORIENTATION DATA for **LOOP**

ltype The unit number, *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*=, see ARRAY PARAMETERS).

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*=, see ARRAY PARAMETERS).

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 units 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*=, see ARRAY PARAMETERS).

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 units 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 relevant unit number and superimpose the other unit numbers in their proper places. An example showing the use of the LOOP option is given below. This 5 × 4 × 3 array of units is a matrix of units that has 5 units stacked in the X direction, 4 units in the Y direction, and 3 units in the Z direction. X increases from left to right, and Y increases from bottom to top. Each Z layer is shown separately.

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 Z 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 Z 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 Z layer.

(5) 3 2 4 1 2 3 1 3 3 1 This loads the 3's in the third Z layer and completes the array data input.

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 Table F17.4.2. 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 F17.4.2 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	$	*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 the end points, j and k, with i entries linearly interpolated between them (i.e., a total of i+2 points). At least one blank must separate j and k. When used for an integer array, the I option should only be used to generate integer steps [i.e., (k-j)/(i+1) should be a whole number]
i	L	*j k*	Provides the end points, j and k, with i entries logarithmically interpolated between them (i.e., a total of i+2 points). At least one blank must separate j and k
	T		Terminates the data reading for the array

NOTE: 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 F17.4.2, consider the following examples. The positions in an array are numbered sequentially from left to right, bottom to top. A 3 × 3 × 1 array has 9 positions and is numbered as shown below. X increases from left to right and Y increases from bottom to top.

7 8 9

4 5 6

1 2 3

EXAMPLE 1. Consider a 3 × 3 × 1 array filled with 8 Unit 1's and a Unit 2 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 4S skips over the next four positions in the array, and the 2 loads a 2 in the next (fifth) 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 4 × 3 × 1 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. The T terminates the data.

EXAMPLE 3. Consider a 6 × 3 × 1 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 the following:

(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 7 × 3 × 1 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.

F17.4.6 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.** Different albedo options are allowed for different global boundaries. If the global unit boundary record definition vector contains a single geometry record label that references a cuboid, any available albedo option is allowed on any surface. However, if a periodic boundary condition is to be used, it must be specified on opposing faces simultaneously. If the global unit boundary record definition vector contains a single geometry record label that references a geometry shaped composed of paired planes, i.e., hexprism, dodecahedron, parallelpiped, rhomboid, etc., any albedo type is allowed but the same type must be on all surfaces. If the global unit boundary record definition vector contains a single geometry record label that references any geometry shape, only the void, mirror, or white albedo condition is allowed and the same condition must be on all surfaces. If the global unit boundary record definition vector contains more than one geometry record label only the void albedo condition is allowed. If no albedo is specified a void albedo is assumed. The white albedo returns an outgoing particle at its point of departure with a random angle.

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 F17.4.3, and the *ALBEDO NAMES* are given in Table F17.4.4.

Table F17.4.3 Face codes for entering boundary (albedo) conditions

Face code	Faces defined by 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
&YZ=	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 F17.4.4 Albedo names available on the KENO-VI albedo library for use with the face codes

DP0H2O DPOH2O DP0 DPO	30.48-cm (12 in.) double P_o water differential albedo with 4 incident angles
H2O WATER	30.48-cm (12 in.) water differential albedo with 4 incident angles
PARAFFIN PARA WAX	30.48-cm (12 in.) paraffin differential albedo with 4 incident angles
CARBON GRAPHITE C	200.00-cm (78.74-in.) carbon differential albedo with 4 incident angles
ETHYLENE POLY CH2	30.48-cm (12-in.) polyethylene differential albedo with 4 incident angles
CONC-4 CON4 CONC4	10.16-cm (4-in.) concrete differential albedo with 4 incident angles
CONC-8 CON8 CONC8	20.32-cm (8-in.) concrete differential albedo with 4 incident angles
CONC-12 CON12 CONC12	30.48-cm (12-in.) concrete differential albedo with 4 incident angles
CONC-16 CON16 CONC16	40.64-cm (16-in.) concrete differential albedo with 4 incident angles
CONC-24 CON24 CONC24	60.96-cm (24-in.) 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
WHITE	White boundary condition

Example: Use a 24-in. concrete albedo boundary condition on the -Z face of a problem with a cuboidal boundary and use mirror image reflection on the +X and -X faces of the cuboid to represent an infinite linear array on a 2-ft-thick concrete pad.
READ BOUNDS -ZB=CON24 XFC=MIRROR END BOUNDS

F17.4.7 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, Sect. F17.4.3) for all *BIAS IDs* 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-VI 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.a weighting library or from the *AUXILIARY DATA* input. The materials available from the KENO V.a weighting library are listed in Table F17.4.5.

BIASING INFORMATION is entered in the following form:

READ BIAS KEYWORD CORRELATION DATA AUXILIARY DATA **END BIAS**

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 chosen from the existing KENO V.a weighting library (Table F17.4.5) or from the *AUXILIARY DATA* as described later. If a material ID appears in both the KENO V.a weighting 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*.

Table F17.4.5 IDs, group structure and incremental thicknesses for weighting data available on the KENO V.a weighting library

Material	Material ID	Group structure for which weights are available	Increment[a] thickness (cm)	Total number of increments available
Concrete	301	16	5	20
		27	5	20
		44	5	20
		218	5	20
		238	5	20
Paraffin	400	16	3	10
		27	3	10
		44	3	10
		218	3	10
		238	3	10
Water	500	16	3	10
		27	3	10
		44	3	10
		218	3	10
		238	3	10
Graphite	6100	16	20	10
		27	20	10
		44	20	10
		218	20	10
		238	20	10

[a]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 F17.4.4, the next *ngp* weights apply to the next increment of that thickness, etc.].

CAUTION--If bias IDs 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 overbiasing or underbiasing.

iend is the BIAS ID of the group-dependent *wtavg's* from the (*iend* – *ibgn* + 1)th interval of material *id*.

GENERIC EXAMPLE: READ BIAS ID=mm ibgn iend END BIAS where mm is a material ID from Table F17.4.5, *ibgn* is the bias ID associated with the 1st interval of material mm, and *iend* is the bias ID associated with the (*iend* – *ibgn* + 1) interval of material mm.

SPECIFIC EXAMPLE: Use CORRELATION DATA to utilize the water biasing factors in BIAS IDs 2 through 11.

READ BIAS ID=500 2 11 END BIAS

AUXILIARY DATA are used to enter user-supplied biasing or weighting information. It can be used to supply biasing information for materials not found in the KENO V.a weighting library or to override the *wtavgs* from that library. When AUXILIARY DATA are entered, CORRELATION DATA must also be entered in order to use the data. *AUXILIARY DATA* must be entered in the order shown.

wttitl enter an arbitrary title name (12 characters maximum), such as CONCRETE, WATER, SPECIALH2O, etc., to identify the material for which you are entering data. Embedded blanks are not allowed.

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.

GENERIC EXAMPLE of AUXILIARY DATA: READ BIAS WT=wttitl id nsets thkinc numinc ngpwt wtavg END BIAS

SPECIFIC EXAMPLE: Enter AUXILIARY DATA to specify biasing factors for SPECIALWATER to be used in BIAS IDs 6 and 7. The SPECIALWATER biasing factors have a value of 0.69 for BIAS ID 6 and 0.86 for BIAS ID 7 in each energy group. Sixteen-group cross sections are being used. Each weighting region is 3.048 cm thick. The MATERIAL ID is arbitrarily chosen to be 510. Note that CORRELATION DATA must be entered to allow the AUXILIARY DATA to be used for BIAS IDs 6 and 7.

```
READ BIAS  WT=SPECIALWATER  510  1  3.048  2  16
16*0.69  16*0.86  ID=510  6 7  END BIAS
```

WARNING: The user should thoroughly understand weighted tracking before attempting to generate and use auxiliary data for biasing. Incorrect weighting can cause the code to produce incorrect results without obvious symptoms.

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. When auxiliary data are entered, correlation data must also be entered in order to use the auxiliary data.

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 K6-125 and K6-128 may be printed.

An example of multiple definitions for the same bias ID follows:

```
READ BIAS ID=400 2 7 ID=500 5 7 END BIAS .
```

The data for paraffin (ID=400) will be used for bias IDs 2, 3, and 4, and the data for water (ID=500) will be used for bias IDs 5, 6, and 7. The paraffin data for bias IDs 5, 6, and 7 have been overwritten by water data.

Multiple definitions for the same bias ID are not necessarily incorrect. However, the user should be cautious about doing it and ensure that the desired biasing or weighting functions are utilized in the desired geometry regions.

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.a 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, and *AUXILIARY DATA* will not be 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

The biasing data would be: READ BIAS ID=400 6 9 END BIAS.

The results of these data are the following:

(1) The group-dependent *wtavg* for the 0- to 3-cm interval of paraffin will be used in the volume defined by the geometry region 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 region having *imp*= 7.

(3) The group-dependent *wtavg* for the 6- to 9-cm interval of paraffin will be used in the volume defined by the geometry region having *imp*= 8.

(4) The group-dependent *wtavg* for the 9- to 12-cm interval of paraffin will be used in the volume defined by the geometry region having *imp*= 9.

F17.4.8 START DATA

START DATA . . . Special start options are available for controlling the initial neutron distribution. The default starting distribution is flat over the volume specified by the unrotated, untranslated geometry record specified in the first position of the global unit boundary record in fissile material only. See Table F17.4.6 for the starting distributions available in KENO-VI.

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
Table F17.4.6 lists the available options under the heading, "Start type."

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 F17.4.6 Starting distributions available in KENO-VI

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 unrotated, untranslated geometry record in the first position of the global unit boundary record, (2) the boundary of the global array having the reflector key set false, or (3) 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[a]	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

[a] 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 previous value of LNU. 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, Sect. F17.4.3).

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 fissionable mixture having the smallest mixture number. 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. Each successive *lfin* should be greater than the previous one. Use for start type 6.

NBX = *nboxst* the unit 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 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 cuboid, XSM must be entered in the start 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 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 cuboid, XSP must be entered in the start 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 cuboid. If RFL is YES and the outer region of the reflector is not a cuboid, YSM must be entered in the start 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 cuboid. If RFL is YES and the outer region of the reflector is not a cuboid, YSP must be entered in the start 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, XSM 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 cuboid. If RFL is YES and the outer region of the reflector is not a cuboid, ZSM must be entered in the start 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 cuboid. If RFL is YES and the outer region of the reflector is not a cuboid, ZSP must be entered in the start 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 = YES.
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 are printed. If it is NO, start type 6 data are 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

F17.4.9 EXTRA 1-D XSECS IDS DATA

EXTRA 1-D CROSS-SECTION IDS . . . Extra 1-D cross-section IDs 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 IDS* **END X1DS**

EXTRA 1-D CROSS SECTION IDS

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).

F17.4.10 MIXING TABLE DATA

CROSS-SECTION MIXING TABLE . . . A cross-section mixing table must be entered if KENO-VI is being run "stand alone" and an ICE mixed cross-section MORSE/KENO V.a format library is not being used. If the parameter LIB=, Sect. F17.4.3, is entered, mixing table data must be entered. 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. F17.5.4.3.

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_o 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. F17.5.4.4.

EPS = *pbxs* where *pbxs* is the value of the P_o 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 IDs and number densities used in the mixtures. It consists of (a) a *MIXTURE ID* and a set of (b) *NUCLIDE IDs*, and (c) *NUMBER DENSITIES* (atoms/b-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-format cross-section library

(c) *NUMBER DENSITY dens* enter the number density (atoms/b-cm) 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 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 used as 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.

F17.4.11 PLOT DATA

PLOT DATA. . . Plots of slices specified through the geometry can be generated and displayed (1) as character plots using alphanumeric characters to represent mixture numbers, unit numbers or bias ID numbers or (2) as color plots which generate a GIF file using colors to represent mixture numbers, unit numbers or bias ID numbers. Color plots require an independent program to display the GIF file to a PC or workstation monitor or to convert the file to be displayed using a plotting device. The keyword **SCR=** is used to control the plot display method. **SCR=YES**, the default value, utilizes the color plot display method. **SCR=NO** utilizes the character plot display method. The value of **SCR** determines the plot display method for all the plots specified in a problem. If **SCR=** is entered more than once, the last entry determines the plot display method. In other words, all plots generated by a problem will be either character plots or color plots. The first LPI entered is used until it is changed. Therefore, the desired LPI should be entered before the end of the first set of plot data.

The *PLOT DATA* can include the data for any or all types of plot. A plot by mixture number is the default. The kind of plot is defined by the parameter **PIC=.** Character 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 plotting of the picture can be suppressed by entering **PLT= NO** in the parameter data or picture data. This allows plot data to be kept in the problem input for reference purposes without actually plotting the picture(s). Entering a value for **PLT** in the plot 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. The upper-left and lower-right coordinates of the plot must be specified relative to the origin of the problem. See Sect. F17.5.9 for a discussion of plot origins and character plot data.

Enter **PLOT DATA** in the form:

READ PLOT *PICTURE PARAMETERS* **END PLOT**

PLOT 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. Multiple sets of plot data can be entered. The parameter input for each plot is terminated by a labeled or unlabeled **END**. The labeled **END** cannot use the word **PLOT** as the first four characters of the label. For example, **END PLT1** is a valid label, but **END PLOT1** is not. If an unlabeled **END** is used, it cannot start in column 1.

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-VI case

PIC = *wrd* The plot type, *wrd*, is followed by one or more blanks and must be one of the keywords listed below. The plot type is initialized to MAT; the default is the value from the previous plot.

MAT MIX MIXT MIXTURE MEDI MEDIA	These keywords will cause the plot to represent the mixture numbers used in the specified geometry slice.
UNT UNIT UNITTYPE	These keywords will cause the plot to represent the units used in the specified geometry slice. In the legend of the color plot, the material number actually refers to the units.
IMP BIAS BIASID ID WTS WEIG WEIGHTS WGT WGTS	These keywords will cause the plot to represent the bias numbers used in the specified geometry slice. In the legend of the color plot, the material number actually refers to the bias ID number.
Plot coordinates	Enter values for the upper-left and lower-right coordinates of the plot as described below. **Data must be entered for all nonzero coordinates unless all six values from the previous plot are to be used.**
Upper left coordinates	Enter the X, Y, and Z coordinates of the upper-left-hand corner of the plot.
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.
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 plot

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 plot. 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 plot

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 plot. 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.

Scaling parameters

Enter one or more scaling parameters to define the size of the plot. NOTE: If any of the scaling parameters are entered for a plot, the value of those that were not entered are recalculated. If none of the scaling parameters are specified for a plot, the values from the previous plot are used.

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 plot.
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 plot.
Default = value from previous plot; initialized to zero if DLX or DLD is entered.

Global scaling parameter

LPI= *lpi*

is a scaling factor used to control the horizontal to vertical proportionality of a plot or plots. SCALE 4.3 and later versions allow **LPI** to be input as a floating point number. For an undistorted character plot, **LPI** should be specified as the number of characters down the page that occupy the same distance as ten characters across the page. For an undistorted color plot, **LPI** should be entered as ten times the ratio of the vertical pixel dimension to the horizontal pixel dimension. The default value of **LPI** is 8.0 for a character plot and 10.0 for a color plot. **LPI=10** will usually display an undistorted color plot.

The value entered for **LPI** applies to all plot data following it until a new value of **LPI** is specified.

NOTE: Plot data must include the specification of the upper left corner of the plot and the direction cosines across and down the plot.

Additional data required to generate a plot are:

1. the lower right corner of the plot, the global scaling parameter, **LPI**, and one of the scaling parameters (**DLX, DLD, NAX, NDN**).

2. the lower right corner of the plot, one of the scaling parameters related to the horizontal specifications of the plot (**DLX** or **NAX**), and one of the scaling

parameters related to the vertical specification of the plot (**DLD** or **NDN**). **LPI**, even if specified will not be used.

3. **NAX** and **NDN** and any two of **LPI, DLX,** and **DLD**. If **LPI, DLX,** and **DLD** are all specified, **LPI** is not used.

The data required to generate a plot may be supplied from (1) defaulted values, (2) data from the previous plot, or (3) data that are specifically entered for the current plot.

Miscellaneous parameters — Enter miscellaneous parameters

RUN= *wrd* — 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.

PLT= *wrd* — 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.

NOTE: The parameters RUN and PLT can also be entered in the PARAMETER data. See Sect. F17.4.3. It is recommended that these parameters be entered only in the parameter data in order to ensure that the data printed in the "Logical Parameters" table is what is actually performed. If RUN and/or PLT are entered in both the parameter data and plot data, the results vary, depending on whether the problem is run (1) stand-alone, (2) as a restarted problem, (3) as CSAS with parm=check, or (4) as CSAS without parm=check. These conditions are detailed below.

KENO VI stand-alone and CSAS with PARM=CHECK — The values of RUN and/or PLT entered in the **KENO** parameter data are printed in the "Logical Parameters" table of the problem output. However, values for RUN and/or PLT entered in the **KENO plot data** will override the values entered in the parameter data.

Restarted KENO VI — The values of RUN and/or PLT printed in the "Logical Parameters" table of the problem output are the final values from the "parent" problem unless those values are overridden by values entered in the **KENO parameter data** of the restarted problem. If the problem is restarted at generation 1, **KENO plot data** can be entered and the values for RUN and/or PLT will override the values printed in the "Logical Parameters" table.

CSAS without PARM=CHECK The values of RUN and/or PLT entered in the KENO parameter data override values entered in the KENO plot data. The values printed in the "Logical Parameters" table control whether the problem is to be executed and whether a plot is performed.

SCR= *wrd* The plot display method is specified by entering either YES or NO for *wrd*. The default value is YES. **SCR=YES** utilizes the color plot display method. **SCR=NO** utilizes the character plot display method. If SCR is entered more than once in a problem, the last value entered is the one that is used.

NCH= *delim CHAR delim* Enter only if plots are to be made utilizing the character plot display method (**SCR=NO**). 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, unit number, or bias ID. 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. For example, assume **PIC=MAT** is specified and 15 mixtures are defined in the mixing table and the geometry data uses only mixtures 3 and 7. By default, a blank will be printed for mixture zero, a 1 will be printed for mixture 3, and a 2 will be printed for mixture 7. If you wish to print a zero for a void (mixture 0), a 3 for mixture 3, and a 7 for mixture 7, enter NCH='037.' See Sect. F17.5.9 for examples.

The default values of *CHAR* are the following:

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	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
SYMBOL	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
MEDIA	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	
SYMBOL	W	X	Y	Z	#	,	$	-	+	)	|	&	>	:	;	
MEDIA	47	48	49	50	51	52	53	54	55	56	57	58				
SYMBOL	.	-	%	*	"	=	!	(	@	<	/	0				

CLR= *num(i) red(num(i)) green(num(i)) blue(num(i))* **END COLOR**

enter only if plots are to be made utilizing the color plot display method (**SCR=YES**). After entering the keyword **CLR=**, 4 numbers are entered *i* times. The first number, *num*(i), represents a media (mixture) number, unit number, or bias ID. The next three numbers, whose values can range from 0 through 255, define the *red, green, and blue* components of the color that will represent this *num(i)* in the plot. The sequence *num(i) red(num(i)) green(num(i)) blue(num(i))* is repeated until the colors associated with all of the media (mixture) numbers, unit numbers, or bias IDs used in the problem have been defined. The smallest number that can be entered for *num(i)* is −1, representing undefined regions in the plot. A *num(i)* of 0 represents void regions, *num(i)* of 1 represents the smallest media, unit, or bias ID used in the problem, *num(i)* of 2 represents the next larger media, unit, or bias ID used in the problem, etc. The color plot definition data is terminated by entering the keywords **END COLOR**. A total of 256 default colors are provided in Table F17.4.7. Two of those colors represent undefined regions, *num*(-1), as black; and void regions, *num*(0), as gray. The remaining 254 colors represent the default values for mixtures, bias IDs, or unit numbers used in the problem. If ***num*** is entered as **-1**, the next three numbers define the color that will be used to represent undefined regions of the plot. The default color for undefined regions is black, represented as 0 0 0. If ***num*** is entered as **0**, the next three numbers define the color that will represent void regions in the plot. The default color for void is gray, represented as 200 200 200. For example, assume a color plot is to be made for a problem that utilizes void regions and mixture numbers 1, 3, and 5. By default, the undefined regions (Index −1) will be black; void regions (Index 0) will be gray; the first mixture, mixture 1 (Index 1), will be medium blue; the next larger mixture, mixture 3 (Index 2), will be turquoise2; and the last mixture, mixture 5 (Index 3), will be green2. If these values are acceptable, data need not be entered for CLR=. If the user decides to define void to be white (255 255 255), mixture 1 to be red (255 0 0), mixture 3 to be bright blue (0 0 255), and mixture 5 to be green (0 255 0), the following data could be entered:

CLR=0 255 255 255 1 255 0 0 2 0 0 255 3 0 255 0 END COLOR

In this example, the first number (0) defines the void and the next three numbers are the red, green, and blue components that combine as the color white. The fifth number (1) represents the smallest mixture number (mixture 1) and the next three numbers are the red, green, and blue components of red. The ninth number (2) represents the next larger mixture number (mixture 3), and the next three numbers are the red, green, and blue components of bright blue. The thirteenth number (3) represents the next larger mixture number (mixture 5), and the next three numbers are the red, green, and blue components of green. The END COLOR terminates the color definition data. Because color data were not entered for num(i) of −1, undefined regions will be represented by the color black, the default specification from Table F17.4.7. The red, green, and blue components of some bright colors are listed below.

Display Color	red	green	blue
black	0	0	0
white	255	255	255
"default void gray"	200	200	200
red	255	0	0
green	0	255	0
brightest blue	0	0	255
yellow	255	255	0
brightest cyan	0	255	255
magenta	255	0	255

The 256 default colors are listed in Table F17.4.7.

Table F17.4.7 Default color specifications for the color plot display method

-1	0	0	0	black gray0 grey0
0	200	200	200	*default
1	0	0	205	medium MediumBlue blue3
2	0	229	238	turquoise
3	0	238	0	green2
4	205	205	0	yellow3
5	238	0	0	red2
6	145	44	238	purple2
7	150	150	150	gray59 grey59
8	240	200	220	white gray100 grey100
9	0	191	255	deep DeepSkyBlue DeepSkyBlue1
10	224	255	255	light LightCyan LightCyan1
11	0	255	127	spring SpringGreen SpringGreen1
12	255	255	224	light LightYellow LightYellow1
13	255	0	0	red red1
14	255	0	255	magenta magenta1
15	67	110	238	RoyalBlue2
16	174	238	238	PaleTurquoise2
17	180	238	180	DarkSeaGreen2
18	238	220	130	LightGoldenrod2
19	238	99	99	IndianRed2
20	238	122	233	orchid2
21	25	25	112	midnight MidnightBlue
22	0	0	128	navy navy NavyBlue
23	100	149	237	cornflower CornflowerBlue
24	72	61	139	dark DarkSlateBlue
25	106	90	205	slate SlateBlue
26	123	104	238	medium MediumSlateBlue
27	30	144	255	dodger DodgerBlue DodgerBlue1
28	135	206	235	sky SkyBlue
29	135	206	250	light LightSkyBlue
30	70	130	180	steel SteelBlue
31	176	196	222	light LightSteelBlue
32	176	224	230	powder PowderBlue
33	0	206	209	dark DarkTurquoise
34	72	209	204	medium MediumTurquoise
35	95	158	160	cadet CadetBlue
36	102	205	170	medium MediumAquamarine aquamarine3
37	127	255	212	aquamarine aquamarine1
38	0	100	0	dark DarkGreen
39	85	107	47	dark DarkOliveGreen
40	143	188	143	dark DarkSeaGreen
41	60	179	113	medium MediumSeaGreen
42	32	178	170	light LightSeaGreen
43	152	251	152	pale PaleGreen
44	0	255	0	green green1
45	127	255	0	chartreuse chartreuse1
46	0	250	154	medium MediumSpringGreen
47	173	255	47	green GreenYellow
48	50	205	50	lime LimeGreen
49	154	205	50	yellow YellowGreen OliveDrab3
50	34	139	34	forest ForestGreen
51	107	142	35	olive OliveDrab
52	189	183	107	dark DarkKhaki
53	240	230	140	khaki
54	238	232	170	pale PaleGoldenrod
55	250	250	210	light LightGoldenrodYellow
56	255	255	224	light LightYellow LightYellow1
57	255	255	0	yellow yellow1
58	255	215	0	gold gold1
59	238	221	130	light LightGoldenrod
60	184	134	11	dark DarkGoldenrod
61	188	143	143	rosy RosyBrown
62	205	92	92	indian IndianRed
63	139	69	19	saddle SaddleBrown chocolate4
64	160	82	45	sienna
65	205	133	63	peru tan3
66	222	184	135	burlywood
67	245	245	220	beige
68	245	222	179	wheat
69	244	164	96	sandy SandyBrown
70	210	105	30	chocolate
71	178	34	34	firebrick
72	165	42	42	brown
73	233	150	122	dark DarkSalmon
74	250	128	114	salmon
75	255	160	122	light LightSalmon LightSalmon1
76	255	165	0	orange orange1
77	255	140	0	dark DarkOrange

Table F17.4.7 (continued)

78	255	127	80	coral
9	240	128	128	light
				LightCoral
80	255	99	71	tomato
				tomato1
81	255	69	0	orange
				OrangeRed
				OrangeRed1
82	255	0	0	red
				red1
83	255	105	180	hot
				HotPink
84	255	20	147	deep
				DeepPink
				DeepPink1
85	255	192	203	pink
86	255	182	193	light
				LightPink
87	219	112	147	pale
				PaleVioletRed
88	176	48	96	maroon
89	199	21	133	medium
				MediumVioletRed
90	208	32	144	violet
				VioletRed
91	238	130	238	violet
92	221	160	221	plum
93	218	112	214	orchid
94	153	50	204	dark
				DarkOrchid
95	148	0	211	dark
				DarkViolet
96	186	85	211	medium
				MediumOrchid
97	138	43	226	blue
				BlueViolet
98	160	32	240	purple
99	147	112	219	medium
				MediumPurple
100	216	191	216	thistle
101	238	233	233	snow2
102	238	229	222	seashell2
103	238	223	204	AntiqueWhite2
104	238	213	183	bisque2
105	238	203	173	PeachPuff2
106	238	207	161	NavajoWhite2
107	238	233	191	LemonChiffon2
108	238	232	205	cornsilk2
109	238	238	224	ivory2
110	224	238	224	honeydew2
111	238	224	229	LavenderBlush2
112	238	213	210	MistyRose2
113	224	238	238	azure2
114	122	103	238	SlateBlue2
115	67	110	238	RoyalBlue2
116	0	0	238	blue2
117	28	134	238	DodgerBlue2
118	92	172	238	SteelBlue2
119	0	178	238	DeepSkyBlue2
120	126	192	238	SkyBlue2
121	164	211	238	LightSkyBlue2
122	185	211	238	SlateGray2
123	188	210	238	LightSteelBlue2
124	178	223	238	LightBlue2
125	209	238	238	LightCyan2
126	174	238	238	PaleTurquoise2
127	142	229	238	CadetBlue2
128	0	238	238	cyan2
129	141	238	238	DarkSlateGray2
130	118	238	198	aquamarine2
131	180	238	180	DarkSeaGreen2
132	78	238	148	SeaGreen2
133	144	238	144	light
				lightGreen
				PaleGreen2
134	0	238	118	SpringGreen2
135	0	238	0	green2
136	118	238	0	chartreuse2
137	179	238	58	OliveDrab2
138	188	238	104	DarkOliveGreen2
139	238	230	133	khaki2
140	238	220	130	LightGoldenrod2
141	238	238	209	LightYellow2
142	238	238	0	yellow2
143	238	201	0	gold2
144	238	180	34	goldenrod2
145	238	173	14	DarkGoldenrod2
146	238	180	180	RosyBrown2
147	238	99	99	IndianRed2
148	238	121	66	sienna2
149	238	197	145	burlywood2
150	238	216	174	wheat2
151	238	154	73	tan2
152	238	118	33	chocolate2
153	238	44	44	firebrick2
154	238	59	59	brown2
155	238	130	98	salmon2
156	238	149	114	LightSalmon2
157	238	154	0	orange2
158	238	118	0	DarkOrange2
159	238	106	80	coral2
160	238	92	66	tomato2
161	238	64	0	OrangeRed2
162	238	0	0	red2
163	238	18	137	DeepPink2
164	238	106	167	HotPink2
165	238	169	184	pink2
166	238	162	173	LightPink2
167	238	121	159	PaleVioletRed2
168	238	48	167	maroon2
169	238	58	140	VioletRed2
170	238	0	238	magenta2
171	238	122	233	orchid2
172	238	174	238	plum2
173	209	95	238	MediumOrchid2
174	178	58	238	DarkOrchid2
175	145	44	238	purple2
176	159	121	238	MediumPurple2
177	238	210	238	thistle2
178	255	250	250	snow
				snow1
179	139	137	137	snow4
180	255	245	238	seashell
				seashell1
181	255	228	196	bisque
				bisque1
182	255	218	185	peach
				PeachPuff
				PeachPuff1
183	255	250	205	lemon
				LemonChiffon
				LemonChiffon1
184	255	248	220	cornsilk
				cornsilk1
185	255	255	240	ivory
				ivory1
186	240	255	240	honeydew
				honeydew1
187	255	240	245	lavender
				LavenderBlush
				LavenderBlush1
188	255	228	225	misty
				MistyRose
				MistyRose1
189	240	255	255	azure
				azure1
190	131	111	255	SlateBlue1
191	72	118	255	RoyalBlue1

Table F17.4.7 (continued)

192	30	144	255	dodger	225	255	127	36	chocolate1
				DodgerBlue	226	255	48	48	firebrick1
				DodgerBlue1	227	255	64	64	brown1
193	99	184	255	SteelBlue1	228	255	140	105	salmon1
194	0	191	255	deep	229	255	160	122	lightsalmon
				DeepSkyBlue					LightSalmon
				DeepSkyBlue1					LightSalmon1
195	135	206	255	SkyBlue1	230	255	165	0	orange
196	176	226	255	LightSkyBlue1					orange1
197	198	226	255	SlateGray1	231	255	127	0	DarkOrange1
198	202	225	255	LightSteelBlue1	232	255	114	86	coral1
199	191	239	255	LightBlue1	233	255	99	71	tomato
200	224	255	255	light					tomato1
				LightCyan	234	255	69	0	orangered
				LightCyan1					OrangeRed
201	187	255	255	PaleTurquoise1					OrangeRed1
202	152	245	255	CadetBlue1	235	255	20	147	deep
203	0	245	255	turquoise1					DeepPink
204	151	255	255	DarkSlateGray1					DeepPink1
205	127	255	212	aquamarine	236	255	110	180	HotPink1
				aquamarine1	237	255	181	197	pink1
206	193	255	193	DarkSeaGreen1	238	255	174	185	LightPink1
207	84	255	159	SeaGreen1	239	255	130	171	PaleVioletRed1
208	154	255	154	PaleGreen1	240	255	52	179	maroon1
209	0	255	127	spring	241	255	62	150	VioletRed1
				SpringGreen	242	255	131	250	orchid1
				SpringGreen1	243	255	187	255	plum1
210	127	255	0	chartreuse	244	224	102	255	MediumOrchid1
				chartreuse1	245	191	62	255	DarkOrchid1
211	192	255	62	OliveDrab1	246	155	48	255	purple1
212	202	255	112	DarkOliveGreen1	247	171	130	255	MediumPurple1
213	255	246	143	khaki1	248	255	225	255	thistle1
214	255	236	139	LightGoldenrod1	249	139	0	139	dark
215	255	255	224	light					darkMagenta
				LightYellow					magenta4
				LightYellow1	250	139	0	0	dark
216	255	215	0	gold					darkRed
				gold1					red4
217	255	193	37	goldenrod1	251	0	139	0	green4
218	255	185	15	DarkGoldenrod1	252	0	0	139	dark
219	255	193	193	RosyBrown1					darkBlue
220	255	106	106	IndianRed1					blue4
221	255	130	71	sienna1	253	0	139	139	dark
222	255	211	155	burlywood1					darkCyan
223	255	231	186	wheat1					cyan4
224	255	165	79	tan1	254	139	139	0	yellow4

F17.4.12 REFERENCES

1. N. M. Greene et al., *AMPX: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Libraries from ENDF/B,* ORNL/TM-3706, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 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 V.a,* ORNL/TM-4660, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 1975.

Tables summarizing KENO-VI input data follow.

Table F17.4.8 Summary of parameter data

TITLE: The title must be entered first (80 characters, including blanks) See Sect. F17.4.3

PARAMETERS: Format: READ PARAM enter parameter data here END PARAM
If parameters are entered, they must follow the title. See Sects. F17.4.3, F17.5.2, and F17.5.3.

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=	79	albedo
TBA=	0.5 min	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	LTB=	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. prod. by array	WRS=	0	write restart
NPG=	300	no. per generation	XS2=	NO	2-D xsecs	HAL=	NO	MKA at highest level			
NSK=	3	generations skipped	PKI=	NO	fission spectrum	PLT=	YES	printer plots			
RES=	0	gens. between restart	P1D=	NO	extra 1-D xsecs	BUG=	NO	debug print			
NBK=	NPG+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=	1000000	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						
NG8=	200	blocks for d.a. unit	GAS=	NO	fiss. & abs. by reg.						
NL8=	512	length of d.a. block									
SIG=	given	standard deviation for problerm termination									

Table F17.4.9 Summary of array data

ARRAY Format: READ ARRAY array parameters data type orientation data END ARRAY See Sects. F17.5.5, F17.5.6, and F17.5.7.

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

ARRAY PARAMETERS

KEYWORD	DEFAULT	DEFINITION
ARA=	none	no. defining the array
TYP=	cuboidal	array type (cuboidal or triangular)
NUX=	none	no. of units in X direction
NUY=	none	no. of units in Y direction
NUZ=	none	no. of units in Z direction
GBL=	none	global, array number**
COM=	none	delim comment delim optional comment is a maximum of 132 characters

**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. Orientation data for FILL is terminated by entering 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	N	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 the end points, j and k, with i entries linearly interpolated between them (i.e., a total of i+2 points). At least one blank must separate j and k. When used for an integer array, the I option should only be used to generate integer steps (i.e., (k-j)/(i+1) should be a whole number).
	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. Orientation data for LOOP 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 0.
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 int he 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.
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 F17.4.10 Summary of biasing data

BIAS (weighting)

Format: READ BIAS keyword correlation data auxiliary data END BIAS
See Sects. F17.4.7 and F17.5.8

KEYWORD	DESCRIPTION		MATERIAL	ID	ENERGY GROUPS	THICKNESS/ INCREMENT
ID=	CORRELATION DATA will be read next.					
	id	material ID. Enter from table at right to use weighting data from the library	concrete	301	16,27,44,218,238	5 cm
			paraffin	400	16,27,44,218,238	3 cm
	ibgn	beginning bias ID	water	500	16,27,44,218,238	3 cm
	iend	ending bias ID	graphite	6100	16,27,44,218,238	20 cm
WT=	AUXILIARY DATA will be read next.					
WTS=	AUXILIARY DATA will be read next.					
	wttitl	material title (12-character maximum)				
	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 x ngpwt values of wtavg				

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=. When AUXILIARY DATA are entered, CORRELATION DATA must also be entered to use the data.

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

Table F17.4.11 Summary of boundary condition data

BNDS (albedo or boundary conditions)

Format: READ BNDS face code albedo name END BNDS
See Sect. F17.4.7

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

Any albedo condition on any face is allowed if the global unit boundary record consists of only one geometry record that is a cuboid.
Any albedo condition , but the same condition on all faces, is allowed if the global unit boundary record consists of one geometry record that is composed of paired planes.
Only void, mirror, or white boundary conditions, the same condition on all face, if the global unit boundary record consists of only one geometry record of any shape.
Only void boundary condition is allowed if the global unit boundary record contains multiple geometry records. Default is void on all global unit boundary surfaces.

ALBEDO NAMES AVAILABLE ON THE KENO V.a ALBEDO LIBRARY, FOR USE WITH THE FACE CODES

ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION
DP0H2O	12-in. double P0 water	CONC-4	4-in. concrete differential	VACUUM	vacuum condition
DP0H2O	differential albedo with	CON4	albedo with 4 incident angles	VOID	
DP0	4 incident angles	CONC4		VACU	
DP0				VAC	
		CONC-8	8-in. concrete differential		
H2O	12-in. water differential	CON8	albedo with 4 incident angles	SPECULAR	mirror image reflection
WATER	albedo with 4 incident angles	CONC8		MIRROR	
				MIRR	
PARAFFIN	12-in. paraffin differential	CONC-12	12-in. concrete differential	SPEC	
PARA	albedo with 4 incident angles	CON12	albedo with 4 incident angles	SPE	
WAX		CONC12		MIR	
CARBON	200-cm carbon differential	CONC-15	16-in. concrete differential	PERIODIC	periodic boundary condition
GRAPHITE	albedo with 4 incident angles	CON16	albedo with 4 incident angles	PERI	
C		CONC16		PER	
ETHYLENE	12-in. polyethylene	CONC-24	24-in. concrete differential	WHITE	white boundary condition
POLY	differential albedo with	CON24	albedo with 4 incident angles		
CH2	4 incident angles	CONC24			

Table F17.4.12 Summary of geometry data

GEOMETRY (region)

Format: READ GEOM enter geometry region data here END GEOM
See Sects. F17.4.4, F17.5.1.2, F17.5.6, and F17.5.7.

GEOMETRY REGION DATA consist of SIMPLE GEOMETRY REGION DATA and EXTENDED GEOMETRY REGION DATA.

ENTER GEOMETRY REGION DATA IN THE FOLLOWING FORM:

GLOBAL SPECIFICATION
UNIT n
OPTIONAL GEOMETRY COMMENT
GEOMETRY, MODIFICATION, MEDIA, and BOUNDARY DATA
* *

ENTER DATA IN THE FOLLOWING FORM:
GLOBAL Enter once 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.
Geometry Keyword Label Input Parameters Optional origin data (ORIGIN) Optional chord data (CHORD) Optional rotation data (ROTATE degrees)
Media Keyword Input Parameters
Boundary Keyword Input Parameters Each unit must have one boundary record.
enter as many geometry and media descriptions as necessary to describe the unit and as many units as necessary to describe the system.

GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	KEYWORD	INPUT PARAMETERS	NOTES
GEOMETRY	CONE	Lbl Rt Zt Rb Zb	
	CUBOID	Lbl +X -X +Y -Y +Z -Z	
	CYLINDER	Lbl R Zt Zb	
	DODECAHEDRON	Lbl R	
	ECYLINDER	Lbl Rx Ry Zt Zb	
	ELLIPSOID	Lbl Rx Ry Rz	
	HEXPRISM	Lbl R Zt Zb	
	HOPPER	Lbl Lxt Lyt Zt Lxb Lyb Zb	
	PARALLELPIPED	Lbl XDIST YDIST ZDIST PSI THETA PHI	
	PLANE	Lbl XPL=a YPL=b ZPL=c CON=d	Data are entered after the = following the respective subordinate keyword. Only nonzero data are required.
	QUADRATIC	Lbl AQU=a BQU=b CQU=c DQU=d EQU=e FQU=f GQU=g HQU=h IQU=i JQU=j	Data are entered after the = following the respective subordinate keyword. Only nonzero data are required.
	RHOMBOID	Lbl DX PSI	
	SPHERE	Lbl R	
	WEDGE	Lbl XBASE XPT YPT ZLNG	
MODIFICATION*	CHORD	+X= -X= +Y= -Y= +Z= -Z=	Only nonzero data are required.
	ORIGIN	X= Y= Z=	Only nonzero data are required.
	ROTATE	A1= A2= A3=	The body is rotated about the Origin using the Euler angle x-convention. Only nonzero data are required.
MEDIA**	MEDIA	Mid Bid i1 i2...	
	HOLE	Hid	
	ARRAY	Aid i1 i2... PLACE nx ny nz xx yy zz	Place positions the array by placing the origin of the unit in array position nx ny nz at xx yy zz. The region boundary containing the array must be enclosed in or on the array boundary.
BOUNDARY	BOUNDARY	i1 i2 ...	Defines the overall volume of the unit.

Table F17.4.13 Summary of mixing table data

MIXTURES

Format: READ MIXT xsec parameters END MIXT
These data are entered only if an AMPX working-format library is being used. (LIB=) in the parameter data, Sect. F17.4.3.
Do not enter if an ICE mixed library is used, (XSC=) in the parameter data. See Sects. F17.4.10 and F17.5.5.

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=	0.00003	cross-section message cutoff value use to suppress message No. K6-60

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 F17.4.14 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 Sects. F17.4.1 and F17.5.9

KEYWORD	DEFAULT	DEFINITION
TTL=	prob. title	delim ptitl delim delim is a one-character delimiter that signals the beginning and end of the title. ptitl is the plot title (max. 132 char.)
PIC=	MAT	Type of plot: MIXTURE, UNIT NO. or BIAS ID NO.
		MIXTURE ---------- MAT
		MIX
		MIXT
		MIXTURE
		MEDI
		MEDIA
		UNIT NO. ----------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=	8.0 (character plots) 10 (color plots)	Vertical to horizontal scaling factor for plot proportionality.
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
SCR=	YES	Display plot method SCR=YES utilizes color plot display SCR=NO utilizes printer plot display
NCH=	CHRS*	delim CHRS delim a one character delimiter signals the beginning and end of the character string
CLR=	Table F17.4.7	num(i) red(num(i)) green (num(i)) blue (num(i)) num(i) defines mix. no., unit no., or bias ID next 3 entries define red, green, and blue components of the color representing num(i).

PLOT ORIGIN:
Coincides with origin of geometry description of the global unit.

*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 F17.4.15 Summary of starting data

START Format: READ START enter start data here END START
The default value of start type is zero. See Sect. F17.4.8.

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	

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=	**	fissile mixture no.
LNU=	0	number of last
NBX=	0	neutron
FCT=	0	source unit number
XSM=	-X	fraction
XSP=	+X	-X of source cuboid
YSM=	-Y	+X of source cuboid
YSP=	+Y	-Y of source cuboid
ZSM=	-Z	+Y of source cuboid
ZSP=	+Z	-Z of source cuboid
RFL=	NO	+Z of source cuboid
PS6=	NO	start in reflector
PSP=	NO	print start 6 input
		print starting points

*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.
**Default to the fissionable mixture by the smallest mixture number.
Unless otherwise specified, the starting volume is defined by the unrotated, untranslated geometric shape specified in the first position in the global unit boundary record.

F17.5 NOTES FOR KENO-VI USERS

This section provides assorted tips primarily designed to assist the KENO-VI user with problem mock-ups. Some information concerning methods utilized by KENO-VI is also included.

F17.5.1 DATA ENTRY

The KENO-VI data input is entered in blocks that begin and end with keywords as described in Sect. F17.4.1. 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.

F17.5.1.1 *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

F17.5.1.2 *Multiple Entries in Geometry Data*

Individual geometry regions cannot be replaced by adding an additional description. However, entire unit 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 10 5.0 CUBOID 20 10.0 -10.0 10.0 -10.0 10.0 -10.0
MEDIA 1 1 10 MEDIA 0 1 20 -10 BOUNDARY 20
UNIT 2 CYLINDER 10 2.0 5.0 -5.0 CUBOID 20 10.0 -10.0 10.0 -10.0 10.0 -10.0
MEDIA 1 1 10 MEDIA 0 1 20 -10 BOUNDARY 20
UNIT 1 CUBOID 10 1.0 -1.5 2.5 -2.0 5.0 -6.0 CUBOID 20 10.0 -10.0 10.0 -10.0 10.0 -10.0
MEDIA 1 1 10 MEDIA 0 1 -10 20 BOUNDARY 20
END GEOM

is the same as entering

READ GEOM UNIT 1 CUBOID 10 1.0 -1.5 2.5 -2.0 5.0 -6.0
CUBOID 20 10.0 -10.0 10.0 -10.0 10.0 -10.0
MEDIA 1 1 10 BOUNDARY MEDIA 0 1 -10 20
UNIT 2 CYLINDER 10 2.0 5.0 -5.0 CUBOID 20 10.0 -10.0 10.0 -10 10.0 -10.0
MEDIA 1 1 10 MEDIA 0 1 -10 20 BOUNDARY 20 END GEOM

or

READ GEOM UNIT 2 CYLINDER 30 2.0 5.0 -5.0 CUBOID 40 6P10.0
MEDIA 1 1 30 MEDIA 0 1 -30 40 BOUNDARY 40
UNIT 1 CUBOID 20 1.0 -1.5 2.5 -2.0 5.0 -6.0 CUBOID 10 6P10.0
MEDIA 1 1 20 MEDIA 0 1 10 -20 BOUNDARY 10
END GEOM

The order of entry for unit 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.

F17.5.2 DEFAULT LOGICAL UNIT NUMBERS FOR KENO-VI

The logical unit numbers for data utilized by KENO-VI are listed in Table F17.5.1.

F17.5.3 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 input 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.

Table F17.5.1 KENO-VI logical unit numbers

Function	Parameter name	Unit No.	Variable name
Problem input data (EBCDIC or ASCII)		5	INPT
Problem input data (binary)		95	BIN
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.

Certain parameter default values should not be overridden unless the user has a very good reason to do so. These parameters are as follows:

1. **X1D=** 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=** defines the number of neutrons that can be entered in the fission bank;

3. **XFB=** 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=** defines the number of neutrons that can be entered in the neutron bank;

5. **XNB=** defines the number of extra positions in the neutron bank (the neutron bank contains information about each history);

6. **WTH=** defines the factor that determines when splitting occurs;

7. **WTA=** defines the default average weight given to a neutron that survives Russian roulette;

8. **WTL=** defines the factor that determines when Russian roulette is played; and

9. **LNG=** 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 keyword 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.

F17.5.4 CROSS SECTIONS

KENO-VI 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[1] (Sect. F8). A mixed cross-section file can be created by (1) executing ICE or (2) using an AMPX working-format library and entering mixing table data in KENO-VI.

F17.5.4.1 *Use An ICE Mixed Cross-Section MORSE/KENO-V.a Format Library*

An ICE mixed cross-section MORSE/KENO-Va format library (premixed cross-section data file) from ICE or a previous KENO-VI case may be used. This file should be specified in the job control language on the unit number associated with the parameter XSC=. If 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-VI case are consistent with those used in the geometry data of the problem.

F17.5.4.2 *Use an AMPX Working-Format Library*

When an AMPX working-format library is used, a file definition must be specified in the job control language to specify the AMPX working-format library on the unit associated with the parameter LIB=. If the mixed cross-section data file is to be saved, a file definition must be specified 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 format library is used. IDs used in the mixing table must match the IDs on the AMPX working-format library.

F17.5.4.3 *Number of Scattering Angles*

The number of scattering angles is defaulted to 1. 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. F17.4.10.

The order of the last Legendre coefficient to be preserved in the scattering distribution is equal to (2 × SCT − 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.

F17.5.4.4 *Cross-Section Message Cutoff*

The cross-section message cutoff value, *pbxs*, is defaulted to 3×10^{-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. F17.4.10.

The default value of *pbxs* is sufficient to ensure that warning messages will not be printed for most of the SCALE P_1 and P_3 cross-section libraries. However, the 238GROUPNDF5 library requires a value of *pbxs* as large as 1×10^{-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=6.9-5 would cause all three messages to be suppressed. A value less than 5.615159-5 and greater than 4.767635-5 would suppress the second message, and a value less than 6.855362-5 and greater than 5.615159-5 would suppress the first two messages.

```
K6-60   THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE 2 HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP 28 TO GROUP 72
        1 MOMENTS WERE ACCEPTED
        THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
        5.615159E-05  1.155527E-06 -2.804013E-05 -1.732067E-06
        THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
        2.057870E-02  4.234578E-04  8.710817E-06
         THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
        2.057870E-02  4.235078E-04  8.710817E-06
        THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
        5.615159E-05  1.155527E-06 -2.804011E-05 -1.732066E-06
        THE WEIGHTS/ANGLES FOR THIS DISTRIBUTION ARE
        9.999995E-01  5.268617E-07
        2.057881E-02 -1.973451E-01
        THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
        2.057870E-02  4.235078E-04  8.710817E-06

K6-60   THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE 2 HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP 31 TO GROUP 75
        1 MOMENTS WERE ACCEPTED
        THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
        4.767635E-05  7.834378E-07 -2.381887E-05 -1.174626E-06
        THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
        1.643242E-02  2.700205E-04  4.451724E-06
        THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
        1.643242E-02  2.700282E-04  4.437279E-06
```

```
          THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
          4.767635E-05  7.834378E-07 -2.381885E-05 -1.174627E-06
          THE WEIGHTS/ANGLES FOR THIS DISTRIBUTION ARE
          9.999858E-01  1.420136E-05
          1.643265E-02 -2.334324E-07
          THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
          1.643242E-02  2.700282E-04  4.437279E-06

K6-60     THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE 2 HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP 32 TO GROUP 74
          1 MOMENTS WERE ACCEPTED
          THE LEGENDRE EXPANSION OF THE CROSS SECTION (P0-PN) IS
          6.855362E-05  1.341944E-06 -3.423741E-05 -2.011613E-06
          THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
          1.957510E-02  3.831484E-04  7.601939E-06
          THE MOMENTS CORRESPONDING TO THE GENERATED DISTRIBUTION ARE
          1.957510E-02  3.832207E-04  7.502292E-06
          THE LEGENDRE EXPANSION CORRESPONDING TO THESE MOMENTS IS
          6.855362E-05  1.341944E-06 -3.423740E-05 -2.011629E-06
          THE WEIGHTS/ANGLES FOR THIS DISTRIBUTION ARE
          9.999056E-01  9.437981E-05
          1.957695E-02 -1.848551E-06
          THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
          1.957510E-02  3.832207E-04  7.502292E-06
```

The user need not attempt to suppress all the K6-60 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 third through sixth lines of the message list the original P_n coefficients and their moments. The seventh through tenth lines list the new corrected moments and their corresponding P_n coefficients.

The weights and angles that are printed in lines 11 through 13 were generated from the corrected moments. The last two lines of the message list the moments generated from those weights and angles. They should match line 8, the moments corresponding to the generated distribution.

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.

F17.5.5 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 performs mixing 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 can be 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 option 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 option 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 option 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 option 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 (0.75×0.5) 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 situation occurs because the nuclide mixing loop is done first, thus defining mixture 2, and 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 option 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.

F17.5.6 GEOMETRY

In general, KENO-VI geometry descriptions consist of (1) geometry data (Sect. F17.4.4) that define the geometrical shapes present in the problem, and (2) array data (Sect. F17.4.5) that define 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 3-D geometrical configuration is described as KENO-VI geometry data, it may be necessary to describe portions of the configuration individually. These individual partial descriptions of the configuration are called units. KENO-VI geometry modeling is subject to the following restrictions:

1. Units are composed of regions. These regions are created using geometric bodies and surfaces that are previously defined or created using quadratic equations. The geometric bodies and surfaces may intersect. Regions are defined relative to the geometric bodies and surfaces in a unit. Holes provide a means of creating complex geometries in a unit and then inserting the unit into existing units. For complex geometries the use of holes may decrease the CPU time required for the problem.

2. All geometrical surfaces must be describable as a set of quadratic equations.

3. When one or more units are utilized to describe an array, each unit used in the array must have a cuboid or hexagonal prism 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. Units are placed directly into regions using HOLES. Only one hole is allowed in a region. As many holes as will snugly fit without intersecting can be placed in a unit. The outer boundary of the unit in the hole must match exactly the region boundary. Holes are described in more detail in Sect. F17.5.6.1, and nested holes are described in Sect. F17.5.6.2.

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 that may be placed in other units either directly or by using holes. Multiple arrays are described in more detail in Sect. F17.5.6.3.

The geometry package allows any shape describable using quadratic equations to be enclosed or intersected by any other allowable shape. Geometry records are used to describe the geometric shapes contained in a unit. Regions are defined by specifying how they relate to the geometric shapes in the unit. An implication of this type of description is that an entire region volume, specified relative to the geometric shapes, contains only one mixture, hole, or array. Media records are used to specify the contents of each region. Material, hole, and array contents are specified using the keywords: Media, Hole, and Array, respectively, in the media records. A void is specified using the keyword Media with a mixture ID of zero.

If the problem contains arrays, each unit that is used in an array must have as its outer surface a rectangular parallelepiped or a hexagonal prism; therefore it may be necessary to define a void region that is used to achieve the required outer boundary. In order to describe the composite overall geometrical characteristics of the problem, these units may be arranged in either a rectangular or hexagonal array by specifying the number of units in the x, y, and z directions. Data must be entered to define the number assigned to the array and the placement of the individual units in the array. The array type is specified using the keyword TYP=, followed by either square or triangular. If TYP is not entered, a cuboidal array is assumed. The array number, the number of units in the x, y, and z directions, the array type, and the placement data are called array data (Sect. 17.4.5).

An array may be placed in a region of any shape provided the region boundary is contained within the array or shares the array boundary. Only the section of the array contained within the region is recognized by the problem. A particle crosses into and out of the array when it crosses the region boundary. Arrays are

positioned relative to the unit origin by placing the origin of a specified unit in the array at a specified location in the unit. The region boundary containing the array is defined using a vector definition array. A global array is not assumed. If a global array is required, such as for collecting matrix information, one must be specified.

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 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 neutron mean free path, a larger percentage of time is spent processing the transformation of coordinates. Because all boundaries in a unit must be checked for crossings, it may be more efficient to break up complex units into several smaller, simpler units. The trade-off involves the time required to process more boundary crossings vs the time required to transform coordinate systems when unit boundaries are crossed.

Geometry dimensions: The geometry dimensions utilized in KENO-VI require an entry for each required dimension. For example, a 20 × 20 × 2.5-cm rectangular parallelepiped would have been described as: CUBOID 1 10.0 -10.0 10.0 -10.0 1.25 -1.25. By using the P option (see Table F17.4.2), the same rectangular parallelepiped could be described as: CUBOID 1 4P10.0 2P1.25 where the first entry is the geometry word label and the remaining entries describe the geometry. The P option simply repeats 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 keyword is followed by a comment whose maximum length is 132 characters. The comment must be preceded and terminated by a delimiter character, which is the first nonblank 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. F17.4.4). See the following example.

```
READ GEOM
UNIT 1
COM=*SPHERICAL METAL UNIT*
SPHERE 1  5.0
CUBOID 2  6P5.0
MEDIA 1 1 1 2
MEDIA 0 1 -1 2
BOUNDARY 2
UNIT 2
CYLINDER 1 5.0 2P5.0
CUBOID 2 6P5.0
MEDIA 1 1 1 2
COM=/CYLINDRICAL METAL UNIT/
MEDIA 0 1 -1 2
```

```
BOUNDARY 2
UNIT 3
SPHERE 1  5.0 CHORD +X=0.0
MEDIA 1 1 1 2
COM='HEMISPHERICAL METAL UNIT'
CUBOID 2  6P5.0
MEDIA 0 1 -1 2
BOUNDARY 2
UNIT 4
COM='ARRAY OF SPHERICAL UNITS'
CUBOID 1 6P15
ARRAY 1 1 PLACE 2 2 2 3*0.0
BOUNDARY 1
UNIT 5
CUBOID 1 6P15.0
COM='ARRAY OF CYLINDRICAL UNITS'
ARRAY 2 1 PLACE 2 2 2 3*0.0
BOUNDARY 1
UNIT 6
COM='ARRAY OF HEMISPHERICAL UNITS'
CUBOID 1 6P15.0
ARRAY 3 1 PLACE 2 2 2 3*0.0
BOUNDARY 1
GLOBAL UNIT 7
COM='ARRAY OF ARRAYS
CUBOID 1 4P15.0 2P45.O
ARRAY 4 1 PLACE 1 1 2 3*0.0
BOUNDARY 1
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. F17.4.5). Examples of correct array comments are given below.

```
READ ARRAY
COM='ARRA OF SPHERICAL METAL UNITS'
ARA=1 NUX=3 NUY=3 NUZ=3 FILL F1 END FILL
ARA=2 COM='ARRAY OF CYLINDRICAL METAL UNITS'
NUX=3 NUY=3 NUZ=3 FILL F2 END FILL
ARA=3 NUX=3 NUY=3 NUZ=3
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-VI 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. F17.5.1.

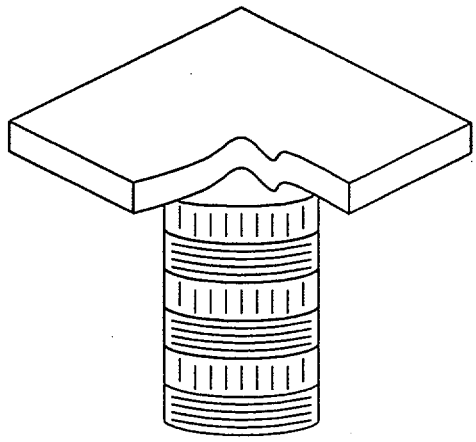


Figure F17.5.1 Stack of disks with a square cap

This problem can be described in a number of ways: as a set of stacked cylinders with a top plate, a set of nested cylinders with a top plate, and an array of cylinders with a top plate, to name just a few. Several alternative acceptable methods that could be used to describe this problem are given below.

This problem can be described using one unit. 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. 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. This method could be used in KENO-V.a to describe this problem. The geometry input is shown below.

Data description 1, Example 1.

```
READ GEOM
GLOBAL UNIT 1
CYLINDER  1  5.0 2.0   0.0
CYLINDER  2  5.0 4.0   2.0
CYLINDER  3  5.0 6.0   4.0
```

```
CYLINDER  4  5.0 8.0   6.0
CYLINDER  5  5.0 10.0  8.0
CYLINDER  6  5.0 12.0  10.0
CUBOID  7  10.0 -10.0 10.0 -10 12.0 0
CUBOID  8  10.0 -10.0 10.0 -10.0 14.5 0.0
MEDIA 1 1 1
MEDIA 2 1 -1 2
MEDIA 1 1 -2 3
MEDIA 2 1 -3 4
MEDIA 1 1 -4 5
MEDIA 2 1 -5 6
MEDIA 0 1 -1 -6 7
MEDIA 3 1 -7 8
BOUNDARY  8
END GEOM
```

An alternative 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. This method could also be used in KENO-V.a to describe the problem.

Data description 2, Example 1.

```
READ GEOM
GLOBAL UNIT 1
CYLINDER 10  5.0   1.0 -1.0
CYLINDER 20  5.0   3.0 -3.0
CYLINDER 30  5.0   5.0 -5.0
CYLINDER 40  5.0   7.0 -5.0
CUBOID   50 10.0 -10.0 10.0 -10.0 7.0 -5.0
CUBOID   60 10.0 -10.0 10.0 -10.0 9.5 -5.0
MEDIA 1 1 10
MEDIA 2 1 -10 20
MEDIA 1 1 -10 -20 30
MEDIA 2 1 -10 -20 -30 40
MEDIA 0 1 -30 -40 50
MEDIA 3 1 -40 -50 60
BOUNDARY  60
END GEOM
```

Example 1 can also be described as an 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  5.0   2.0  0.0
CUBOID  2 10.0 -10.0 10.0 -10.0 2.0 0.0
MEDIA 1 1 1
MEDIA 0 1 -1 2
BOUNDARY 2
UNIT 2
CYLINDER  1  5.0   2.0  0.0
CUBOID    2 10.0 -10.0 10.0 -10.0 2.0 0.0
MEDIA 2 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
UNIT 3
CUBOID    1 10.0 -10.0 10.0 -10.0 2.5 0.0
MEDIA 31 1
BOUNDARY 1
GLOBAL UNIT 4
CUBOID 1  10 -10 10 -10 14.5 0.0
ARRAY 1 1  PLACE 1 1 1 3*0.0
BOUNDARY 1
END GEOM
READ ARRAY ARA=1 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  5.0   1.0 -1.0
CUBOID  2   10.0 -10.0 10.0 -10.0 1.0 -1.0
MEDIA 1 1 1
MEDIA 0 1 -1 2
BOUNDARY  2
UNIT 2
CYLINDER 1  5.0   1.0 -1.0
CUBOID  2  10.0 -10.0 10.0 -10.0 1.0 -1.0
MEDIA 2 1 1
MEDIA 0 1 -1 2
BOUNDARY  2
UNIT 3
CUBOID 1 10.0 -10.0 10.0 -10.0 1.25 -1.25
MEDIA 3 1 1
```

BOUNDARY 1
GLOBAL UNIT 4
CUBOID 1 10 -10 10 -10 14.5 0.0
ARRAY 1 1 PLACE 1 1 1 0.0 0.0 1.0
BOUNDARY 1
END GEOM
READ ARRAY ARA=1 NUX=1 NUY=1 NUZ=7 FILL 1 2 1 2 1 2 3 END ARRAY

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. Care should be taken when assigning coordinates to the unit used to place the array in its surrounding region.

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 5.0 2.0 0.0
CYLINDER 2 5.0 4.0 2.0
CUBOID 3 10.0 -10.0 10.0 -10.0 4.0 0.0
MEDIA 1 1 1
MEDIA 2 1 -1 2
MEDIA 0 1 -1 -2 3
BOUNDARY 3
UNIT 2
CUBOID 1 10.0 -10.0 10.0 -10.0 1.25 -1.25
MEDIA 3 1 1
BOUNDARY 1
GLOBAL UNIT 3
CUBOID 1 10 -10 10 -10 14.5 0.0
*ARRAY 1 1 PLACE 1 1 1 3*0.0*
BOUNDARY 1
END GEOM
READ ARRAY ARA=1 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 global unit 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  5.0   2.0  0.0
CUBOID  2   10.0 -10.0 10.0 -10.0 2.0  0.0
MEDIA 1 1 1
MEDIA 0 1 -1 2
BOUNDARY  2
UNIT 2
CYLINDER  1  5.0   2.0  0.0
CUBOID  2  10.0 -10.0 10.0 -10.0 2.0  0.0
MEDIA 2 1 1
MEDIA 0 1 -1 2
BOUNDARY  2
GLOBAL UNIT 3
CUBOID 1  10.0 -10.0 10.0 -10.0 6.0 -6.0
CUBOID 2  10.0 -10.0 10.0 -10.0 8.5 -6.0
ARRAY 1 1 PLACE 1 1 1 0.0 0.0 -6.0
MEDIA 3 1 -1 2
BOUNDARY  2
END GEOM
READ ARRAY ARA=1 NUX=1 NUY=1 NUZ=6 FILL 1 2 1 2 1 2 END ARRAY
```

The user could have chosen the origin of the global unit to be at the center bottom of the array in which case the geometry description for the global unit would be:

```
CUBOID  1  10.0 -10.0 10.0 -10.0 12.0 0.0
CUBOID  2  10.0 -10.0 10.0 -10.0 14.5 0.0
ARRAY 1 1 PLACE 1 1 1 3*0.0
MEDIA 3 1 -1 2
BOUNDARY  2
```

Any position in the array could be used to place the array within the region, keeping in mind the restriction that the region boundary must either coincide with the surface of the array or be contained within the array. If array position (1,1,3) was used to position the array in description 6 of example 1, the global unit would be:

```
CUBOID  1  10.0 -10.0 10.0 -10.0 12.0 0.0
CUBOID  2  10.0 -10.0 10.0 -10.0 14.5 0.0
ARRAY 1 1 PLACE 1 1 3 0.0 0.0 4.0
MEDIA 3 1 -1 2
BOUNDARY  2
```

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
UNIT 1
CYLINDER 1  5.0  2.0  0.0
CYLINDER 2  5.0  4.0  2.0
CUBOID  3  10.0 -10.0 10.0 -10.0 4.0 0.0
MEDIA 1 1 1
MEDIA 2 1 -1 2
MEDIA 0 1 -1 -2 3
BOUNDARY 3
GLOBAL UNIT 2
CUBOID  1  10.0 -10.0 10.0 -10.0  12.0 0.0
CUBOID  2  10.0 -10.0 10.0 -10.0 14.5 0.0
ARRAY 1 1 PLACE 1 1 1 3*0.0
MEDIA 3 1 -1 2
BOUNDARY 2
END GEOM
READ ARRAY ARA=1 NUX=1 NUY=1 NUZ=3 FILL F1  END FILL  END ARRAY
```

Note that a global unit must be specified for each problem, including a single unit problem. Also each unit must have a boundary record specifying the outermost boundary of the unit.

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
GLOBAL UNIT 1
CYLINDER 1 5.0   1.0  -1.0
CYLINDER 2 5.0   3.0  -3.0
CYLINDER 3 5.0   5.0  -5.0
CYLINDER 4 5.0   7.0  -5.0
CYLINDER 5 8.0  13.0  -5.0
CYLINDER 6 8.25 13.0  -5.25
CUBOID 7 10.0  -10.0  10.0 -10.0 13.0 -5.25
CUBOID 8 10.0  -10.0  10.0 -10.0 15.5 -5.25
MEDIA 1 1 1
MEDIA 2 1 -1 2
MEDIA 1 1 -1 -2 3
MEDIA 2 1 -1 -2 -3 4
MEDIA 0 1 -3 -4 5
MEDIA 6 1 -5 6
```

MEDIA 0 1 -5 -6 7
MEDIA 3 1 -6 -7 8
BOUNDARY 8
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 of cylinders. 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 description 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 description defines the square plate and the unit boundary. Omission of the first cuboid description 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 in this description.

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 description is shown below.

Data description 2, Example 2.

READ GEOM
UNIT 1
CYLINDER 1 5.0 2.0 0.0
CYLINDER 2 5.0 4.0 2.0
CUBOID 3 5.0 -5.0 5.0 -5.0 4.0 0.0
MEDIA 1 1 1
MEDIA 2 1 -1 2
MEDIA 0 1 -1 -2 3
BOUNDARY 3
GLOBAL UNIT 2
CUBOID 1 5.0 -5.0 5.0 -5.0 12.0 0.0
*ARRAY 1 1 PLACE 1 1 1 3*0.0*
CYLINDER 2 8.0 18.0 0.0
MEDIA 0 1 -1 2
CYLINDER 3 8.25 18.0 -0.25
MEDIA 6 1 -1 -2 3
CUBOID 4 10.0 -10.0 10.0 -10.0 20.5 18.0
CUBOID 5 10.0 -10.0 10.0 -10.0 20.5 -0.25
MEDIA 3 1 -2 -3 4
MEDIA 0 1 -3 -4 5
BOUNDARY 5
END GEOM
READ ARRAY ARA NUX=1 NUY=1 NUZ=3 FILL F1 END FILL 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. F17.5.1. This unit is used to build the array portion of the geometry description. The first cuboid in unit 2 contains the array of disks. Everything after the first cuboid is considered part of the reflector. The first cylinder in Unit 2 defines the void interior of the cylindrical container. The next cylinder defines the walls of the container. The next-to-last cuboid defines the square plate of material 3 that is sitting on top of the container. The 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. Example 2 can be described as a reflected array regardless of the inner radius of the container because, unlike KENO-V.a, intersections are allowed and only the section of the array in the region boundary is considered part of the problem.

Another method to describe Example 2 is as an array composed of units that contain both the stack and container. This description requires a minimum of four units to describe the problem. This configuration is given below in data description 3, Example 2.

Data description 3, Example 2.

```
READ GEOM
UNIT 1
CYLINDER  1  8.25   0.25   0.0
CUBOID  2   10.0  -10.0   10.0 -10.0 0.25  0.0
MEDIA 6 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
UNIT 2
CYLINDER  10  5.0   2.0   0.0
CYLINDER  20  5.0   4.0   0.0
CYLINDER  30  8.0   4.0   0.0
CYLINDER  40  8.25  4.0   0.0
CUBOID  50   10.0  -10.0   10.0 -10.0 4.0   0.0
MEDIA 1 1 10
MEDIA 2 1 20 -10
MEDIA 0 1 30 -20 -10
MEDIA 6 1 40 -30 -20 -10
MEDIA 0 1 50 -40 -30 -20 -10
BOUNDARY  50
UNIT 3
CYLINDER  1  8.0   3.0  -3.0
CYLINDER  2  8.25  3.0  -3.0
CUBOID   3 10.0  -10.0   10.0 -10.0 5.5   3.0
CUBOID   4 10.0  -10.0   10.0 -10.0 5.5  -3.0
MEDIA 0 1 1
MEDIA 6 1 2 -1
MEDIA 3 1 3 -2 -1
MEDIA 0 1 4 -3 -2- 1
BOUNDARY  4
GLOBAL UNIT 4
CUBOID  1  10.0 -10.0 10.0 -10.0 20.75 0.0
```

*ARRAY 1 1 PLACE 1 1 1 3*0.0*
BOUNDARY 1
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 - (3 × 4.0) = 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 - (3 × 4.0) = 6.0]. Unit 4 is the global unit that contains the array. The array is placed in the unit by aligning the bottom center of Unit 1 with the origin of Unit 4.

EXAMPLE 3. Refer to Example 1, Fig. F17.5.1, and imagine a hole 1.5 cm in diameter drilled along the centerline of the stack through the disks and the square plate. In KENO-V.a this hole would eliminate the possibility of describing the system as a single unit 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. However, KENO-VI can easily describe this configuration as a single unit.

Data description 1, Example 3.

READ GEOM
GLOBAL UNIT 1
CYLINDER 1 0.75 7.0 -5.0
CYLINDER 2 5.0 1.0 -1.0
CYLINDER 3 5.0 3.0 -3.0
CYLINDER 4 5.0 5.0 -5.0
CYLINDER 5 5.0 7.0 -5.0
CYLINDER 6 8.0 13.0 -5.0
CYLINDER 7 8.25 13.0 -5.25
CUBOID 8 10.0 -10.0 10.0 -10.0 15.5 13.0
CUBOID 9 10.0 -10.0 10.0 -10.0 15.5 -5.25
MEDIA 0 1 1
MEDIA 1 1 2 -1
MEDIA 2 1 -1 -2 3
MEDIA 1 1 -1 -2 -3 4
MEDIA 2 1 -1 -2 -3 -4 5
MEDIA 0 1 -1 -2 -3 -4 -5 6
MEDIA 6 1 -1 -4 -5 -6 7
MEDIA 3 1 -6 -7 8

MEDIA 0 1 -7 -8 9
BOUNDARY 9
END GEOM

In data description 1, Example 3 above, Unit 1 contains the entire problem description. The first cylinder describes the 1.5-cm-diam hole through the stack. The next four cylinders define the stack. The sixth and seventh cylinders describe the void and container. The two cuboids describe the top plate and surrounding global region of void. The media cards are used to place the materials in the appropriate regions.

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. F17.5.2. 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.

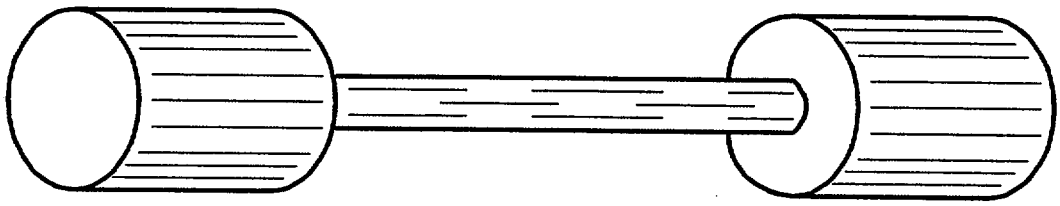


Figure F17.5.2 Two large cylinders joined axially by a small cylinder

Data description 1, Example 4.

READ GEOM
GLOBAL UNIT 1
CYLINDER 1 0.5 5.0 -5.0
CYLINDER 2 2.5 5.0 -5.0
CYLINDER 3 2.5 10.0 -10.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 1 1 3 -2
BOUNDARY 3
END GEOM

The origin is at the center of the small cylinder, which is described by the first cylinder description. The second cylinder description defines a void cylinder surrounding the small cylinder. Its radius is the same as the large cylinders, and its height (length) coincides with that of the small cylinder. The last cylinder description defines the large cylinders on either end of the small cylinder. Because the problem was not rotated, the long axes of the cylinders are assumed to coincide with the z axis.

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. F17.5.3.

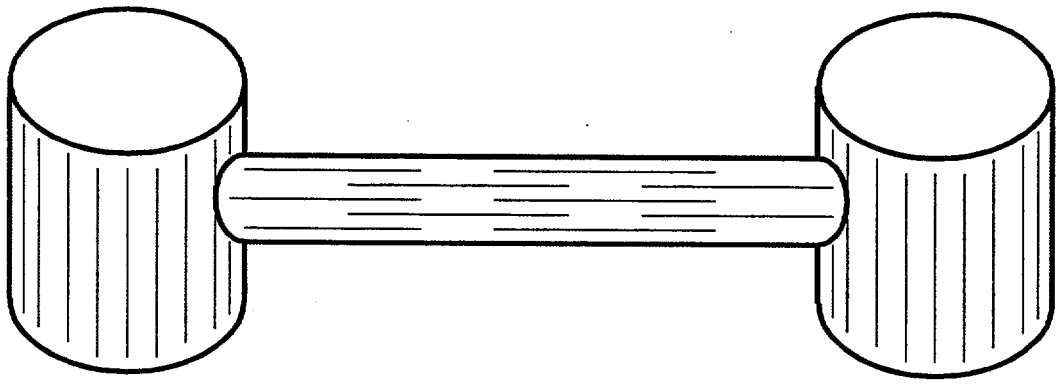


Figure F17.5.3 Two large cylinders radially connected by a small cylinder

This system can be rigorously described in KENO-VI geometry because intersections are allowed. However, either the two large cylinders or the small connecting cylinder must be rotated The small cylinder is rotated in description 1, example 5. The two large cylinders are rotated in description 2, example 5.

Data description 1, Example 5.

```
READ GEOM
GLOBAL UNIT 1
CYLINDER  1  2.5 2.5 -2.5
CYLINDER  2  2.5 2.5  -2.5 ORIGIN Y=15.0
CYLINDER  3 1.5 15.0 0.0  ROTATE A2=-90
CUBOID  4  5.0 -5.0 17.5 -2.5 2.5 -2.5
MEDIA 1 1 1
MEDIA 1 1 2
MEDIA 1 1 3 -1 -2
MEDIA 0 1 4 -3 -2 -1
BOUNDARY  4
END GEOM
```

The first and second cylinders define the two large cylinders, and the third cylinder describes the small connecting cylinder. The two large cylinders are oriented along the z axis. The second large cylinder is translated so its origin is at position (0.0, 15.0, 0.0). The small cylinder is rotated so that its centerline is parallel to the y axis. Region 1 consists of the material in the first large cylinder. Region 2 consists of the material in the second large cylinder. Region 3 consists of the material in the small cylinder but not in either of the large cylinders. Region 4 is the boundary region.

Data description 2, Example 5.

```
READ GEOMETRY
GLOBAL UNIT 1
CYLINDER  1  2.5 2.5 -2.5 ROTATE A2=90
CYLINDER  2 2.5 2.5  -2.5 ROTATE A2=90 ORIGIN Z=15.0
CYLINDER  3 1.5 15.0 0.0
CUBOID  4  2.5 -2.5 2.5 -2.5 17.5 -2.5
MEDIA 1 1 1
MEDIA 1 1 2
MEDIA 1 1 3 -1 -2
MEDIA 0 1 4 -3 -2 -1
BOUNDARY  4
END GEOM
```

Again, the first and second cylinders describe the two large cylinders. They are both rotated so their long axis is parallel to the y axis. The first large cylinder still maintains its origin at the unit origin. The origin of the second large cylinder is translated to position (0.0, 0.0, 15.0). The small connecting cylinder has its centerline along the z axis. The media data are identical to that of the previous problem description. It does not matter what order the rotation and translation data are listed after a geometry card. Rotations are always done about the origin prior to the translation. Only nonzero data need be entered for the ROTATE and ORIGIN options.

EXAMPLE 6. Assume two 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. F17.5.4.

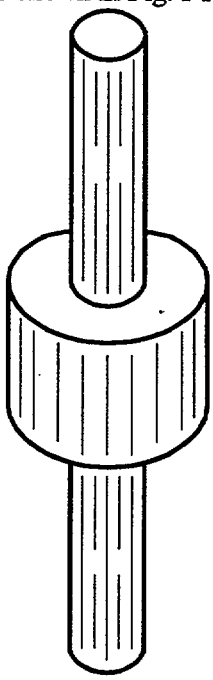


Figure F17.5.4 Two small cylinders joined axially by a large cylinder

This problem is very similar to example 4. It can be described as a single unit. 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. Assume the centerline of the cylinders lies along the z axis.

Data description 1, Example 6.

```
READ GEOM
GLOBAL UNIT 1
CYLINDER  1  1.0 12.5  -12.5
CYLINDER  2  2.5   2.5  -2.5
CUBOID   3   4P2.5 12.5 -12.5
MEDIA 1 1 1
MEDIA 1 1 2 -1
MEDIA 0 1 3 -2 -1
BOUNDARY  3
END GEOM
```

The first cylinder defines a long, narrow cylinder that includes the top and bottom small cylinders and the inner 1.0-radius cylinder of the large cylinder. The second cylinder defines the remainder of the large cylinder. The cuboid defines the boundary region. It is possible to represent this problem as a single unit because regions do not need to fully enclose all previous regions in the unit as KENO-V.a requires.

EXAMPLE 7. Assume an $11 \times 5 \times 3$ square-pitched 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
UNIT 1
SPHERE  1  3.75
CUBOID  2  6P5.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
GLOBAL UNIT 2
CUBOID 10 55.0 -55.0 25.0 -25.0 15.0 -15.0
ARRAY 1 10 PLACE 6 3 2 3*0.0
BOUNDARY  10
END GEOM
READ ARRAY NUX=11  NUY=5 NUZ=3 FILL F1 END FILL END ARRAY
```

EXAMPLE 8. Assume an 11 × 5 × 3 square-pitched 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
UNIT 1
SPHERE 1 3.75
CUBOID 2 5.0 -5.0 7.5 -7.5 10.0 -10.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
GLOBAL UNIT 2
CUBOID 1 55.0 -55.0 37.5 -37.5 30.0 -30.0
ARRAY 1 1 PLACE 6 3 2 3*0.0
BOUNDARY 1
END GEOM
READ ARRAY NUX=11 NUY=5 NUZ=3 FILL F1 END FILL END ARRAY
```

EXAMPLE 9. Assume an 11 × 5 × 3 square-pitched 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, and weighted tracking (biasing) is to be used in the water reflector. 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
UNIT 1
SPHERE 1 3.75
CUBOID 2 6P5.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
GLOBAL UNIT 2
CUBOID 1 55.0 -55.0 25.0 -25.0 15.0 -15.0
CUBOID 2 58.0 -58.0 28.0 -28.0 18.0 -18.0
CUBOID 3 61.0 -61.0 31.0 -31.0 21.0 -21.0
CUBOID 4 64.0 -64.0 34.0 -34.0 24.0 -24.0
CUBOID 5 67.0 -67.0 37.0 -37.0 27.0 -27.0
CUBOID 6 70.0 -70.0 40.0 -40.0 30.0 -30.0
CUBOID 7 73.0 -73.0 43.0 -43.0 33.0 -33.0
CUBOID 8 76.0 -76.0 46.0 -46.0 36.0 -36.0
CUBOID 9 79.0 -79.0 49.0 -49.0 39.0 -39.0
CUBOID 10 82.0 -82.0 52.0 -52.0 42.0 -42.0
CUBOID 11 85.0 -85.0 55.0 -55.0 45.0 -45.0
```

```
ARRAY 1 1 PLACE 6 3 2 3*0.0
MEDIA 2 2 2 -1
MEDIA 2 3 3 -2
MEDIA 2 4 4 -3
MEDIA 2 5 5 -4
MEDIA 2 6 6 -5
MEDIA 2 7 7 -6
MEDIA 2 8 8 -7
MEDIA 2 9 9 -8
MEDIA 2 10 10 -9
MEDIA 2 11 11 -10
BOUNDARY 11
END GEOM
READ ARRAY NUX=11 NUY=5 NUZ=3 FILL F1 END FILL END ARRAY
READ BIAS ID=500 2 11 END BIAS
```

The basic unit used to construct the array is defined in Unit 1. The array is positioned in Unit 2 (the global unit) using the ARRAY card and the PLACE option. The array is then surrounded by ten reflector regions, each 3.0 cm thick, on all sides. The first bias ID of the reflector is 2, so the last bias ID will be 11 because 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 F17.4.5. If the biasing data block is omitted from the problem description, the ten reflector regions will not have a biasing function applied to them, and the default value of the average weight will be used. This omission 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
UNIT 1
SPHERE 1 3.75
CUBOID 2 6P5.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
GLOBAL UNIT 2
CUBOID 1 55.0 -55.0 25.0 -25.0 15.0 -15.0
CUBOID 2 58.0 -58.0 28.0 -28.0 15.0 -18.0
CUBOID 3 61.0 -61.0 31.0 -31.0 15.0 -21.0
CUBOID 4 64.0 -64.0 34.0 -34.0 15.0 -24.0
CUBOID 5 67.0 -67.0 37.0 -37.0 15.0 -27.0
CUBOID 6 70.0 -70.0 40.0 -40.0 15.0 -30.0
CUBOID 7 70.24 -70.24 40.24 -40.24 15.0 -30.24
ARRAY 1 1 PLACE 6 3 2 3*0.0
```

MEDIA 2 2 2 -1
MEDIA 2 3 3 -2
MEDIA 2 4 4 -3
MEDIA 2 5 5 -4
MEDIA 2 6 6 -5
MEDIA 2 7 7 -6
BOUNDARY 7
END GEOM
READ ARRAY NUX=11 NUY=5 NUZ=3 FILL F1 END FILL END ARRAY
READ BIAS ID=500 2 7 END BIAS

The first cuboid in Unit 2 represents the boundary for the array. The next five cuboids represent the first 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. A total thickness of 15 cm of reflector material is on the appropriate faces. The last cuboid represents 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. 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 F17.4.5. 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.

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 entire array cannot be contained within the spherical reflector.

Data description 1, Example 11.

READ GEOM
UNIT 1
SPHERE 1 3.75
CUBOID 2 6P5.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
UNIT 2
CYLINDER 1 5.0 5.0 -5.0
CUBOID 2 6P5.0
MEDIA 4 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
GLOBAL UNIT 3
CUBOID 1 55.0 -55.0 25.0 -25.0 15.0 -15.0
SPHERE 2 65.0
SPHERE 3 70.0

```
SPHERE 4 75.0
SPHERE 5 80.0
SPHERE 6 85.0
ARRAY 1 1 PLACE 6 3 2 3*0.0
MEDIA 0 1 2 -1
MEDIA 3 2 3 -2
MEDIA 3 3 4 -3
MEDIA 3 4 5 -4
MEDIA 3 5 6 -5
BOUNDARY 6
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. Unit 3 defines the global unit. The ARRAY card defines the origin of the array to be at the center of the global unit. The sphere following the ARRAY card defines the inner radius of the reflector. The next four SPHERE cards will generate four spherical regions of material 3, each 5.0 cm thick. The first ten entries following the word LOOP fill the 11 × 5 × 3 array with Unit 1. The next ten 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 5 cm and a height of 15 cm. The cylinders are then nested to create these regions . A biasing data block is not entered because default biasing is desired throughout the cylinder.

Data description 1, Example 12.

```
READ GEOM
GLOBAL UNIT 1
CYLINDER 1 1.0  1.5  -1.5
CYLINDER 2 2.0  3.0  -3.0
CYLINDER 3 3.0  4.5  -4.5
CYLINDER 4 4.0  6.0  -6.0
CYLINDER 5 5.0  7.5  -7.5
MEDIA 1 1 1
MEDIA 1 1 2 -1
MEDIA 1 1 3 -2
MEDIA 1 1 4 -3
MEDIA 1 1 5 -4
BOUNDARY 5
END GEOM
```

EXAMPLE 13. Assume a cross composed of two Plexiglas cylinders (material 3) having an inner diameter of 13.335 cm and an outer diameter of 16.19 cm. The bottom and side legs of the cross are closed by a 3.17-cm-thick piece of Plexiglas. From the center of the intersection the bottom and side legs are 91.44 cm long, and the top leg is 121.92 cm long. The cross is filled with a UO_2F_2 solution (material 1) to a height of 28.93 cm above the center of the cylinder intersection. The cross is then surrounded by a water reflector (material 2) that extends from the center of the intersection: 111.74 cm in the ±x directions, 20.64 cm in the ±y directions, 29.03 cm in the +z direction, and -118.428 cm in the -z direction. A schematic of the assembly is shown in Fig. F17.5.5.

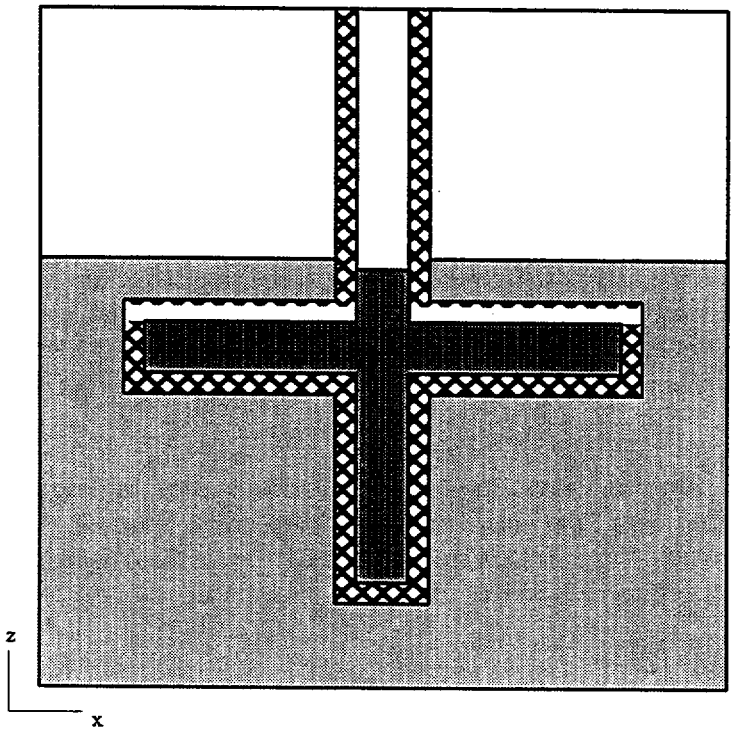


Figure F17.5.5 Plexiglas UO_2F_2-filled cross

Data description of Example 13.

```
READ GEOMETRY
GLOBAL UNIT 1
CYLINDER  10 13.335 28.93 -88.27
CYLINDER  20 13.335 121.92 -88.27
CYLINDER  30 16.19 121.92 -91.44
CYLINDER  40 13.335 88.27 -88.27 ROTATE A1=90. A2=90.
CYLINDER  50 16.19 91.44 -91.44 ROTATE A1=90. A2=90.
```

CUBOID 60 2P111.74 2P20.64 29.03 -118.428
CUBOID 70 2P111.74 2P20.64 121.92 -118.428
MEDIA 1 1 10
MEDIA 0 1 20 -10
MEDIA 3 1 30 -20 -40
MEDIA 1 1 40 -10
MEDIA 3 1 50 -40 -10 -30
MEDIA 2 1 60 -30 -50
MEDIA 0 1 70 -30 -60
BOUNDARY 70
END GEOMETRY

EXAMPLE 14. Assume a Y-shaped aluminum cylinder (material 2) with a 13.95-cm inner radius and a 0.16-cm wall thickness is filled with a UO_2F_2 solution (material 1). From the center where the Y intersects the cylinder, the bottom leg is 76.7 cm long, the top leg is 135.4 cm long, and the Y leg is 126.04 cm long canted at a 29.26-degree angle. The bottom of the bottom leg and the top of the Y leg are sealed with 1.3-cm caps. The Y cylinder is filled to a height of 52.8 cm above the center where the Y leg intersects the vertical cylinder. The cylinder is surrounded by a water reflector (material 3) that extends out 37.0 cm in the ±x direction, 100.0 and −37.0 cm in the ±y direction, and 135.4 and -99.6 in the ±z direction. A schematic of the assembly is shown in Fig. F17.5.6.

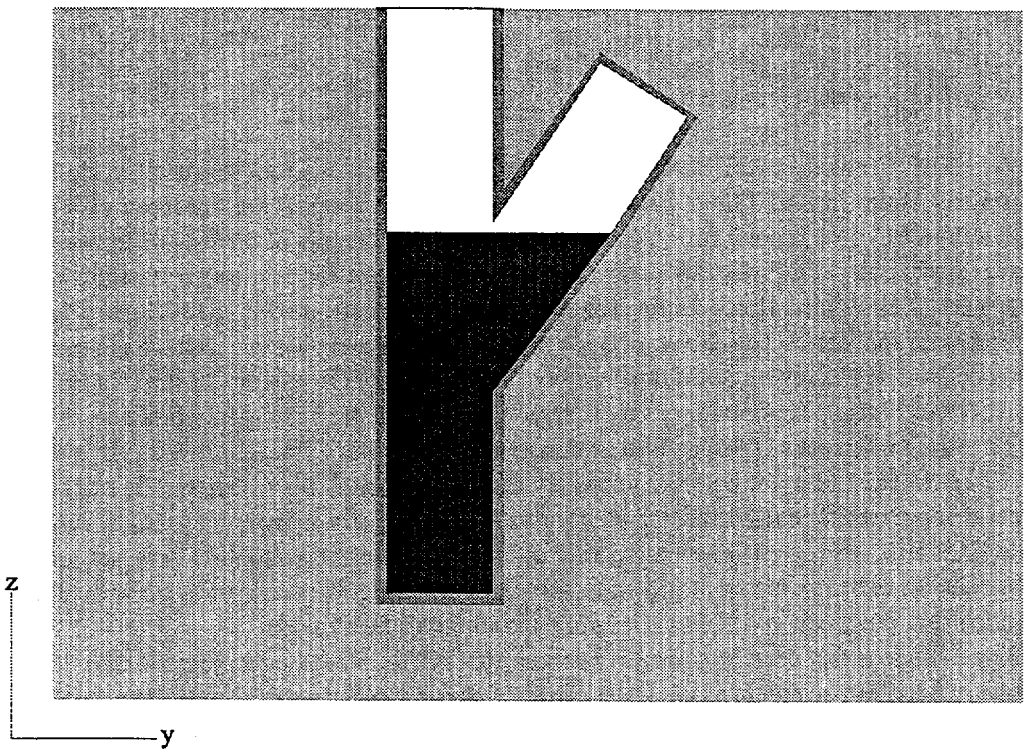


Figure F17.5.6 Y-shaped UO_2F_2-filled aluminum cylinder

Data description of Example 14.

```
READ GEOMETRY
GLOBAL UNIT 1
COM='30 DEG Y CYLINDER'
CYLINDER 10 13.95 135.4 -75.4
CYLINDER 20 14.11 135.4 -76.7
CYLINDER 30 13.95 124.74 0.0  ROTATE A2=-29.26
CYLINDER 40 14.11 126.04 0.0 ROTATE A2=-29.26
CUBOID 50 2P37.0 100.0 -37.0 52.8 -75.4
CUBOID 60 2P37.0 100.0 -37.0 135.4 -99.6
MEDIA 1 1 10 50
MEDIA 2 1 20 -10 -30
MEDIA 1 1 30 50 -10
MEDIA 2 1 40 -30 -20 -10
MEDIA 0 1 10 -50
MEDIA 0 1 30 -50 -10
MEDIA 3 1 60 -20 -40
BOUNDARY 60
END GEOMETRY
```

F17.5.6.1 *Use of Holes in the Geometry*

Section F17.5.6 tells how KENO-VI geometry allows regions in a unit to intersect, thus eliminating the need for holes. Holes in KENO-VI, however, can be useful in simplifying the input of a problem and decreasing the total CPU time needed for a problem. 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 unit where another unit is to be placed. The information contained in a hole geometry card is: (1) the geometry word, HOLE, (2) the unit number of the unit to be placed, and (3) the region definition vector. A hole is placed inside the geometry region defined by the region definition vector. The boundary dimensions of the unit being placed in the hole must precisely match the boundary of the region containing the hole. Only one hole is allowed in a region and holes cannot intersect. Holes can be nested to any depth (see Sect. F17.5.6.2). It is not advisable to use holes that are tangent to other holes or geometry regions, although it is theoretically possible to do so. Frequently holes that are exactly tangent to each other or to other geometry regions may fail to run because the computer code finds that the regions are intersecting due to precision and roundoff. It is not uncommon for a problem that runs on one computing platform to fail on another platform using the same data. Therefore, it is recommended that tangency and boundaries shared with holes be avoided.

Tracking in regions that contain holes can be more efficient than tracking in regions that do not contain holes because every boundary region in a unit must be checked for a crossing whenever a crossing is possible. Putting small but complex geometries in a hole will lessen the number of boundaries that need to be checked for possible crossings. However, the indiscriminant use of holes is not advised since the particle must change

coordinate systems every time a hole is entered or exited. Therefore, holes should be used carefully only when the system can be simplified significantly by their use.

EXAMPLE 15. One example of a unit that requires holes in KENO-V.a but is better described not using holes in KENO-VI as shown in Fig. F17.5.7, representing nine close-packed rods in an annulus. 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.

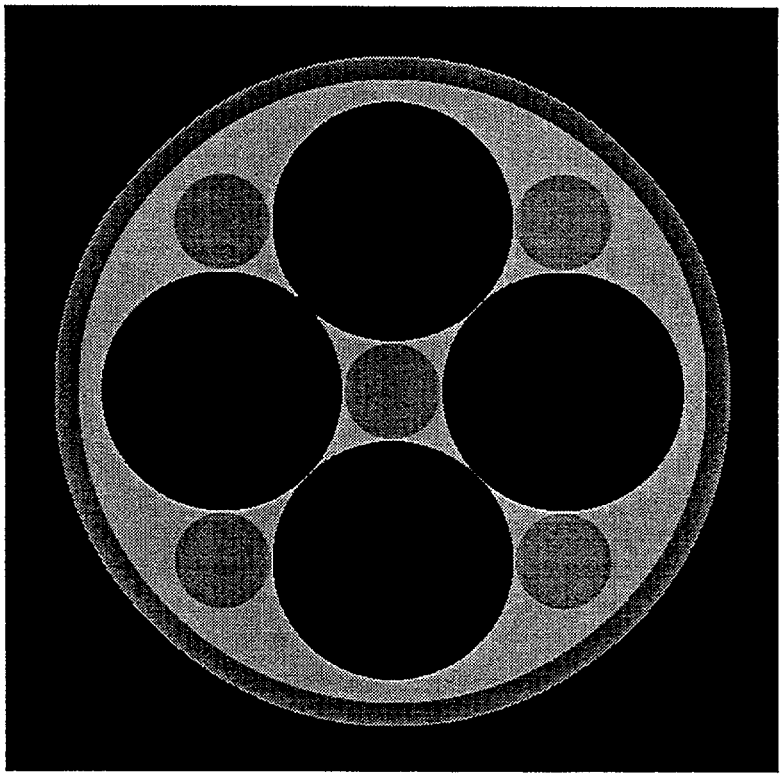


Figure F17.5.7 Close-packed rods in an annulus

Data description of Example 15.

```
READ GEOM
GLOBAL UNIT 1
CYLINDER 1 0.6 2P15.0
CYLINDER 2 0.6 2P15.0 ORIGIN X=2.0 Y=-2.0
CYLINDER 3 0.6 2P15.0 ORIGIN X=2.0  Y=2.0
CYLINDER 4 0.6 2P15.0 ORIGIN X=-2.0 Y=2.0
```

CYLINDER 5 0.6 2P15.0 ORIGIN X=-2.0 Y=-2.0
CYLINDER 6 1.4 2P15.0 ORIGIN X=2.0
CYLINDER 7 1.4 2P15.0 ORIGIN Y=2.0
CYLINDER 8 1.4 2P15.0 ORIGIN X=-2.0
CYLINDER 9 1.4 2P15.0 ORIGIN Y=-2.0
CYLINDER 10 3.6 2P15.0
CYLINDER 11 3.8 2P15.0
CUBOID 12 4P4.0 2P16.0
MEDIA 1 1 1 -6 -7 -8 -9
MEDIA 1 1 2 -9 -6
MEDIA 1 1 3 -6 -7
MEDIA 1 1 4 -7 -8
MEDIA 1 1 5 -8 -9
MEDIA 3 1 6 -2 -3 -1
MEDIA 3 1 7 -3 -4 -1
MEDIA 3 1 8 -4 -5 -1
MEDIA 3 1 9 -5 -2 -1
MEDIA 0 1 10 -1 -2 -3 -4 -5 -6 -7 -8 -9
MEDIA 2 1 11 -10
MEDIA 0 1 12 -11
BOUNDARY 12
END GEOM

The first CYLINDER description represents the middle small rod. The next four cylinder cards describe the four remaining small rods surrounding the middle rod. The ORIGIN modification card is used to shift the origin of each cylinder to the appropriate location. The following four cylinder cards represent the four large rods. Again the ORIGIN modification card is used to shift the origin of each cylinder to the appropriate location. Only the nonzero dimensions need to be entered in the ORIGIN data. The tenth cylinder card is the void in the annulus that contains the rods. The last cylinder card defines the outer surface of the annulus. Finally, the cuboid card describes the surrounding unit boundary.

An array of the unit shown in Fig. F17.5.7 can be easily represented using array description data. For example, a 5 × 3 × 2 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 defining the global unit as the one containing the array and utilizing the following array data:

READ ARRAY NUX=5 NUY=3 NUZ=2 FILL F1 END FILL END ARRAY

or

*READ ARRAY NUX=5 NUY=3 NUZ=2 FILL 30*1 END FILL END ARRAY*

or

READ ARRAY NUX=5 NUY=3 NUZ=2 LOOP 1 1 5 1 1 3 1 1 2 1 END LOOP END ARRAY

F17.5.6.2 *Nesting Holes*

This section illustrates how holes are nested. Holes can be nested to any level. Consider the configuration that was illustrated in Fig. F17.5.7 and replace the large rods with a complicated geometric arrangement. The resultant figure is shown in Fig. F17.5.8. Figure F17.5.9 shows the complicated geometric arrangement that replaced the large rods of Fig. F17.5.7. Figure F17.5.10 shows a component of the arrangement shown in Fig. F17.5.9.

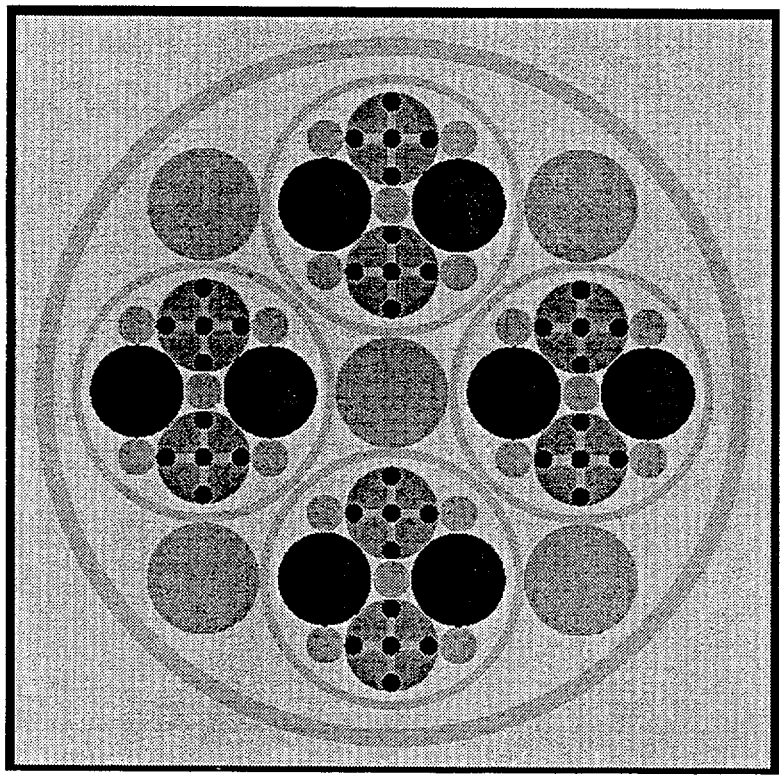


Figure F17.5.8 Configuration using nested holes

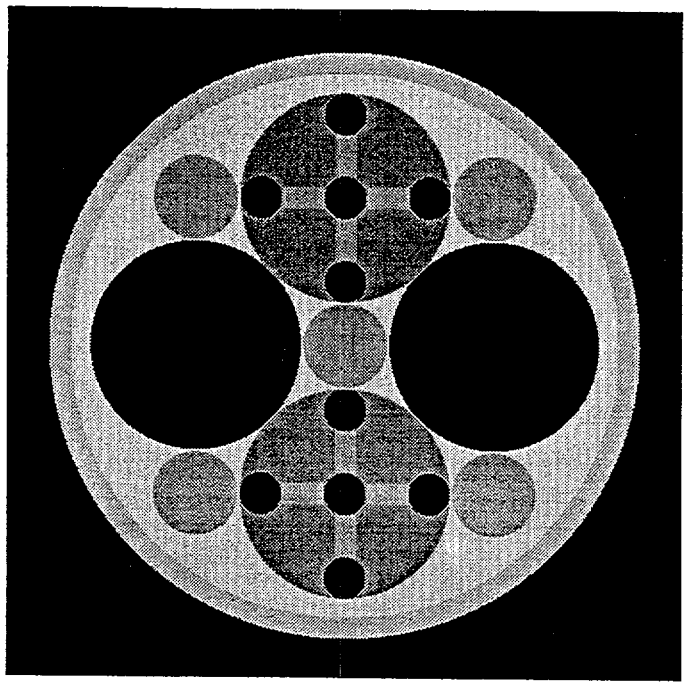


Figure F17.5.9 Complicated geometric arrangement represented by Unit 2

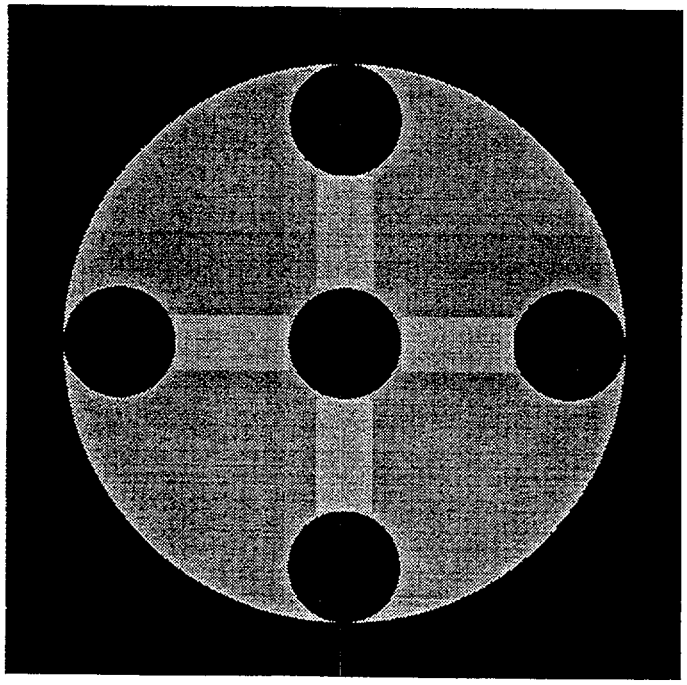


Figure F17.5.10 Geometric component represented by Unit 1

EXAMPLE 16. 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. F17.5.8, it may be most convenient to start the geometry mock-up at the deepest nesting level, as shown in Fig. F17.5.10. The small cylinders are composed of mixture 1, and are each 0.1 cm in radius and 30 cm long. Five small cylinders are used in the schematic shown in Fig. F17.5.10. Their centers are located at (0,0,0) for the central one, at (0,-0.4,0) for the bottom one, at (0.4,0,0) for the right one, at (0,0.4,0) for the top one, and at (-0.4,0,0) 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. A possible geometry mock-up for this system is described as follows:

1. define Unit 1 to contain the five small cylinders and four blocks,

2. define Unit 2 to contain the next two larger-sized cylinders and Unit 1 as holes, and

3. define Global Unit 3 to contain the large cylinders and Unit 2 as holes.

```
UNIT 1
CYLINDER  1 0.1 2P15.0
CYLINDER  2 0.1 2P15.0 ORIGIN Y=-0.4
CYLINDER  3 0.1 2P15.0 ORIGIN X=0.4
CYLINDER  4 0.1 2P15.0 ORIGIN Y=0.4
CYLINDER  5 0.1 2P15.0 ORIGIN X=-0.4
CUBOID  6 2P0.1 2P0.05 2P15.0 ORIGIN X=-0.2
CUBOID  7 2P0.1 2P0.05 2P15.0 ORIGIN X=0.2
CUBOID  8 2P0.05 2P0.1 2P15.0 ORIGIN Y=-0.2
CUBOID  9 2P0.05 2P0.1 2P15.0 ORIGIN Y=0.2
CYLINDER  10 0.5 2P15.0
MEDIA 1 1 1 -6 -7 -8 -9
MEDIA 1 1 2 -8
MEDIA 1 1 3 -7
MEDIA 1 1 4 -9
MEDIA 1 1 5 -6
MEDIA 2 1 6 -1 -5
MEDIA 2 1 7 -1 -3
MEDIA 2 1 8 -1 -2
MEDIA 2 1 9 -1 -4
MEDIA 3 1 -1 -2 -3 -4 -5 -6 -7 -8 -9 10
BOUNDARY 10
```

geometry card **1** places the central rod;
geometry card **2** places the bottom cylinder;
geometry card **3** places the cylinder at the right;
geometry card **4** places the top cylinder;
geometry card **5** places the cylinder at the left;
geometry card **6** places the left cuboid whose length is in x;
geometry card **7** places the right cuboid whose length is in x;
geometry card **8** places the bottom cuboid whose length is in y;
geometry card **9** places the top cuboid whose length is in y, and
geometry card **10** is the surrounding cylinder that defines the system boundary.

Now that Fig. F17.5.10 has been described, consider Fig. F17.5.9. 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 1 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. Unit 2, Fig. F17.5.9, 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 0.707107. This value is from $x^2 + y^2 = 1.0$ (radius unit 4 + radius unit 5), where x and y are equal. The geometry mock-up for this portion of the problem follows:

```
UNIT 2
CYLINDER 1 0.2 2P15.0
CYLINDER 2 0.2 2P15.0 ORIGIN X=0.707107 Y=0.707107
CYLINDER 3 0.2 2P15.0 ORIGIN X=-0.707107 Y=0.707107
CYLINDER 4 0.2 2P15.0 ORIGIN X=-0.707107 Y=-0.707107
CYLINDER 5 0.2 2P15.0 ORIGIN X=0.707107 Y=-0.707107
CYLINDER 6 0.5 2P15.0 ORIGIN X=0.707107
CYLINDER 7 0.5 2P15.0 ORIGIN X=-0.707107
CYLINDER 10 1.3 2P15.0
CYLINDER 11 1.4 2P15.0
MEDIA 2 1 1
MEDIA 2 1 2
MEDIA 2 1 3
MEDIA 2 1 4
MEDIA 2 1 5
MEDIA 1 1 6
MEDIA 1 1 7
HOLE 1 ORIGIN Y=0.707107
HOLE 1 ORIGIN Y=-0.707107
MEDIA 0 1 10 -1 -2 -3 -4 -5 -6 -7
MEDIA 4 1 11 -10
BOUNDARY 11
```

In unit 2, cylinder **1** places a small cylinder of mixture 2 at the origin,
cylinder **2** places the small cylinder of mixture 2 in the upper-right quadrant,
cylinder **3** places the small cylinder of mixture 2 in the upper-left quadrant,
cylinder **4** places the small cylinder of mixture 2 in the lower-left quadrant,
cylinder **5** places the small cylinder of mixture 2 in the lower-right quadrant,
cylinder **6** places the larger cylinder of mixture 1 at the right with its origin at (0.707107,0.0,0.0),
cylinder **7** places the larger cylinder of mixture 1 at the left with its origin at (0.0,0.707107.0.0),
place the top cylinder that contains the geometric component defined in Unit 1,
place the bottom cylinder that contains the geometric component defined in Unit 1,
cylinder **10** defines the inner surface of the annulus, and
cylinder **11** defines the outer surface of the annulus and the unit boundary.

To complete the geometry mock-up, consider Fig. F17.5.8. Define Unit 3 to be the central rod and four outer rods of 0.6 cm radius and 30.0 cm length, and the large annulus of 3.6 cm inner radius, 3.8 cm outer radius, and 30 cm length centered in a cuboid having an 8-cm-square cross section and a length of 32 cm.

```
GLOBAL UNIT 3
CYLINDER 1 0.6 2P15.0
CYLINDER 2 0.6 2P15.0 ORIGIN X=2.0 Y=2.0
CYLINDER 3 0.6 2P15.0 ORIGIN X=-2.0 Y=2.0
CYLINDER 4 0.6 2P15.0 ORIGIN X=-2.0 Y=-2.0
CYLINDER 5 0.6 2P15.0 ORIGIN X=2.0 Y=-2.0
CYLINDER 10 3.6 2P15.0
CYLINDER 11 3.8 2P15.0
CUBOID  12  4P4.0 2P16.0
MEDIA 2 1 1
MEDIA 2 1 2
MEDIA 2 1 3
MEDIA 2 1 4
MEDIA 2 1 5
HOLE 2 ORIGEN X=2
HOLE 2 ORIGEN Y=2
HOLE 2 ORIGEN X=-2
HOLE 2 ORIGEN Y=-2
MEDIA 0 1 10 -1 -2 -3 -4
MEDIA 4 1 11 -10
MEDIA 0 1 12 -11
BOUNDARY  12
```

In unit 3, cylinder **1** defines the rod of mixture 2, centered in the annulus,
cylinder **2** places a rod of mixture 2 in the upper-right quadrant of the annulus,
cylinder **3** places a rod of mixture 2 in the upper-left quadrant of the annulus,
cylinder **4** places a rod of mixture 2 in the lower-left quadrant,
cylinder **5** places a rod of mixture 2 in the lower-right quadrant,

the composite annulus of Unit 2 is placed to the right of the central rod,
the composite annulus of Unit 2 is placed above the central rod,
the composite annulus of Unit 2 is placed to the left of the central rod,
the composite annulus of Unit 2 is placed below the central rod,
cylinder **10** defines the void volume between the central rod and the annulus,
cylinder **11** defines the outer surface of the annulus, and
cuboid **12** defines the unit boundary

This problem illustrates two levels of hole nesting. The total input data for the problem is given below. The nuclide IDs 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. F17.5.8. This problem includes the data for a character 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. F17.5.11a, and the right half is given in Fig. F17.5.11b. 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.

Data description of Example 16.

```
=KENOVI
NESTED HOLES SAMPLE
READ PARAM  RUN=NO LIB=41 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 0.1 2P15.0
CYLINDER 2 0.1 2P15.0 ORIGIN Y=-0.4
CYLINDER 3 0.1 2P15.0 ORIGIN X=0.4
CYLINDER 4 0.1 2P15.0 ORIGIN Y=0.4
CYLINDER 5 0.1 2P15.0 ORIGIN X=-0.4
CUBOID 6 2P0.1 2P0.05 2P15.0 ORIGIN X=-0.2
CUBOID 7 2P0.1 2P0.05 2P15.0 ORIGIN X=0.2
CUBOID 8 2P0.05 2P0.1 2P15.0 ORIGIN Y=-0.2
CUBOID 9 2P0.05 2P0.1 2P15.0 ORIGIN Y=0.2
CYLINDER 10 0.5 2P15.0
MEDIA 1 1 1 -6 -7 -8 -9
MEDIA 1 1 2 -8
MEDIA 1 1 3 -7
MEDIA 1 1 4 -9
```

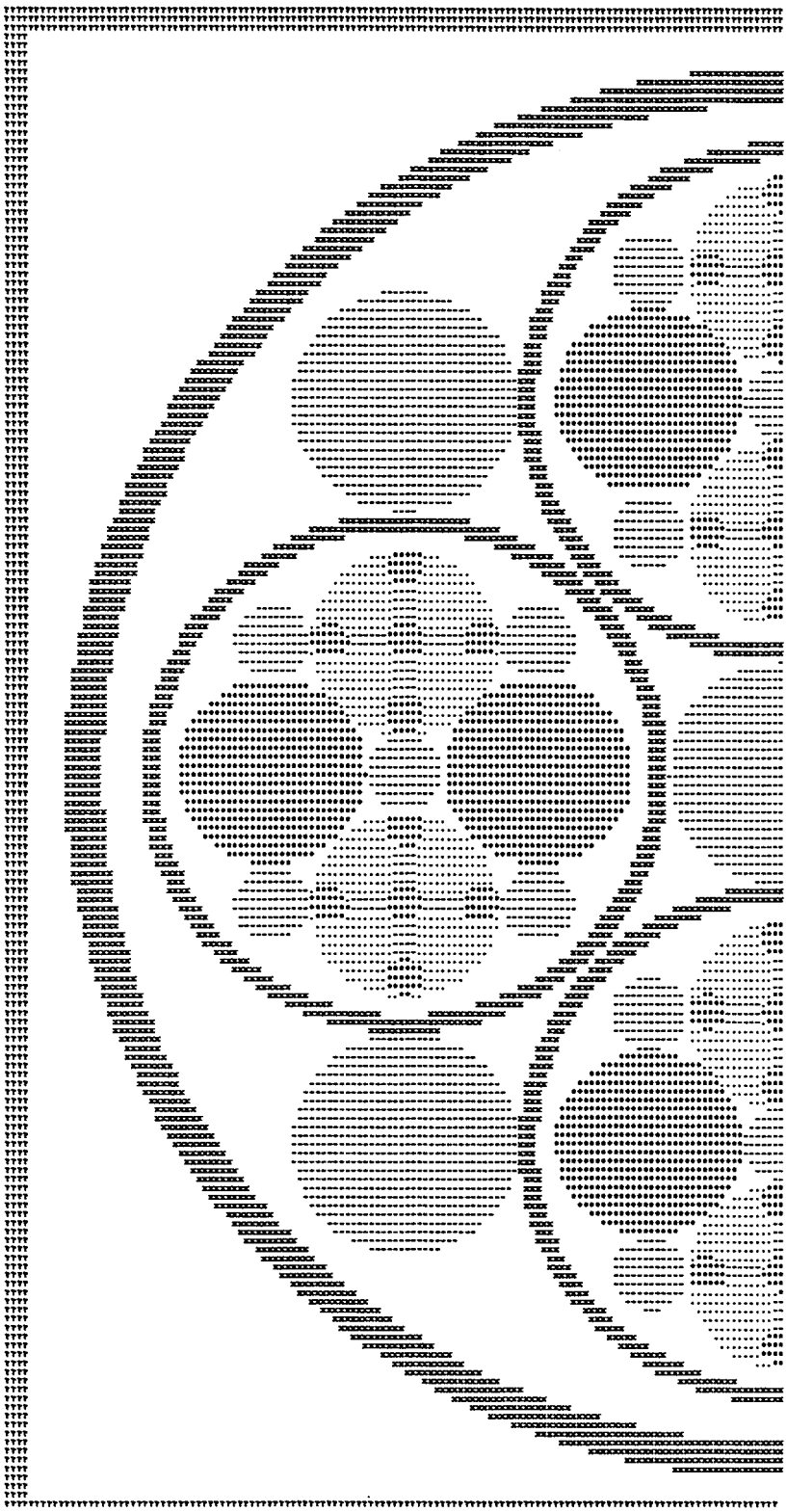


Figure F17.5.11a Left half of printer plot

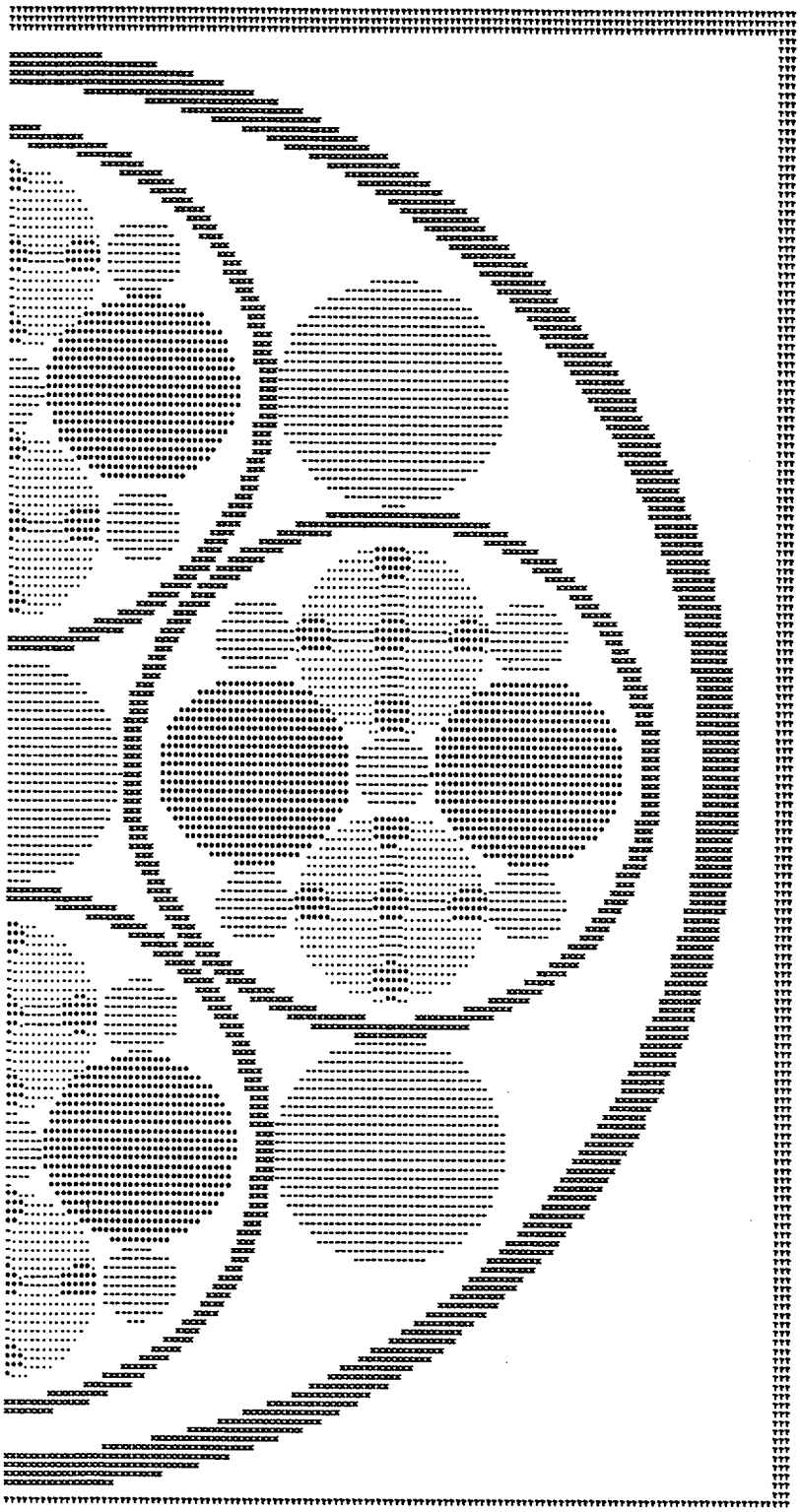


Figure F17.5.11b Right half of printer plot

```
MEDIA 1 1 5 -6
MEDIA 2 1 6 -1 -5
MEDIA 2 1 7 -1 -3
MEDIA 2 1 8 -1 -2
MEDIA 2 1 9 -1 -4
MEDIA 3 1 -1 -2 -3 -4 -5 -6 -7 -8 -9 10
BOUNDARY 10
UNIT 2
CYLINDER 1 0.2 2P15.0
CYLINDER 2 0.2 2P15.0 ORIGIN X=0.707107 Y=0.707107
CYLINDER 3 0.2 2P15.0 ORIGIN X=-0.707107 Y=0.707107
CYLINDER 4 0.2 2P15.0 ORIGIN X=-0.707107 Y=-0.707107
CYLINDER 5 0.2 2P15.0 ORIGIN X=0.707107 Y=-0.707107
CYLINDER 6 0.5 2P15.0 ORIGIN X=0.707107
CYLINDER 7 0.5 2P15.0 ORIGIN X=-0.707107
CYLINDER 10 1.3 2P15.0
CYLINDER 11 1.4 2P15.0
MEDIA 2 1 1
MEDIA 2 1 2
MEDIA 2 1 3
MEDIA 2 1 4
MEDIA 2 1 5
MEDIA 1 1 6
MEDIA 1 1 7
HOLE 1 ORIGIN Y=0.707107
HOLE 1 ORIGIN Y=-0.707107
MEDIA 0 1 10 -1 -2 -3 -4 -5 -6 -7
MEDIA 4 1 11 -10
BOUNDARY 11
GLOBAL UNIT 3
CYLINDER 1 0.6 2P15.0
CYLINDER 2 0.6 2P15.0 ORIGIN X=2.0 Y=2.0
CYLINDER 3 0.6 2P15.0 ORIGIN X=-2.0 Y=2.0
CYLINDER 4 0.6 2P15.0 ORIGIN X=-2.0 Y=-2.0
CYLINDER 5 0.6 2P15.0 ORIGIN X=2.0 Y=-2.0
CYLINDER 10 3.6 2P15.0
CYLINDER 11 3.8 2P15.0
CUBOID 12 4P4.0 2P16.0
MEDIA 2 1 1
MEDIA 2 1 2
MEDIA 2 1 3
MEDIA 2 1 4
MEDIA 2 1 5
HOLE 2 ORIGIN X=2
HOLE 2 ORIGIN Y=2
HOLE 2 ORIGIN X=-2
```

```
HOLE 2 ORIGIN Y=-2
MEDIA 0 1 10 -1 -2 -3 -4 -5
MEDIA 4 1 11 -10
MEDIA 0 1 12 -11
BOUNDARY 12
END GEOM
READ PLOT
TTL='X-Y SLICE AT Z MIDPOINT. NESTED HOLES'
XUL=-4.1 YUL=4.1 ZUL=0.0 XLR=4.1 YLR=-4.1 ZLR=0
UAX=1.0 VDN=-1.0 NAX=260 NCH=' *-.X' SCR=NO END
END PLOT
END DATA
END
```

F17.5.6.3 *Multiple Arrays*

EXAMPLE 17. Section F17.5.6 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. F17.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[1,2] 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. F17.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. The array is considered to be bare and suspended with no supports.

```
UNIT 1
CYLINDER 1 9.525 2P8.89
CYLINDER 2 10.16 2P9.525
CUBOID 3 4P10.875 2P10.24
MEDIA 2 1 1
MEDIA 3 1 2 -1
MEDIA 0 1 3 -2
BOUNDARY 3
```

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 5.748 2P5.3825
CUBOID  2  4P6.59 2P6.225
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
```

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 arrangement is accomplished by defining a Unit 3 which contains array 1, the array of Plexiglas solution containers. The overall dimensions of this unit are 21.75 cm in x, 43.5 cm in y, and 40.96 cm in z. The overall dimensions of the array need to be calculated. An array position needs to be chosen which is used to place the array in the unit. The array is placed in the unit by aligning the origin of the unit in the chosen array position with a calculated position in the surrounding unit. The chosen array position and the X, Y, and Z positions are entered after the keyword PLACE on the ARRAY card. Unit 3 is defined as follows:

```
UNIT 3
CUBOID  1 2P10.875 2P21.75 2P20.48
ARRAY 1 1 PLACE 1 1 1 0.0 -10.875 -10.24
BOUNDARY  1
```

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.

```
UNIT 4
CUBOID 1 2P6.59 2P13.18 2P12.45
CUBOID 2 2P6.59 2P21.75 2P20.48
ARRAY  2  1 PLACE 1 1 1 0.0 -6.59 -6.225
MEDIA 0 1 2 -1
BOUNDARY 2
```

Now that Unit 3 and Unit 4 have been defined, they must be placed in the global or universe array contained in the global unit. The global unit is specified by putting the keyword GLOBAL prior to the word unit.

```
GLOBAL UNIT 5
CUBOID 1 34.93 0.0 43.5 0.0 40.96 0.0
ARRAY 3 1 PLACE 1 1 1 6.59 21.75 20.48
BOUNDARY 1
```

The array of arrays is defined as the global array in the array data as follows:

```
GBL=3 ARA=3 NUX=2 NUY=1 NUZ=1 FILL 4 3 END FILL
```

This discussion completes the geometry description for the problem. The complete input description for the problem is given below. The nuclide IDs are for the 16-group Hansen-Roach working-format library.

Data description of Example 17.

```
=KENO-VI
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 1 9.525 2P8.89
CYLINDER 2 10.16 2P9.525
CUBOID 3 4P10.875 2P10.24
MEDIA 2 1 1
MEDIA 3 1 2 -1
MEDIA 0 1 3 -2
BOUNDARY 3
UNIT 2
CYLINDER 1 5.748 2P5.3825
CUBOID 2 4P6.59 2P6.225
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
UNIT 3
CUBOID 1 2P10.875 2P21.75 2P20.48
ARRAY 1 1 PLACE 1 1 1 0.0 -10.875 -10.24
BOUNDARY 1
```

```
UNIT 4
CUBOID 1 2P6.59 2P13.18 2P12.45
CUBOID 2 2P6.59 2P21.75 2P20.48
ARRAY  2  1 PLACE 1 1 1 0.0 -6.59 -6.225
MEDIA 0 1 2 -1
BOUNDARY
GLOBAL UNIT 5
CUBOID 1 34.93 0.0 43.5 0.0 40.96 0.0
ARRAY 3 1 PLACE 1 1 1 6.59 21.75 20.48
BOUNDARY  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 plot of an x-y slice taken through the bottom layer of the array is shown in Fig. F17.5.12. A plot of an x-z slice taken through the left half of the array is shown in Fig. F17.5.13. These plots were used to verify the geometry mock-up.

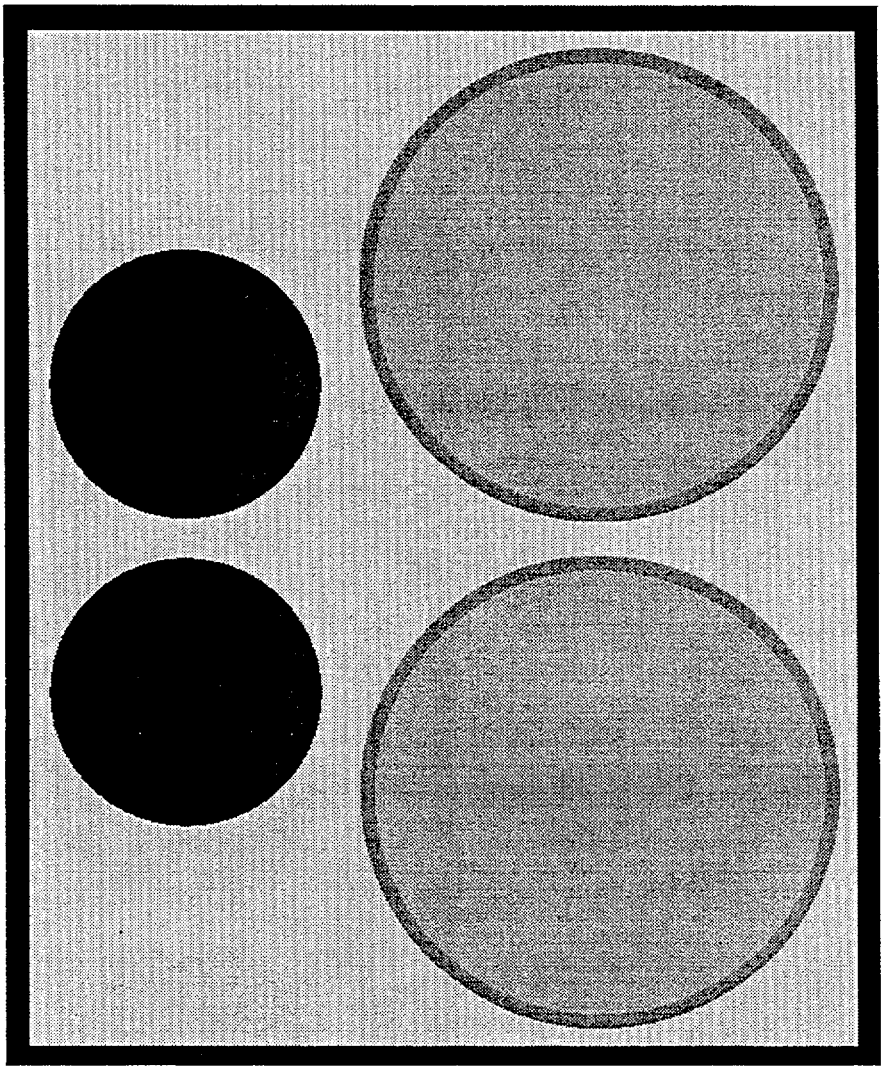


Figure F17.5.12 x-y plot of mixed array

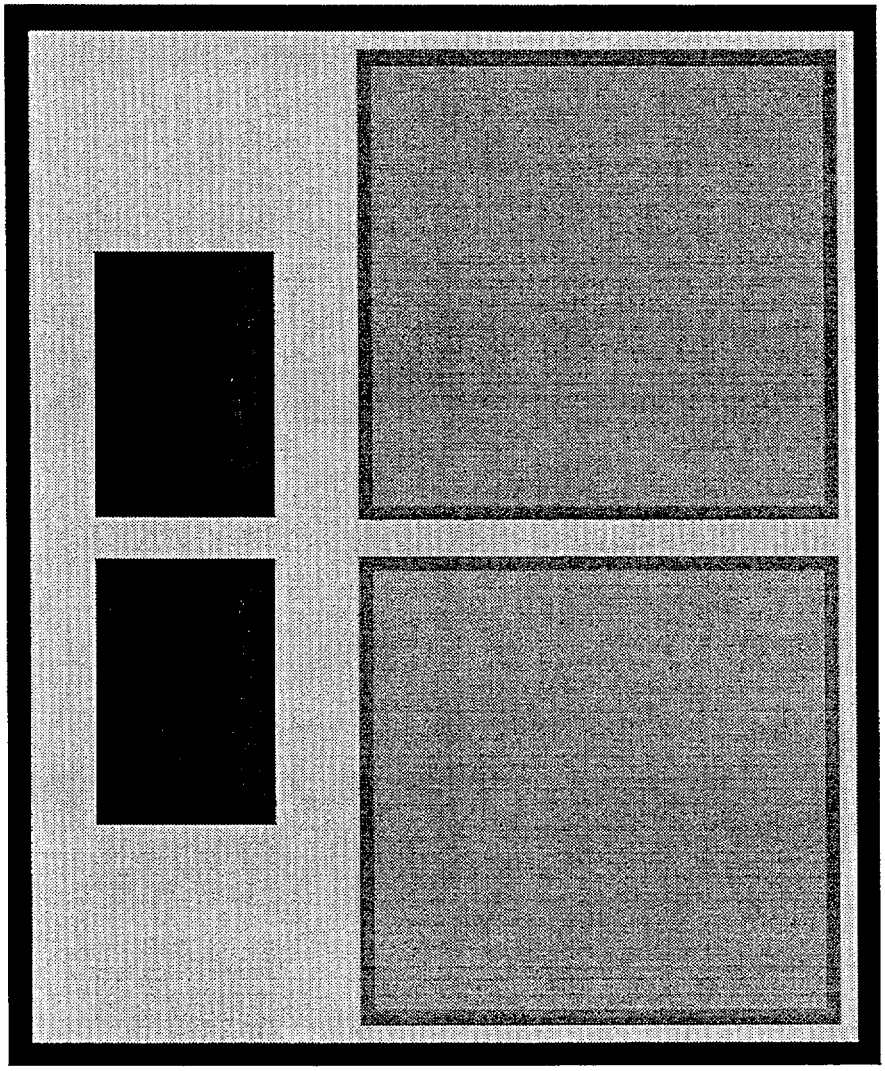


Figure F17.5.13 An x-z plot of mixed array

EXAMPLE 18. Consider a storage array of highly enriched uranium buttons, each 2.54 cm (1 in.) tall and 10.16 cm (4 in.) in diameter. These buttons are stored on stainless steel shelves with a center-to-center spacing of 60.96 cm (2 ft) between them. The shelves are 0.64 cm (1/4 in.) thick, 45.72 cm (18 in.) wide, 609.6 cm (20 ft) long, and are 45.72 cm (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 15.24 cm (6 in.) above the floor. The storage room is 586.56 cm (19.5 ft) in the x direction by 1293.44 cm (43 ft) in the y direction, with 365.76 cm (12-ft) ceilings in the z direction. The walls, ceiling, and floor are composed of 30.48-cm-(1-ft-)thick concrete. All the aisles between the storage racks are 91.44 cm (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 been chosen arbitrarily 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 5.08  2.54  0.0
CUBOID  2  2P22.86  2P30.48  45.72  0.0
CUBOID  3  2P22.86  2P30.48  45.72  -.635
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 2 1 3 -2 -1
BOUNDARY 3
```

Array 1 creates an array of these buttons that fills one shelf. Unit 2 then contains one of the shelves shown in Fig. F17.5.14.

```
ARA=1 NUX=1 NUY=10 NUZ=1 FILL F1 END FILL
```

```
UNIT 2
CUBOID 1 45.72 0.0 609.60 0.0 46.355 0.0
ARRAY 1 1 PLACE 1 1 1 22.86 30.48 0.635
BOUNDARY 1
```

Stack four Unit 2's vertically to obtain one of the racks shown in Fig. F17.5.14. 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
CUBOID 1 45.72 0.0 609.60 0.0 185.42 0.0
ARRAY 2 1 PLACE 1 1 1 3*0.0
BOUNDARY 1
UNIT 4
CUBOID 1 2P22.86 2P45.72 185.42 0.0
 MEDIA 0 1 1
BOUNDARY 1
```

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 configuration is shown in Fig. F17.5.14.

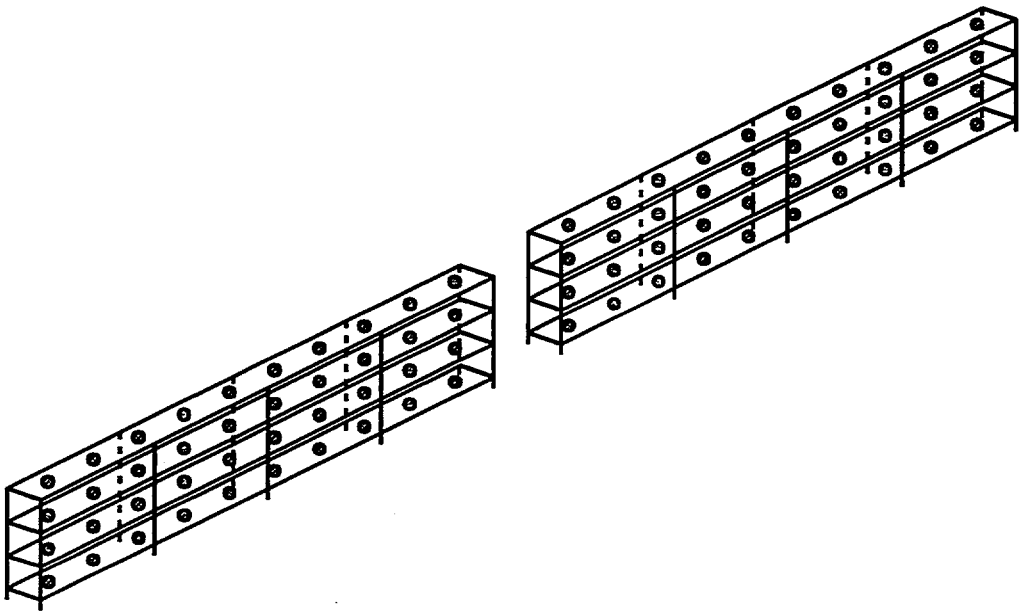


Figure F17.5.14 Two racks of uranium buttons

ARA=3 NUX=1 NUY=3 NUZ=1 FILL 3 4 3 END FILL

UNIT 5
CUBOID 1 45.72 0.0 1310.64 0.0 185.42 0.0
*ARRAY 3 1 PLACE 1 1 1 3*0.0*
BOUNDARY 1

Create a Unit 6, which is an aisle 91.44 cm (3 ft) wide in the x direction and 1310.64 cm (43 ft) in the y direction (full length of the room).

UNIT 6
CUBOID 1 91.44 0.0 1310.64 0.0 185.42 0.0
MEDIA 0 1 1
BOUNDARY 1

Stack Units 5 and 6 in the x direction to achieve the array of racks in the room. Then put the 15.24-cm (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. F17.5.15.

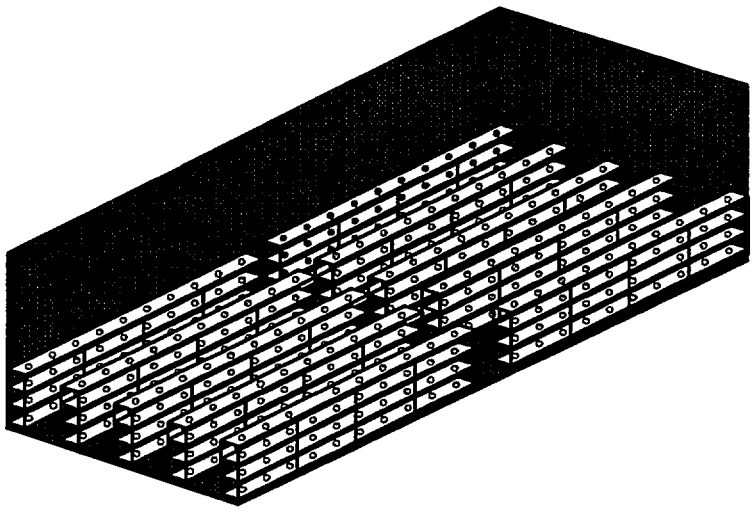


Figure F17.5.15 Entire storage array in the room

```
GBL=4 ARA=4 NUX=9 NUY=1 NUZ=1 FILL 5 6 3Q2 5 END FILL

GLOBAL UNIT 7
CUBOID 1 594.36     0.0 1310.64     0.0 185.42     0.0
CUBOID 2 594.36     0.0 1310.64     0.0 350.52 -15.24
CUBOID 3 599.36  -5.00 1315.64  -5.00 355.52 -20.24
CUBOID 4 604.36 -10.00 1320.64 -10.00 360.52 -25.24
CUBOID 5 609.36 -15.00 1325.64 -15.00 365.52 -30.24
CUBOID 6 614.36 -20.00 1330.64 -20.00 370.52 -35.24
CUBOID 7 619.36 -25.00 1335.64 -25.00 375.52 -40.24
CUBOID 8 624.36 -30.00 1340.64 -30.00 380.52 -45.24
CUBOID 9 624.84 -30.48 1341.12 -30.48 381.00 -45.72
ARRAY 4 1 PLACE 1 1 1 3*0.0
MEDIA 0 1 2 -1
MEDIA 3 2 3 -2 -1
MEDIA 3 3 -3
MEDIA 3 4 5 -4
MEDIA 3 5 6 -5
MEDIA 3 6 7 -6
MEDIA 3 7 8 -7
MEDIA 3 8 9 -8
BOUNDARY 9
```

The final mock-up for this room is given below: The 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 plots for this system are not included as figures. The user can generate the plots if it is desirable to see them. The nuclide IDs used in this problem are for the 16-group Hansen-Roach working-format library.

Data description of Example 18.

```
=KENOVI
STORAGE ARRAY
READ PARAMETERS  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  5.08  2.54  0.0
CUBOID  2  2P22.86  2P30.48  45.72  0.0
CUBOID  3  2P22.86  2P30.48  45.72  -.635
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 2 1 3 -2 -1
BOUNDARY 3
```

```
UNIT 2
CUBOID 1 45.72 0.0 609.60 0.0 46.355 0.0
ARRAY 1 1 PLACE 1 1 1 22.86 30.48 0.635
BOUNDARY 1
UNIT 3
CUBOID 1 45.72 0.0 609.60 0.0 185.42 0.0
ARRAY 2 1 PLACE 1 1 1 3*0.0
BOUNDARY 1
UNIT 4
CUBOID 1 2P22.86 2P45.72 185.42 0.0
MEDIA 0 1 1
BOUNDARY 1
UNIT 5
CUBOID 1 45.72 0.0 1310.64 0.0 185.42 0.0
ARRAY 3 1 PLACE 1 1 1 3*0.0
BOUNDARY 1
UNIT 6
CUBOID 1 91.44 0.0 1310.64 0.0 185.42 0.0
MEDIA 0 1 1
BOUNDARY 1
GLOBAL UNIT 7
CUBOID 1 594.36 0.0 1310.64 0.0 185.42 0.0
CUBOID 2 594.36 0.0 1310.64 0.0 350.52 -15.24
CUBOID 3 599.36 -5.00 1315.64 -5.00 355.52 -20.24
CUBOID 4 604.36 -10.00 1320.64 -10.00 360.52 -25.24
CUBOID 5 609.36 -15.00 1325.64 -15.00 365.52 -30.24
CUBOID 6 614.36 -20.00 1330.64 -20.00 370.52 -35.24
CUBOID 7 619.36 -25.00 1335.64 -25.00 375.52 -40.24
CUBOID 8 624.36 -30.00 1340.64 -30.00 380.52 -45.24
CUBOID 9 624.84 -30.48 1341.12 -30.48 381.00 -45.72
ARRAY 4 1 PLACE 1 1 1 3*0.0
MEDIA 0 1 2 -1
MEDIA 3 2 3 -2 -1
MEDIA 3 3 4 -3
MEDIA 3 4 5 -4
MEDIA 3 5 6 -5
MEDIA 3 6 7 -6
MEDIA 3 7 8 -7
MEDIA 3 8 9 -8
BOUNDARY 9
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=5 NBX=5 END START
READ PLOT PLT=YES 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=640 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=640 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=320 END
END PLOT
END DATA
END

F17.5.6.4 *Arrays and Holes*

Sections F17.5.6.1 and F17.5.6.2 describe the use of holes, and Sect. F17.5.6.3 describes multiple arrays and arrays of arrays. Holes can also be used to place arrays at locations in other units. This section contains examples to illustrate the combined use of arrays and holes.

EXAMPLE 19. 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. F17.5.16. The rods are positioned on the floor of the container and have a 0.1-cm gap between their tops and the top of the container. Material 1 is uranium and material 2 is steel.

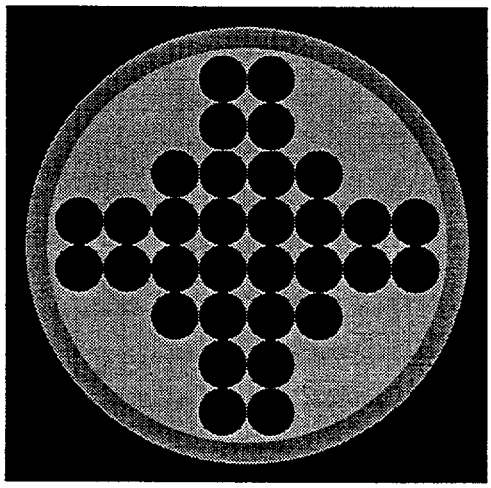


Figure F17.5.16 Uranium rods in a cylindrical container

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 and Unit 2 to be a void cuboid with the same square pitch.

```
UNIT 1
CYLINDER 1 0.5  2P5.0
CUBOID  2  4P0.5  2P5.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
UNIT 2
CUBOID 1 4P0.5 2P5.0
MEDIA 0 1 1
BOUNDARY  1
```

Define array 1 to be the central square 10 × 10 array, consisting of 32 rods and 68 void positions that can be used to represent the array shown in Fig. F17.5.16.

```
ARA=1  NUX=10  NUY=10  NUZ=1  FILL  14*2 1 1 8*2 1 1 7*2 4*1 4*2 8*1 2 2 8*1 4*2 4*1 7*2 1 1
8*2 1 1 14*2  END FILL
```

Now place array 1 in Unit 3. The first cylinder card defines the array boundary. Everything external to this boundary is not considered part of the problem. The positions in the array that do not contain rods are filled with cuboids consisting of void. The array boundary must either coincide with the outer boundary of the array or be contained within the array. An exterior void region is placed around the array boundary to coincide with the size of the interior radius of the container. The 10 × 10 array with the array boundary is shown in Fig. 17.5.16a.

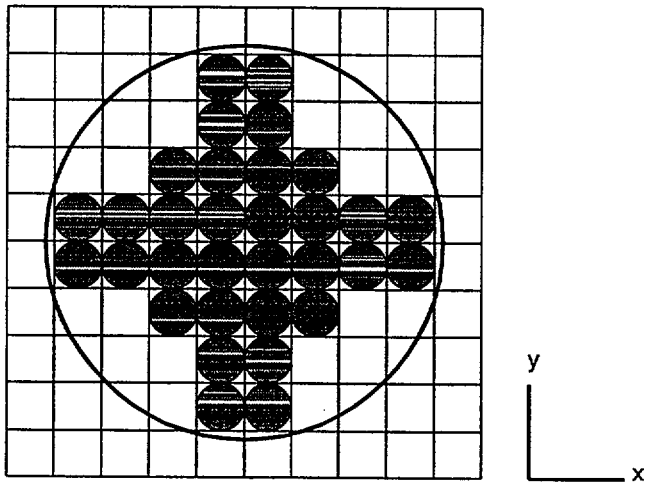


Figure F17.5.17 The 10 × 10 array of uranium 32 rods and 68 void cuboids with the array boundary

```
UNIT 3
CYLINDER 1  4.15 5.0 -5.0
CYLINDER 2 4.15 5.1 -5.0
ARRAY 1 1 PLACE 5 5 1   -0.5 -0.5 -0.0
MEDIA 0 1 2 -1
BOUNDARY  2
```

The unit containing the array is now placed within the global unit using a hole media card. The location of the hole is determined using ORIGIN data to match the origin of the unit in the hole with a X, Y, Z position in the surrounding unit. In this problem the origin of the unit needs to be at position (0,0,0). Since only nonzero data must be entered, ORIGIN data are not needed for this problem. The boundary region consists of the steel container.

```
GLOBAL UNIT 4
CYLINDER 2 4.6 6.37 -6.27
HOLE 3 ORIGIN X=0.0 Y=0.0 Z=0.0
MEDIA 2 1 2
BOUNDARY  2
```

The overall problem description is shown below. Two of the color plots used for verification of this mock-up are shown in Figs. F17.5.18 and F17.5.19.

Data description of Example 19.

```
=KENO VI
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  0.5  2P5.0
CUBOID   2   4P0.5  2P5.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
UNIT 2
CUBOID 1 4P0.5 2P5.0
MEDIA 0 1 1
BOUNDARY  1
UNIT 3
CYLINDER 1  4.15 5.0 -5.0
CYLINDER 2 4.15 5.1 -5.0
ARRAY 1 1 PLACE 5 5 1   -0.5 -0.5 -0.0
```

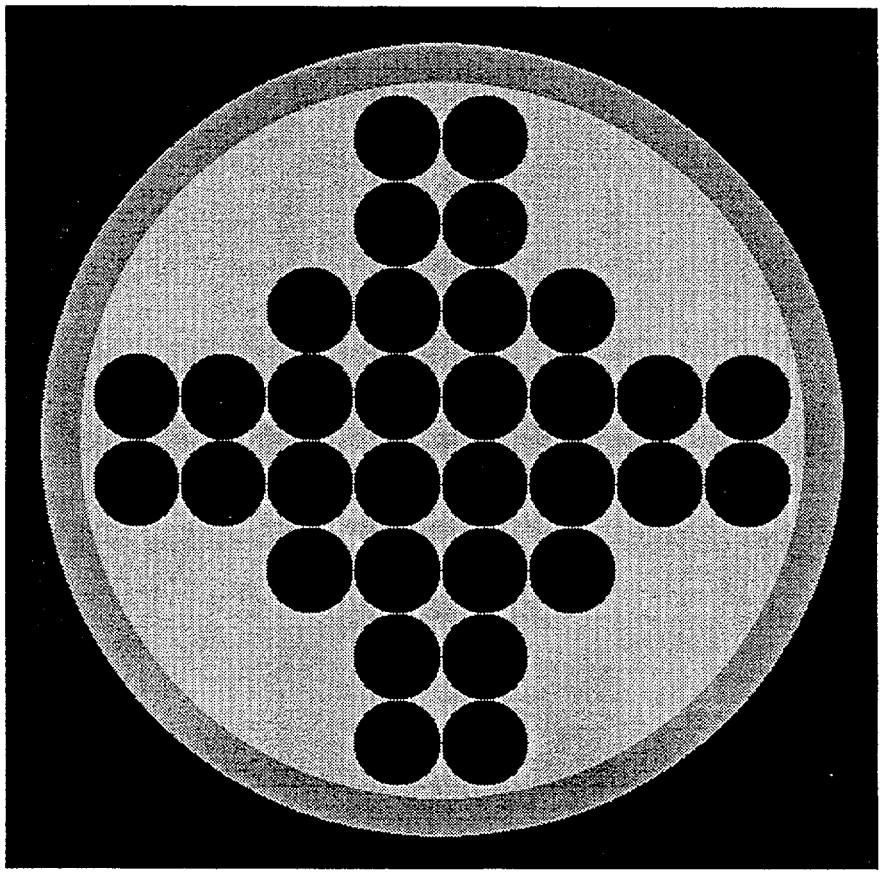


Figure F17.5.18 An x-y slice of uranium rods in a cylindrical container

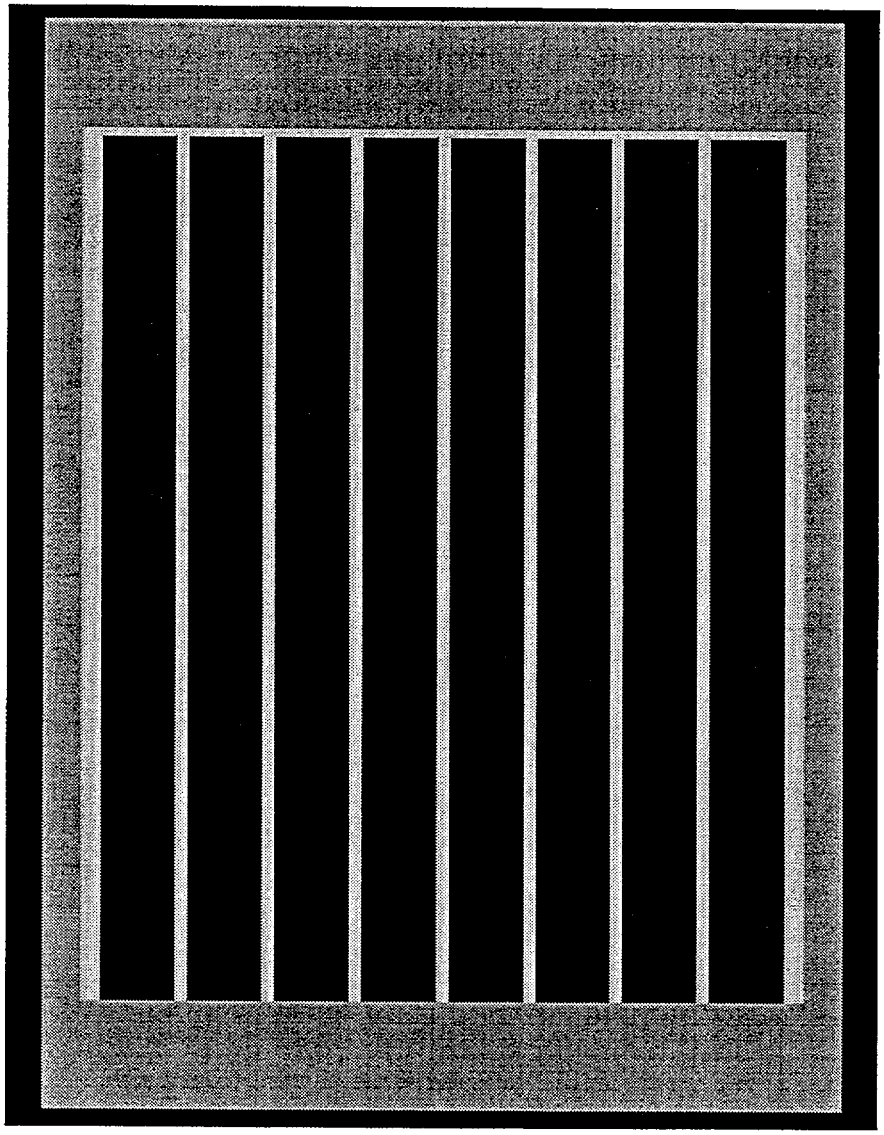


Figure F17.5.19 An x-z slice of uranium rods in a cylindrical container

```
MEDIA 0 1 2 -1
BOUNDARY 2
GLOBAL UNIT 4
CYLINDER 2 4.6 6.37 -6.27
HOLE 3 ORIGIN X=0.0 Y=0.0 Z=0.0
MEDIA 2 1 2
BOUNDARY 2
END GEOM
READ ARRAY
ARA=1 NUX=10 NUY=10 NUZ=1 FILL 14*2 1 1 8*2 1 1 7*2 4*1 4*2 8*1 2 2 8*1 4*2 4*1 7*2 1 1
8*2 1 1 14*2 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=640 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=640 END
END PLOT
END DATA
END
```

EXAMPLE 20. A TYPICAL PWR SHIPPING CASK

Consider a typical PWR shipping cask, illustrated in Fig. F17.5.20. The interior and exterior of the cask is carbon steel (mixture 7) and a depleted uranium gamma shield (mixture 6) is present in the annulus. The shipping cask contains seven PWR fuel assemblies. Each assembly is a 17 × 17 array of fuel rods with water holes as shown. Each assembly is contained in stainless steel (mixture 5). Each fuel rod is clad with Zircaloy (mixture 2) and is composed of 4% enriched UO_2 (mixture 1). Rods of B_4C (mixture 4) clad with stainless steel are positioned between the fuel assemblies. The entire cask is filled with water (mixture 3).

To describe the geometry of the cask, start by defining some simple units, as shown in Fig. F17.5.21. 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 .41148 365.76 0.0
CYLINDER 2 .48133 365.76 0.0
CUBOID 3 .63754 -.63754 .63754 -.63754 365.76 0.0
MEDIA 1 1 1
MEDIA 2 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY 3

UNIT 2
CUBOID 1 .63754 -.63754 .63754 -.63754 365.76 0.0
MEDIA 3 1 1
BOUNDARY 1
```

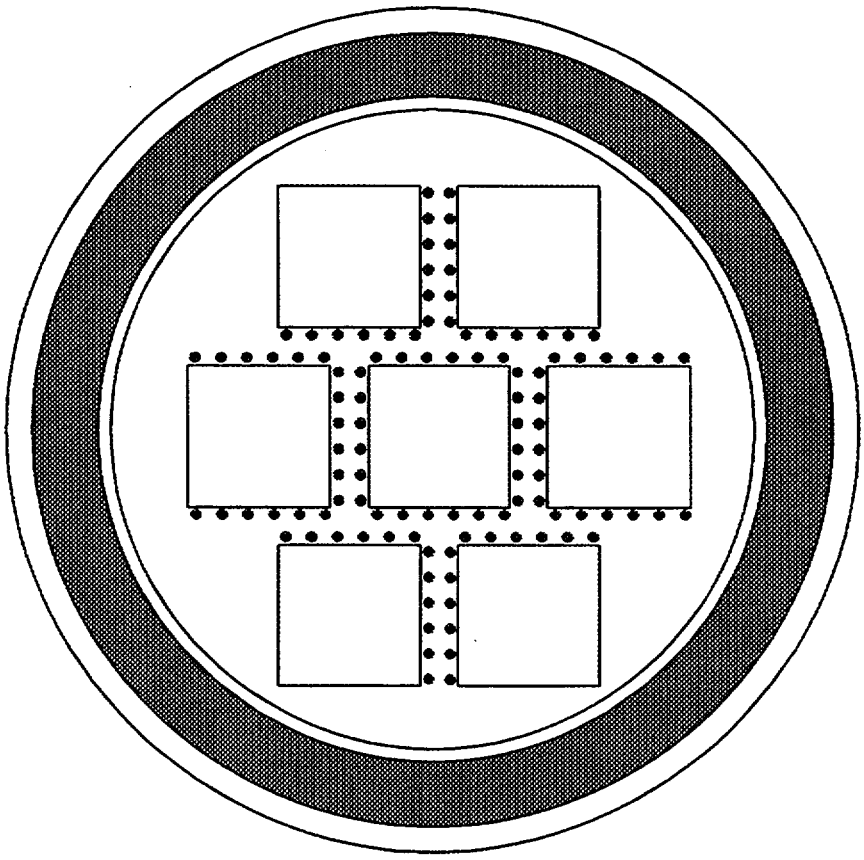


Figure F17.5.20 Typical PWR shipping cask

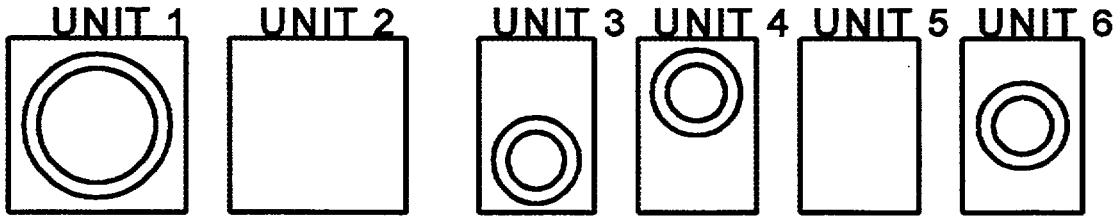


Figure F17.5.21 Simple units

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  1  .584 365.76 0.0
CYLINDER 2   .635 365.76 0.0
CUBOID    3   .9912 -.9912 2.2352 -1.27 365.76 0.0
MEDIA 4 1 1
MEDIA 5 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY  3

UNIT 4
CYLINDER 1  .584 365.76 0.0
CYLINDER 2  .635 365.76 0.0
CUBOID   3   .9912 -.9912 1.2702 -2.235 365.76 0.0
MEDIA 4 1 1
MEDIA 5 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY  3

UNIT 5
CUBOID 1   .9912 -.9912 1.7526 -1.7526 365.76 0.0
MEDIA 3 1 1
BOUNDARY  1

UNIT 6
CYLINDER 1  .584 365.76 0.0
CYLINDER 2  .635 365.76 0.0
CUBOID  3    1.1875215 -1.1875215 1.883706 -1.883706 365.76 0.0
MEDIA 4 1 1
MEDIA 5 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY  3
```

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. F17.5.22. 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. F17.5.23.

```
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
CUBOID  1 10.83818 -10.83818 10.83818 -10.83818 365.76 0.0
CUBOID  2 11.112495 -11.112495 11.112495 -11.112495 365.76 0.0
CUBOID  3 11.302238 -11.302238 11.302238 -11.302238 365.76 0.0
```

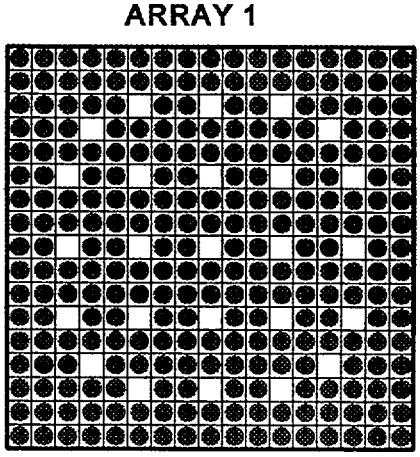


Figure F17.5.22 Fuel pin array

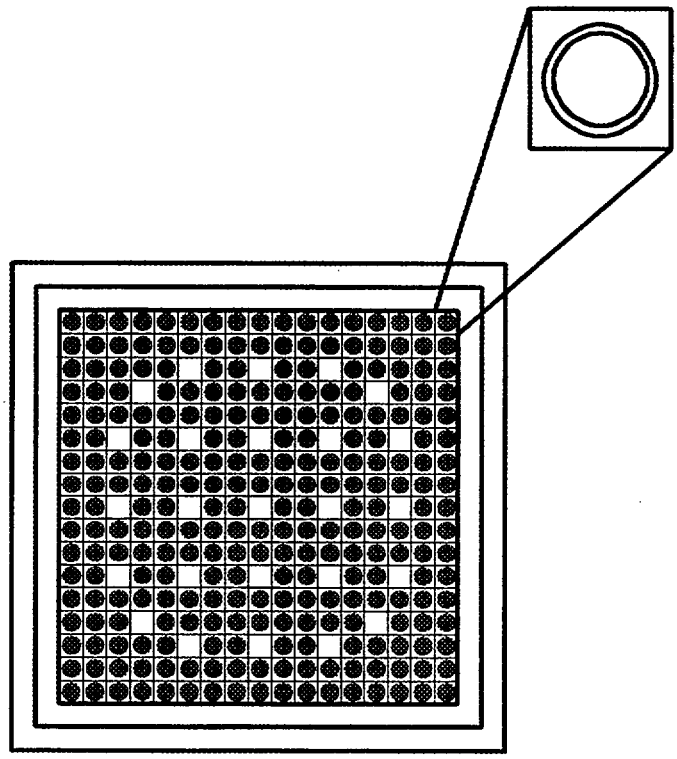


Figure F17.5.23 Fuel assembly

*ARRAY 1 1 PLACE 9 9 1 3*0.0*
MEDIA 3 1 2 -1
MEDIA 5 1 3 -2 -1
BOUNDARY 3

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 placed directly in the problem, as shown in Fig. F17.5.24.

ARA=2 NUX=2 NUY=6 NUZ=1 FILL F6 END FILL

The next step is to create the central array of three fuel assemblies with B_4C rods between then. This configuration is accomplished by placing fuel assemblies (Unit 7) and B_4C rod arrays (Array 2) into a unit (unit 8). The resultant geometry is shown in Fig. F17.5.25.

UNIT 8
CUBOID 4 -11.302238 -16.052324 11.302238 -11.302238 365.76 0.0
CUBOID 5 16.052324 11.302238 11.302236 -11.302236 365.76 0.0
CUBOID 6 38.052324 -38.052324 11.302236 -11.302236 365.76 0.0
HOLE 7
HOLE 7 ORIGIN X= -27.354562
HOLE 7 ORIGIN X= 27.354562
ARRAY 2 4 PLACE 1 1 1 -14.8648025 -9.418530 0.0
ARRAY 2 5 PLACE 1 1 1 12.4897595 -9.418530 0.0
MEDIA 0 1 6 -5 -4
BOUNDARY 7

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. F17.5.26.

ARA=3 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=4 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

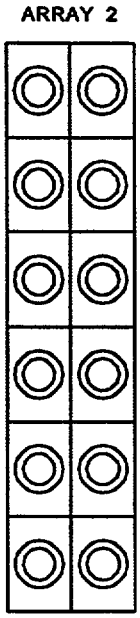


Figure F17.5.24 2 × 6 array of B_4C rods

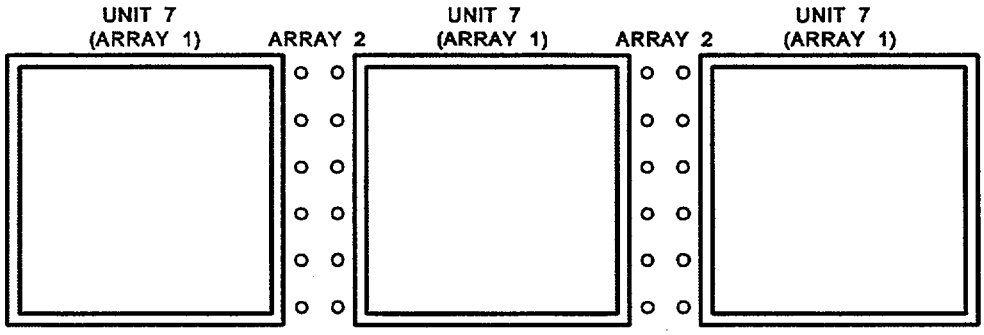


Figure F17.5.25 Central array

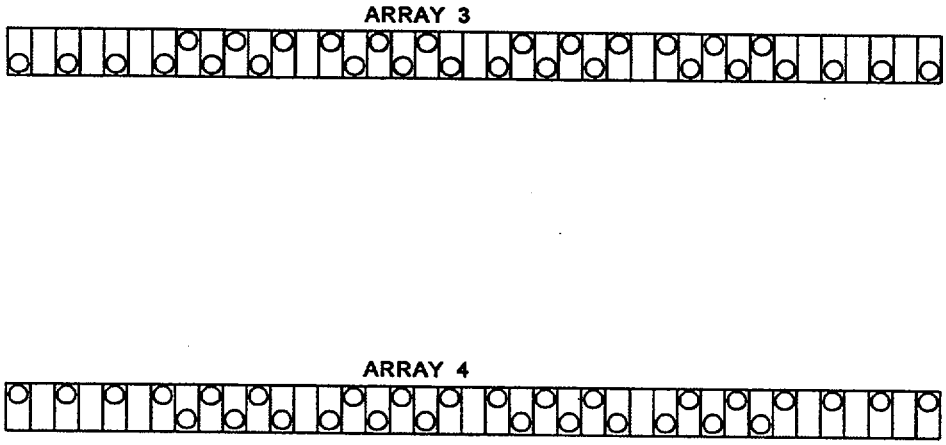


Figure F17.5.26 Long B_4C rod arrays

Unit 8 is positioned between arrays 3 and 4, forming the central section of the global unit, as shown in Fig. F17.5.27. The description below shows how the three central fuel assemblies and B_4C rods are combined. This is the case of the global unit.

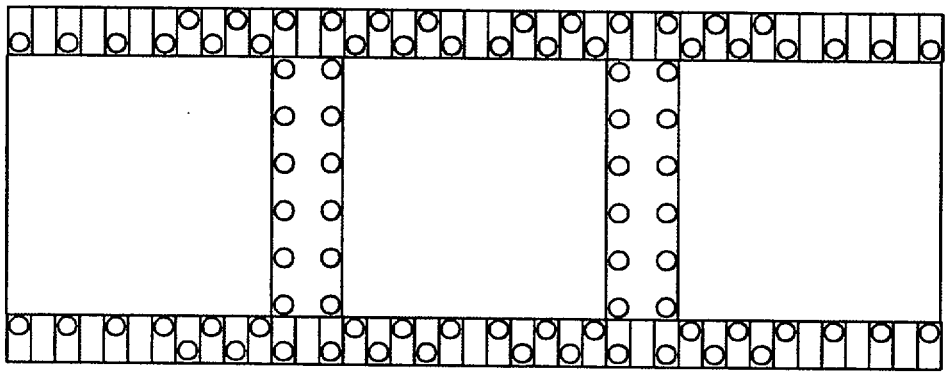


Figure F17.5.27 Central section of global unit with long B_4C arrays

CUBOID 2 38.052324 -38.052324 14.807436 11.302236 365.76 0.0
CUBOID 3 38.052324 -38.052324 11.302236 -14.807436 365.76 0.0
HOLE 8
ARRAY 3 2 PLACE 20 1 1 0.0 13.054836 0.0
ARRAY 4 3 PLACE 20 1 1 0.0 13.054836 0.0

Next, Unit 7 and array 2 are stacked together to form the array of two fuel assemblies separated by B_4C rods, as shown in Fig. F17.5.28. This arrangement is designated as unit 9. The origin of unit 9 is specified at the center of the B_4C array in the x and y directions and the bottom of the array in the z direction.

```
UNIT 9
CUBOID 1 2.375043 -2.375043  11.302236 -11.302236 365.76 0.0
CUBOID 4  24.979519  -23.7919975 11.302238 -11.302238 365.76 0.0
ARRAY 2 1 PLACE 1 1 1 -1.1875215 -9.418530 0.0
HOLE 7 ORIGIN X=-13.67728
HOLE 7 ORIGIN X=13.67728
MEDIA 0 1 4 -1
BOUNDARY  4
```

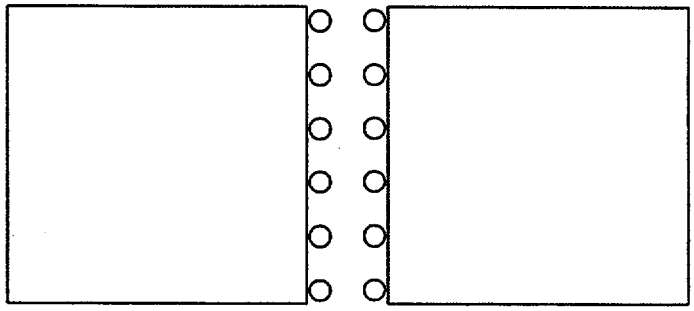


Figure F17.5.28 Two fuel assemblies and B_4C rods

All necessary subassemblies that make up the shipping cask have been built. The shipping cask is now assembled by first specifying the origin of the central section (see Fig. F17.5.27) to be at the center of the array in x and y and the bottom of the array in the z direction. A cylinder of water defining the interior of the shipping cask is described around the array. A HOLE is used to place a Unit 9 (Fig. F17.5.28) below the array, and a second HOLE is used to place another Unit 9 above the array. At this point 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. The lid is then placed on the shipping cask and a boundary region is defined that encloses the entire cask assembly. This information completes the shipping cask description of Fig. F17.5.20. 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 .41148 365.76 0.0
CYLINDER  2 .48133 365.76 0.0
CUBOID  3   .63754 -.63754 .63754 -.63754 365.76 0.0
MEDIA 1 1 1
MEDIA 2 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY  3
UNIT 2
CUBOID  1  .63754 -.63754 .63754 -.63754 365.76 0.0
MEDIA 3 1 1
BOUNDARY  1
UNIT 3
CYLINDER 1   .584 365.76 0.0
CYLINDER 2   .635 365.76 0.0
CUBOID  3   .9912 -.9912 2.2352 -1.27 365.76 0.0
MEDIA 4 1 1
MEDIA 5 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY  3
UNIT 4
CYLINDER 1  .584 365.76 0.0
CYLINDER 2  .635 365.76 0.0
CUBOID  3   .9912 -.9912 1.2702 -2.235 365.76 0.0
MEDIA 4 1 1
MEDIA 5 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY  3
UNIT 5
CUBOID 1   .9912 -.9912 1.7526 -1.7526 365.76 0.0
MEDIA 3 1 1
BOUNDARY  1
UNIT 6
CYLINDER 1  .584 365.76 0.0
CYLINDER 2  .635 365.76 0.0
CUBOID  3   1.1875215 -1.1875215 1.883706 -1.883706 365.76 0.0
MEDIA 4 1 1
MEDIA 5 1 2 -1
MEDIA 3 1 3 -2 -1
BOUNDARY  3
UNIT 7
CUBOID 1  10.83818 -10.83818 10.83818 -10.83818 365.76 0.0
CUBOID 2  11.112495 -11.112495 11.112495 -11.112495 365.76 0.0
CUBOID 3  11.302238 -11.302238 11.302238 -11.302238 365.76 0.0
ARRAY 1 1  PLACE 9 9 1  3*0.0
MEDIA 3 1 2 -1
```

```
MEDIA 8 1 3 -2 -1
BOUNDARY 3
UNIT 8
CUBOID 4 -11.302238 -16.052324 11.302238 -11.302238 365.76 0.0
CUBOID 5 16.052324 11.302238 11.302236 -11.302236 365.76 0.0
CUBOID 6 38.052324 -38.052324 11.302236 -11.302236 365.76 0.0
HOLE 7
HOLE 7 ORIGIN X= -27.354562
HOLE 7 ORIGIN X= 27.354562
ARRAY 2 4 PLACE 1 1 1 -14.8648025 -9.418530 0.0
ARRAY 2 5 PLACE 1 1 1 12.4897595 -9.418530 0.0
MEDIA 0 1 6 -5 -4
BOUNDARY 6
UNIT 10
CUBOID 1 1.1875215 -1.1875215 11.302236 -11.302236 365.76 0.0
CUBOID 4 23.7919975 -23.7919975 11.302238 -11.302238 365.76 0.0
ARRAY 2 1 PLACE 1 1 1 3*0.0
HOLE 7 ORIGIN X=-12.4897595
HOLE 7 ORIGIN X=12.4897595
MEDIA 0 1 4 -1
BOUNDARY 4
GLOBAL UNIT 11
CUBOID 2 38.052324 -38.052324 13.284638 11.302236 365.76 0.0
CUBOID 3 38.052324 -38.052324 11.302236 -13.284638 365.76 0.0
CYLINDER 6 47.625 419.10 -11.43
CYLINDER 7 48.895 419.10 -14.605
CYLINDER 8 59.06 419.10 -24.13
CYLINDER 9 47.625 429.26 421.64
CYLINDER 10 63.01 434.34 -27.94
HOLE 8
ARRAY 3 2 PLACE 20 1 1 0.0 12.293438 0.0
ARRAY 4 3 PLACE 20 1 1 0.0 12.293438 0.0
HOLE 9 ORIGIN Y=24.586876
HOLE 9 ORIGIN Y=-24.586876
MEDIA 3 1 6 -3 -2
MEDIA 7 1 7 -6 -3 -2
MEDIA 6 1 8 -7
MEDIA 7 1 9
MEDIA 6 1 10 -9 -8
BOUNDARY 10
END GEOM
READ ARRAY
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
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
```

F17.5.6.5 Triangular Pitched Arrays

EXAMPLE 21. Triangular pitched arrays can be described in KENO-VI by defining the units that make up the array as hexprism and in the array data block setting TYP=TRIANGULAR. This includes close-packed triangular pitched arrays. Since the arrays are constructed by stacking hexprisms care must be taken to ensure the array boundary is completely enclosed within the stacked unit. Below is an example of a triangular pitched array.

The first and second unit are the hexprisms that make up the array. Unit 1 is the fuel cell that is stacked in a triangular pitched or hexagonal lattice. Unit 2 is a dummy unit that is used to fill in the array so the array boundary is contained withing the stacked units. Since the array is not moderated unit 2 contains void. Figure F17.5.29 shows an X-Y cross-section schematic of units 1 and 2.

```
UNIT 1
COM='SINGLE CELL FUEL CAN IN HEXPRISM'
CYLINDER 10  10.16  18.288  0.0
CYLINDER 20  10.312 18.288 -0.152
HEXPRISM 30  10.503 18.288 -0.152
MEDIA  1  1  10
MEDIA  2  1  20  -10
MEDIA  0  1  30  -20  -10
BOUNDARY  30
```

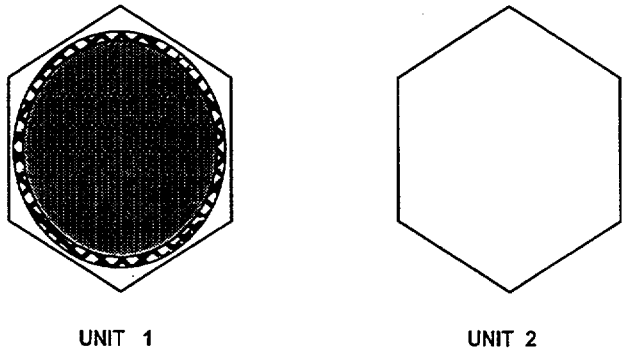


Figure F17.5.29 Fuel cell and empty cell set up as hexprisms

```
UNIT 2
COM='EMPTY CELL'
HEXPRISM 10 10.503 18.288 -0.152
MEDIA 0 1 10
BOUNDARY 10
```

Unit 3 is the global unit that contains the array and array boundary. The array is an unmoderated triangular pitched assembly of 7 cells. Figure F17.5.30 shows an X-Y cross-section schematic of the assembly.

```
GLOBAL UNIT 3
COM='7 CYLINDERS IN A CIRCLE WITH CYLINDRICAL BOUNDARY'
CYLINDER 10 32.000 18.288 -0.152
ARRAY 1 10 PLACE 3 3 1 3*0.0
BOUNDARY 10
READ ARRAY GBL=1 ARA=1 TYP=TRIANGULAR NUX=5 NUY=5 NUZ=1
FILL 7*2 2*1 2*2 3*1 2*2 2*1 7*2 END FILL END ARRAY
```

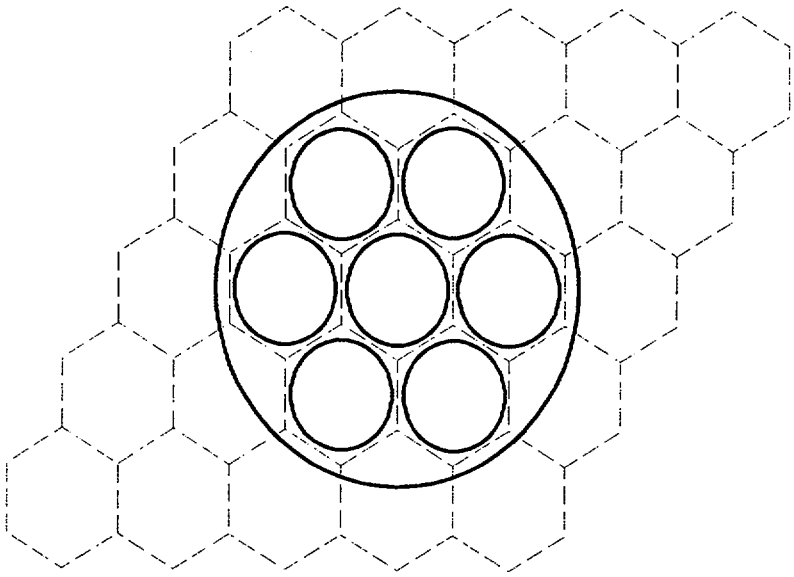


Figure F17.5.30 Seven cylinders stacked in a triangular pitched array with a cylindrical array boundary

The overall problem description is shown below. The working format library used in the mixing table was generated from the 27groupndf4 master library using BONAMI and NITAWL and stored in unit 4. An X-Y cross-section color plot used for verification of this mock-up is shown in Fig. F17.5.31.

Data description of Example 21.

```
=KENOVI
TRIANGULAR PITCHED ARRAY 7 PINS IN A CIRCLE
READ PARAMETERS LNG=20000 LIB=4 END PARAMETERS
READ MIXT SCT=2
MIX=1  892235 1.37751E-03  892238 9.92354E-05  808016 3.32049E-02
    809019 2.95349E-03  801001 6.05028E-02
MIX=2 1413027 6.02374E-02
END MIXT
READ GEOMETRY
UNIT 1
COM='SINGLE CELL FUEL CAN IN HEXPRISM'
CYLINDER 10  10.16  18.288  0.0
CYLINDER 20  10.312 18.288 -0.152
HEXPRISM 30  10.503 18.288 -0.152
MEDIA  1 1 10
MEDIA  2 1 20 -10
MEDIA 0 1 30 -20 -10
BOUNDARY 30
```

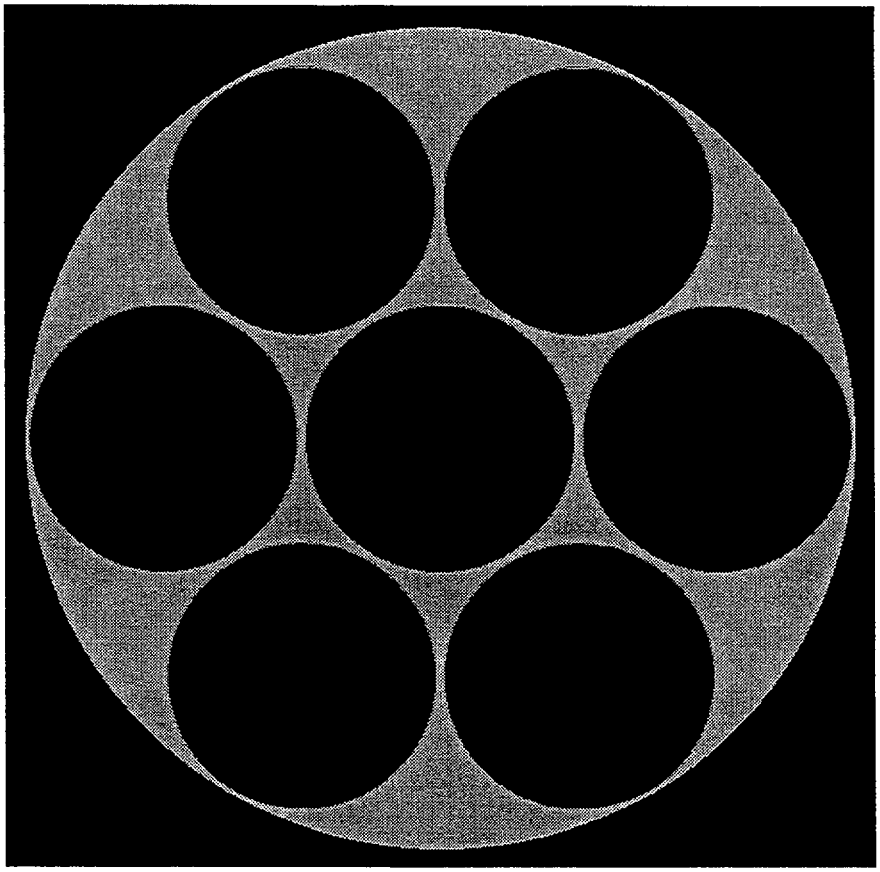


Figure F17.5.31 X-Y slice of 7 cylinders in a triangular pitched array

```
UNIT 2
COM='EMPTY CELL'
HEXPRISM 10  10.503 18.288 -0.152
MEDIA  0 1 10
BOUNDARY  10
GLOBAL UNIT 3
CYLINDER 10  32.000 18.288 -0.152
COM='7 CYLINDERS IN A CIRCLE WITH CYLINDRICAL BOUNDARY'
ARRAY  1 10 PLACE 3 3 1 3*0.0
BOUNDARY  10
END GEOMETRY
READ ARRAY  GBL=1 TYP=TRIANGULAR NUX=5 NUY=5 NUZ=1
FILL 7*2 2*1 2*2 3*1 2*2 2*1 7*2 END FILL  END ARRAY
READ PLOT
TTL='TRIANGULAR PITCHED ARRAY, 7 PINS IN A CIRCLE'
XUL=-33.0  YUL=33.0  ZUL=0.0
XLR=33.0  YLR=-33.0  ZLR=0.0
UAX=1  VDN=-1  NAX=640   END
END PLOT
END DATA
END
```

F17.5.7 ALTERNATIVE SAMPLE PROBLEM MOCK-UPS

The geometry data for KENO-VI can often be described correctly in several ways. Some alternative geometry descriptions are given here for sample problems 12 and 13. (See Appendix F17.D.)

F17.5.7.1 *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, 4, and 5, 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 1  9.525 8.89 -8.89
CYLINDER 2  10.16 9.525 -9.525
CUBOID  3   10.875 -10.875 10.875 -10.875 10.24 -10.24
MEDIA 2 1 1
MEDIA 3 1 2 -1
MEDIA 0 1 3 -2
BOUNDARY  3
```

```
UNIT 2
CYLINDER 1  5.748 9.3975 -1.3675 ORIG  X=4.285 Y=4.285
CUBOID  2   10.875 -10.875 10.875 -10.875 10.24 -10.24
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
UNIT 3
CYLINDER 1  5.748 9.3975 -1.3675 ORIG X=4.285 Y=-4.285
CUBOID  2   10.875 -10.875 10.875 -10.875 10.24 -10.24
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
UNIT 4
CYLINDER 1  5.748 1.3675 -9.3975 ORIG X=4.285 Y=4.285
CUBOID  2   10.875 -10.875 10.875 -10.875 10.24 -10.24
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
UNIT 5
CYLINDER 1  5.748 1.3675 -9.3975 ORIG X=4.285 Y=-4.285
CUBOID  2   10.875 -10.875 10.875 -10.875 10.24 -10.24
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
GLOBAL UNIT 6
CUBOID 1 21.75 -21.75 21.75 -21.75 20.48 -20.48
ARRAY 1 PLACE 1 1 1 -10.875 -10.875 -10.24
BOUNDARY  1
END GEOM
READ ARRAY ARA=1  NUX=2 NUY=2 NUZ=2 FILL 2 1 3 1 4 1 5 1 END ARRAY
```

F17.5.7.2 *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. The origin of the global unit is the point in the array where units 1 through 8 meet. 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 5.748 5.3825 -5.3825
CUBOID  2  6.59 -5.748 6.59 -14.445 6.225 -13.54
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY  2
UNIT 2
```

CYLINDER 1 9.525 8.89 -8.89
CYLINDER 2 10.16 9.525 -9.525
CUBOID 3 10.16 -10.875 10.875 -10.16 10.24 -9.525
MEDIA 2 1 1
MEDIA 3 1 2 -1
MEDIA 0 1 3 -2
BOUNDARY 3
UNIT 3
CYLINDER 1 5.748 5.3825 -5.3825
CUBOID 2 6.59 -5.748 14.444 -6.59 6.225 -13.54
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
UNIT 4
CYLINDER 1 9.525 8.89 -8.89
CYLINDER 2 10.16 9.525 -9.525
CUBOID 3 10.16 -10.875 10.16 -10.875 10.24 -9.525
MEDIA 2 1 1
MEDIA 3 1 2 -1
MEDIA 0 1 3 -2
BOUNDARY 3
UNIT 5
CYLINDER 1 5.748 5.3825 -5.3825
CUBOID 2 6.59 -5.748 6.59 -14.445 13.54 -6.225
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
UNIT 6
CYLINDER 1 9.525 8.89 -8.89
CYLINDER 2 10.16 9.525 -9.525
CUBOID 3 10.16 -10.875 10.875 -10.16 9.525 -10.24
MEDIA 2 1 1
MEDIA 3 1 2 -1
MEDIA 0 1 3 -2
BOUNDARY 3
UNIT 7
CYLINDER 1 5.748 5.3825 -5.3825
CUBOID 2 6.59 -5.748 14.445 -6.59 13.54 -6.225
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
UNIT 8
CYLINDER 1 9.525 8.89 -8.89
CYLINDER 2 10.16 9.525 -9.525
CUBOID 3 10.16 -10.875 10.16 -10.875 9.525 -10.24
MEDIA 2 1 1
MEDIA 3 1 2 -1

```
MEDIA 0 1 3 -2
BOUNDARY 3
GLOBAL UNIT 9
CUBOID 1 20.035 -12.748 20.67 -20.67 19.765 -19.765
ARRAY 1 1 PLACE 1 1 1 -7.00 -6.225 -6.225
BOUNDARY 1
END GEOM
READ ARRAY ARA=1 NUX=2 NUY=2 NUZ=2 FILL 6I1 8 END FILL END ARRAY
```

F17.5.7.3 *Sample Problem 13, Alternative*

This mock-up maintains the same overall unit dimensions that were used in sample problem 13, Appendix F17.D. In sample problem 13, the origin of the unit is located at the center of the ring in the x and y directions and the base of the ring in the z direction. The uranium metal cuboids are directly placed in the problem. In this mock-up, the problem is set up as an array of three units. The origin of Units 1 and 2 are 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 0.2566 -12.4434 6.35 -6.35 3.81 -3.81
CYLINDER 2 13.97 3.81 -3.81
CYLINDER 3 19.05 3.81 -3.81
CUBOID 4 19.05 -19.05 19.05 -19.05 3.81 -3.81
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 1 1 3 -2 -1
MEDIA 0 1 4 -3 -2 -1
BOUNDARY 4
UNIT 2
CUBOID 1 12.4434 -0.2566 6.35 -6.35 4.28 -4.28
CYLINDER 2 13.97 4.28 -4.28
CYLINDER 3 19.05 4.28 -4.28
CUBOID 4 19.05 -19.05 19.05 -19.05 4.28 -4.28
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 1 1 3 -2 -1
MEDIA 0 1 4 -3 -2 -1
BOUNDARY 4
UNIT 3
CUBOID 1 12.4434 -0.2566 6.35 -6.35 1.308 -1.308
CUBOID 2 19.05 -19.05 19.05 -19.05 1.308 -1.308
MEDIA 1 1 1
MEDIA 0 1 2 -1
BOUNDARY 2
GLOBAL UNIT 4
CUBOID 1 19.05 -19.05 19.05 -19.05 6.896 -11.90
```

ARRAY 1 1 PLACE 1 2 1 0.0 0.0 0.0
BOUNDARY 1
END GEOM
READ ARRAY ARA=1 NUX=1 NUY=1 NUZ=3 FILL 1 2 3 END ARRAY

F17.5.8 BIASING OR WEIGHTING DATA

Section F17.2.3 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. F17.5.6. Section F17.4.7 discusses the input directions for entering biasing data.

Every MEDIA-type 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 IDs 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 biasing data. Table F17.4.5 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-VI. 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 IDs 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 ENSURE CONSISTENCY.

2. Biasing data must be entered. These data must include the material ID for the reflector material (from Table F17.4.5 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 IDs 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 F17.4.5 for a list of the increment thickness for each material in the weighting library. It is acceptable for the thickness of the last reflector, and thus the region, to be significantly different than the increment thickness. For example, a reflector card specifying five generated regions with a thickness 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 surrounding a cube of material 1, 20 cm on a side, is desired. The required reflector description and biasing data could be entered as follows:

```
GLOBAL UNIT 1
CUBOID 1 6P10.0
CUBOID 2 6P13.0
CUBOID 3 6P16.0
CUBOID 4 6P19.0
CUBOID 5 6P22.0
CUBOID 6 6P25.0
CUBOID 7 6P25.24
MEDIA 1 1 1
MEDIA 2 2 2 -1
MEDIA 2 3 3 -2
MEDIA 2 4 4 -3
MEDIA 2 5 5 -4
MEDIA 2 6 6 -5
MEDIA 2 7 7 -6
BOUNDARY 7
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:

```
GLOBAL UNIT 1
CUBOID 1 6P10.0
CUBOID 2 6P13.0
CUBOID 3 6P16.0
CUBOID 4 6P19.0
CUBOID 5 6P22.0
CUBOID 6 6P25.24
MEDIA 1 1 1
MEDIA 2 2 2 -1
MEDIA 2 3 3 -2
MEDIA 2 4 4 -3
MEDIA 2 5 5 -4
MEDIA 2 6 6 -5
BOUNDARY 6
READ BIAS ID=500 2 6 END BIAS
```

Here the weighting functions associated with bias IDs 2, 3, 4, and 5 each have a thickness of 3.0 cm, corresponding exactly to the increment thickness for water in Table F17.4.5. 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 F17.4.5 for bias IDs 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 IDs 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 F17.4.5 for bias IDs 2 through 4 and the weighting function for water for bias IDs 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 IDs outside the range 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 IDs 2, 3, and 4, and the data for water (ID=500) will be used for bias IDs 5, 6, and 7.

EXAMPLE 1. USE OF BIASING DATA

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
GLOBAL UNIT 1
SPHERE 1 5.0
SPHERE 2 10.0
SPHERE 3 15.0
SPHERE 4 20.0

SPHERE 5 25.0
MEDIA 2 1 1
MEDIA 1 2 2 -1
MEDIA 1 3 3 -2
MEDIA 1 4 4 -3
MEDIA 1 5 5 -4
BOUNDARY 5
END GEOM
READ BIAS ID=301 2 5 END BIAS

In the above example, the bias ID for the first reflector 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 energy-dependent weighting function of the first 5 cm of concrete is represented by bias ID 2, the second 5-cm interval by bias ID 3, the third 5-cm interval by bias ID 4, and the fourth 5-cm interval by bias ID 5.

F17.5.9 CHARACTER OR COLOR PLOTS

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. Character plots are generated only if **SCR=NO** is entered in the plot data. See Sect. F17.4.11 for a description of 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 the origin of the global unit.

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-VI 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 plot and the direction cosines down the plot define the direction of the vector across the plot and the vector down the plot 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 plot 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 plot and z down the plot 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 plot with x=0.0 at the left and x=10.0 at the right, only the x component of the direction cosines across the plot 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 plot. VAX and WAX could be omitted. Z is to be plotted down the plot 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 plot 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 plot. 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. 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. Color plots plot LPI pixel across per 10 pixel down. The portion of the plot that is printed in the vertical direction will be reduced by a factor of 10 relative to the portion plotted in the horizontal direction ((5.0*LPI)/(0.5*10)).

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 plot; 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 plot. It may be convenient for the user to specify the number of characters that can be printed across the plot (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 for a character plot, the number of lines per inch that are printed on a page. It should be entered to be consistent with the printer that will be used. The default value is 8 lines per inch. For a color plot, LPI is the number of pixels down per 10 pixel across. The default value is 10. This parameter must 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. F17.5.6.1, F17.5.6.2, F17.5.6.3, and F17.5.6.4. 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:

```
=KENOVI
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
GLOBAL UNIT 1
CYLINDER 1 2.6 2P15.0
CYLINDER 2 3.6 2P15.0
CYLINDER 3 3.8 2P15.0
CUBOID 4 4P3.8 2P15.0
CUBOID 5 4P4.3 2P16.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 4 1 3 -2 -1
MEDIA 0 1 4 -3 -2 -1
MEDIA 2 1 5 -4
BOUNDARY  5
END GEOM
```

CHARACTER PLOT DATA DESCRIPTION

```
READ PLOT   SCR=NO
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
```

COLOR PLOT DATA DESCRIPTION

```
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=640  END
PIC=UNIT   END
END PLOT
END DATA
END
```

The plot data blocks included above are 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. In the above examples, the geometry dimensions extend from x = -4.3 to x = 4.4, from y = -4.3, to y = 4.3, and from z = -8.0 to z = 8.0. An x-y slice is 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 X =4.6 YLR=-4.6 ZLR=0.0. It is desired to print x across the plot and y down the plot. Therefore, the x direction cosine is specified across the plot, in the direction from x = -4.6 to x = 4.6 as UAX=1.0. The y direction cosine is specified down the plot, from y = 4.6 to y = -4.6 as VDN=-1.0.

For a printer plot, question marks will be printed for points outside the range of the problem geometry description. By setting the plot dimension slightly larger than the geometry dimension, a border of questions marks will be printed around the specified plot. This verifies that the outer boundaries of the geometry are contained within the plot dimensions. It was desirable for the printer 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 plots and associated data are shown in Figs. F17.5.32 and F17.5.33. 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. F17.5.34 and F17.5.35.

For a screen plot, a black border will be printed for points outside the range of the problem geometry description. By setting the plot dimension slightly larger than the geometry dimension, a black border will be printed around the specified plot. This verifies that the outer boundaries of the geometry are contained within the plot dimensions. NAX is the number of pixels for a screen plot. The size of the computer screen will determine the maximum number of pixels to be viewed. A recommended range for NAX is between 400 and 600 pixels. NCH is not used for screen plots.

The resultant screen plots and associated data are shown in Figs. F17.5.36 and F17.5.37. A second screen plot covering the same area shows a unit map rather than a mixture map. This unit map and associated data are shown in Figs. F17.5.38 and F17.5.39.

```
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     -4.6000E+00           4.6000E+00
y      4.6000E+00          -4.6000E+00
z      0.0000E+00           0.0000E+00
          u axis       v axis
          (down)       (across)
x          .00000       1.00000
y        -1.00000        .00000
z          .00000        .00000
nu=   86   nv=  130     delu= 1.0615E-01     delv= 7.0769E-02
```

Figure F17.5.32 Associated data for single unit mixture map

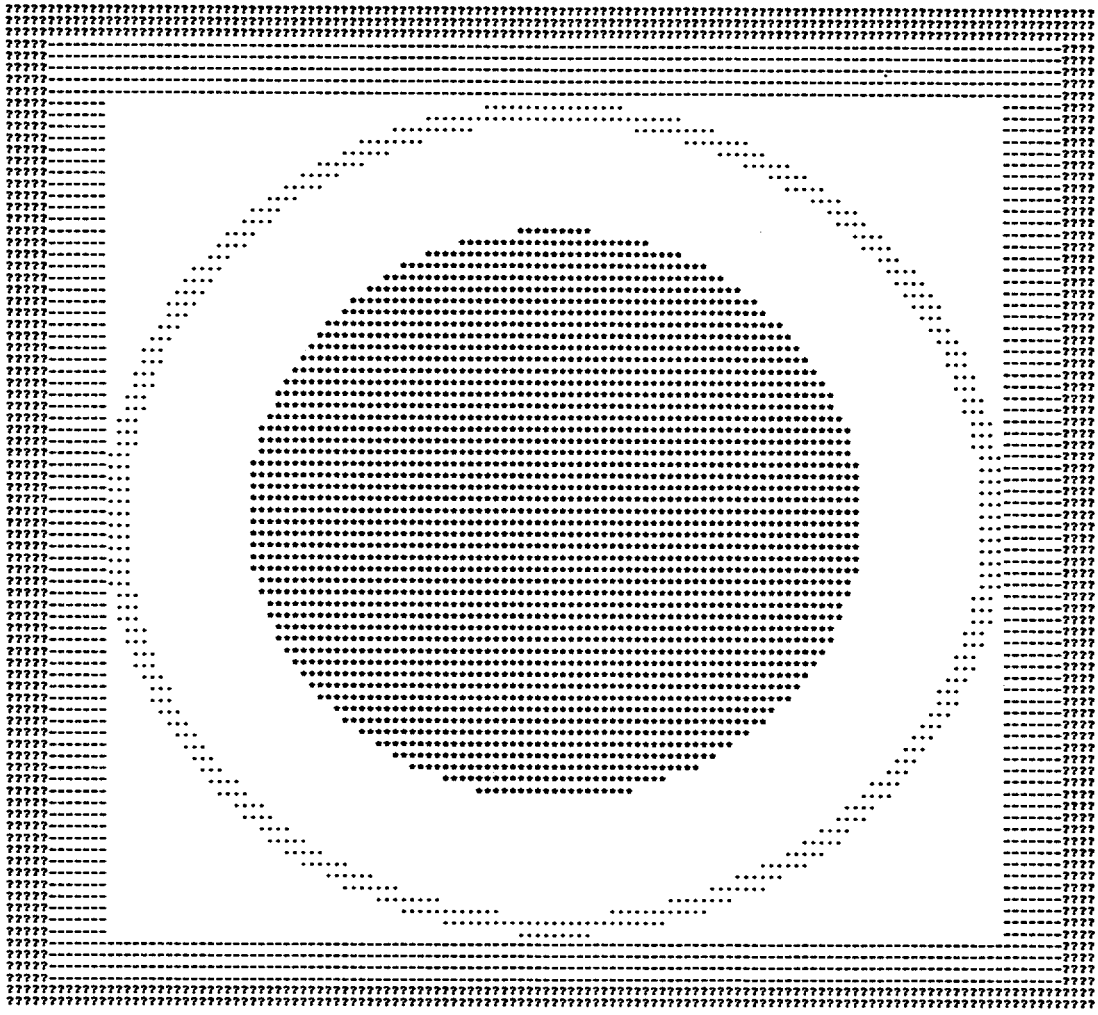


Figure F17.5.33 Mixture map of single unit with centered origin

```
x-y slice at z midpoint.  single unit concentric cyls.

                                                    unit map
   unit  1
 symbol  1
       upper  left          lower right
       coordinates          coordinates
x      -4.6000E+00           4.6000E+00
y       4.6000E+00          -4.6000E+00
z       0.0000E+00           0.0000E+00
           u axis       v axis
           (down)       (across)
x           .00000       1.00000
y         -1.00000         .00000
z           .00000         .00000
nu=   86   nv=  130     delu= 1.0698E-01     delv= 7.0769E-02
```

Figure F17.5.34 Associated data for single unit map

Figure F17.5.35 shows a block of 1's surrounded by a border of ?'s. This arrangement indicates that the entire slice specified in the plot data was part of Unit 1. For this problem the entire volume is Unit 1.

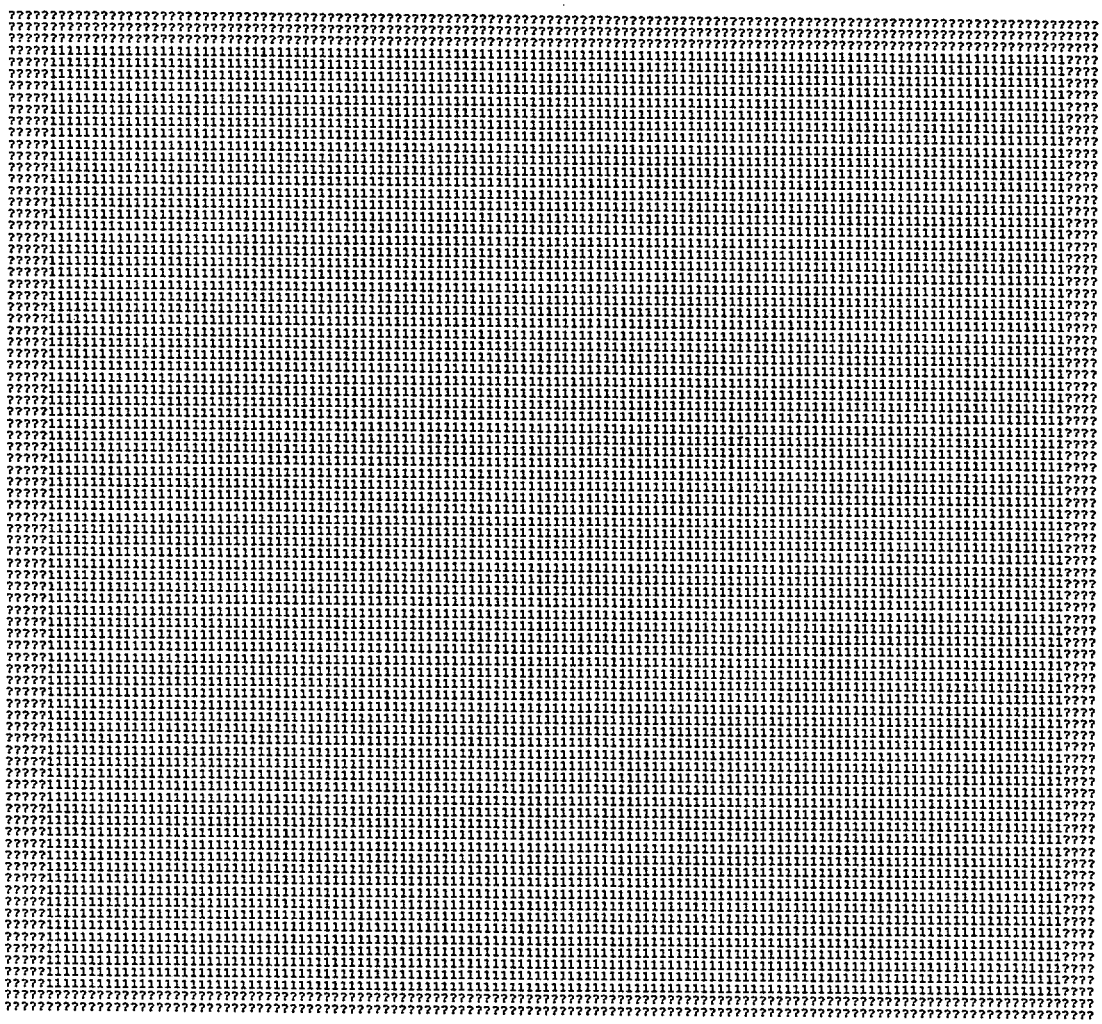


Figure F17.5.35 Unit map character plot for single unit with centered origin

```
                                        mixture map

mixture  0 1 2 4
 symbol    1 2 3
      upper  left          lower right
      coordinates          coordinates
x     -4.6000E+00           4.6000E+00
y      4.6000E+00          -4.6000E+00
z      0.0000E+00           0.0000E+00
          u axis      v axis
          (down)      (across)
x         0.00000      1.00000
y        -1.00000      0.00000
z         0.00000      0.00000
nu=  640   nv=  640     delu= 1.4375E-02     delv= 1.4375E-02     lpi=  10.000
```

Figure F17.5.36 Associated data for single unit mixture map

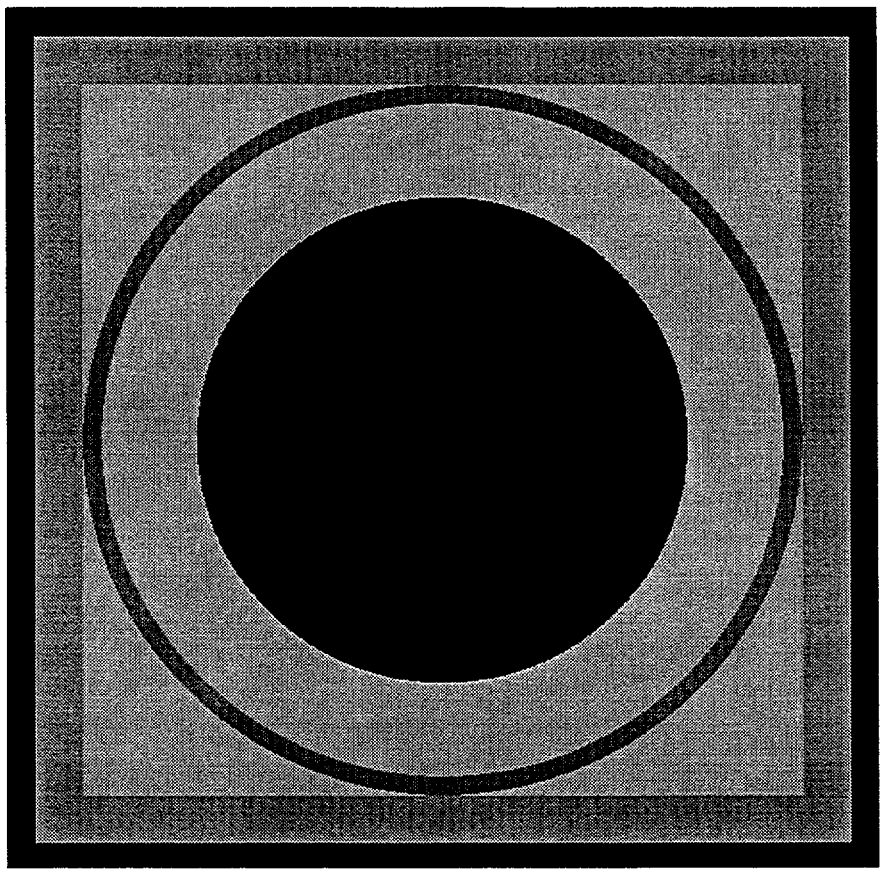


Figure F17.5.37 Mixture color plot for single unit with centered origin

```
x-y slice at z midpoint. single unit concentric cyls

                                        unit map

   unit  1
 symbol  1
      upper  left          lower right
      coordinates          coordinates
x     -4.6000E+00           4.6000E+00
y      4.6000E+00          -4.6000E+00
z      0.0000E+00           0.0000E+00
          u axis       v axis
          (down)       (across)
x         0.00000       1.00000
y        -1.00000       0.00000
z         0.00000       0.00000
nu=  640   nv=  640     delu= 1.4375E-02     delv= 1.4375E-02     lpi=  10.000
```

Figure F17.5.38 Associated data for single unit map

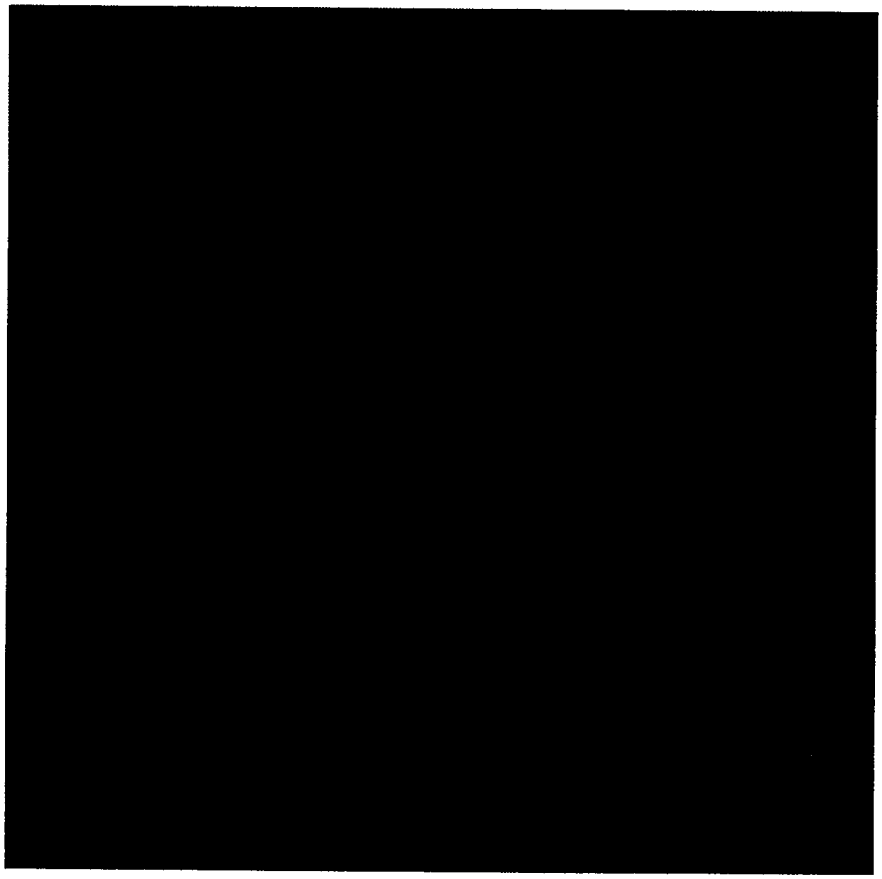


Figure F17.5.39 Unit map color plot for single unit with centered origin

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. Thus the cylinders must 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:

```
=KENOVI
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
GLOBAL UNIT 1
CYLINDER 1 2.6 31.0 1.0 ORIGIN X=4.3 Y=4.3
CYLINDER 2 3.6 31.0 1.0 ORIGIN X=4.3 Y=4.3
CYLINDER 3 3.8 31.0 1.0 ORIGIN X=4.3 Y=4.3
CUBOID  4 8.1 0.5 8.1 0.5 31.0 1.0
CUBOID 5  8.6 0.0 8.6 0.0 32.0 0.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 4 1 3  -2 -1
MEDIA 0 1 4 -3 -2 -1
MEDIA 2 1 5 -4
BOUNDARY  5
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=640 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 black border around the plot. The associated plot data specification for the mixture map is shown in Fig. F17.5.40, the mixture map is shown in Fig. F17.5.41, and the associated plot data for the unit map is shown in Fig. F17.5.42. The unit map is identical to Fig. F17.5.35 and is not included. Note that Fig. F17.5.41 is identical to Fig. F17.5.30.

```
 x-y slice at z midpoint.  single unit concentric cyls.

                                    mixture map

mixture  0 1 2 4
 symbol    1 2 3
      upper  left          lower right
      coordinates          coordinates
x     -3.0000E-01           8.9000E+00
y      8.9000E+00          -3.0000E-01
z      1.6000E+01           1.6000E+01
          u axis       v axis
          (down)       (across)
x         0.00000       1.00000
y        -1.00000       0.00000
z         0.00000       0.00000
nu=  640   nv=  640     delu= 1.4375E-02     delv= 1.4375E-02     lpi=  10.000
```

Figure F17.5.40 Associated data for mixture map of single unit with offset origin

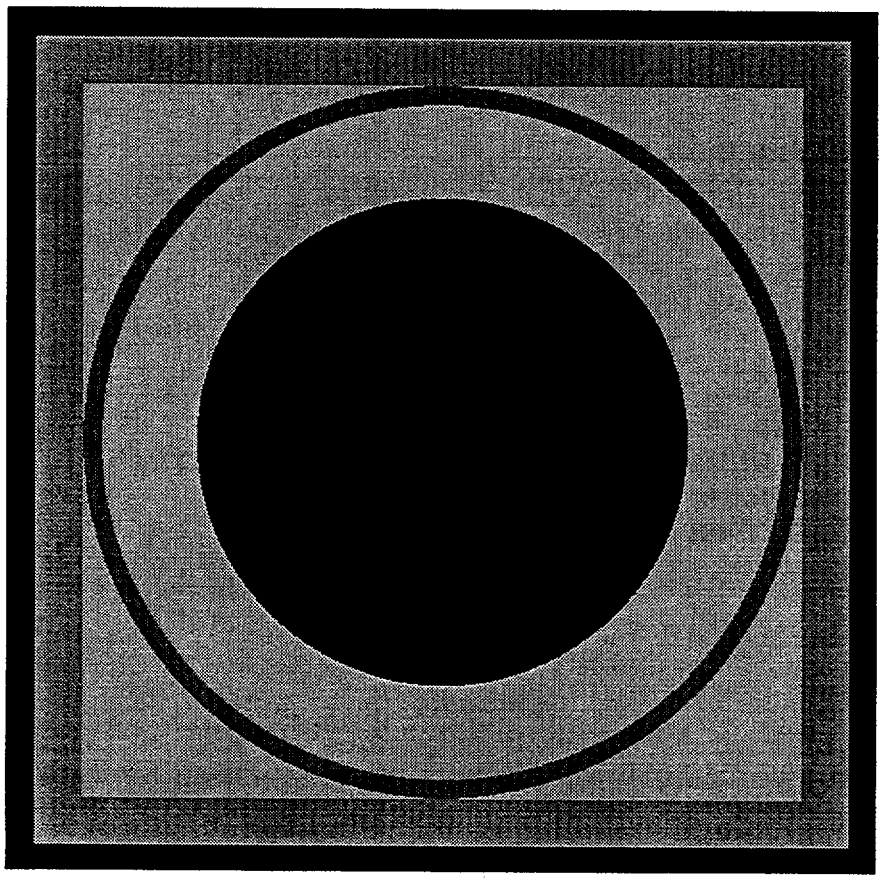


Figure F17.5.41 Mixture map of single unit with offset origin

```
x-y slice at z midpoint.  single unit concentric cyls.

                                        unit map

   unit  1
 symbol  1
      upper  left          lower right
      coordinates          coordinates
x     -3.0000E-01           8.9000E+00
y      8.9000E+00          -3.0000E-01
z      1.6000E+01           1.6000E+01
          u axis       v axis
          (down)      (across)
x         0.00000      1.00000
y        -1.00000      0.00000
z         0.00000      0.00000
nu=  640   nv=  640     delu= 1.4375E-02     delv= 1.4375E-02     lpi=  10.000
```

Figure F17.5.42 Associated data for unit map of a single unit with offset origin

EXAMPLE 3. A 2 × 2 × 2 UNREFLECTED ARRAY OF CONCENTRIC CYLINDERS IN CUBOIDS

The physical representation of this example is a 2 × 2 × 2 array of the configuration described in Example 1 of this section. The input data description for this array is given below:

```
=KENOVI
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
UNIT 1
CYLINDER 1 2.6 2P15.0
CYLINDER 2 3.6 2P15.0
CYLINDER 3 3.8 2P15.0
CUBOID 4 4P3.8 2P15.0
CUBOID 5 4P4.3 2P16.0
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 4 1 3 -2 -1
MEDIA 0 1 4 -3 -2 -1
MEDIA  2 1 -5 -4
BOUNDARY  5
GLOBAL UNIT 2
CUBOID 1 17.2 0.0 17.2 0.0 64.0 0.0
ARRAY 1 1 PLACE 1 1 1 4.3 4.3 16.0
BOUNDARY  1
END GEOM
READ ARRAY  ARA=1 NUX=2 NUY=2 NUZ=2 FILL F1 END FILL  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=640 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=320 END
END PLOT
END DATA
END

As stated at the beginning of Sect. F17.5.9, the origin of the plot is located at the most negative point 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 color plot is to generate an x-y slice through the array at the half-height of the first layer, as shown in Fig. F17.5.43. This plot occurs at z=16.0 cm. It is desirable to define the outer boundaries of the array 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 black border 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. These coordinate choices would have eliminated the black border. The existing picture was made using XUL=-0.3 YUL=17.5 ZUL=16.0 XLR=17.5 YLR=-0.3 ZLR=16.0.

The second color plot is to generate an x-z slice through the center of the front row of the array. In order to obtain a black border, 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 resultant mixture map is shown in Fig. F17.5.44.

EXAMPLE 4. A 2 × 2 × 2 REFLECTED ARRAY WITH THE ORIGIN AT THE MOST NEGATIVE POINT OF THE ARRAY

The array is the array described in Example 3 of this section and has a 15.24-cm (6-in.) concrete reflector on all faces. The input data description for this array is given below.

=KENOVI
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
UNIT 1
CYLINDER 1 2.6 2P15.0
CYLINDER 2 3.6 2P15.0
CYLINDER 3 3.8 2P15.0
CUBOID 4 4P3.8 2P15.0
CUBOID 5 4P4.3 2P16.0

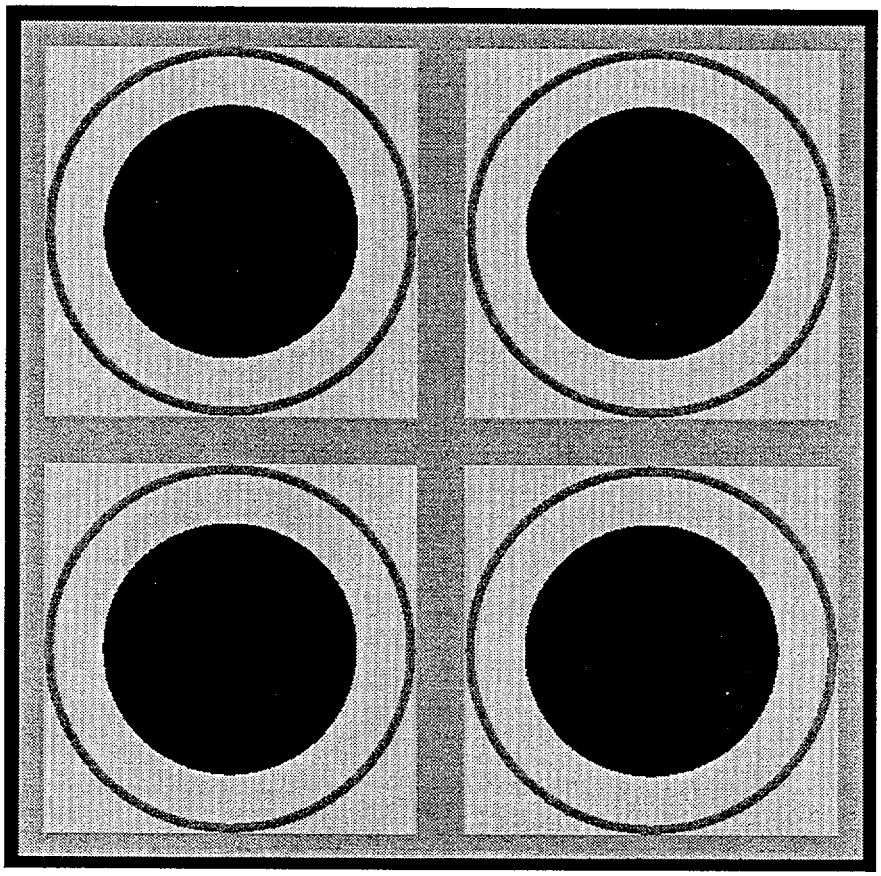


Figure F17.5.43 x-y plot of 2 × 2 × 2 bare array

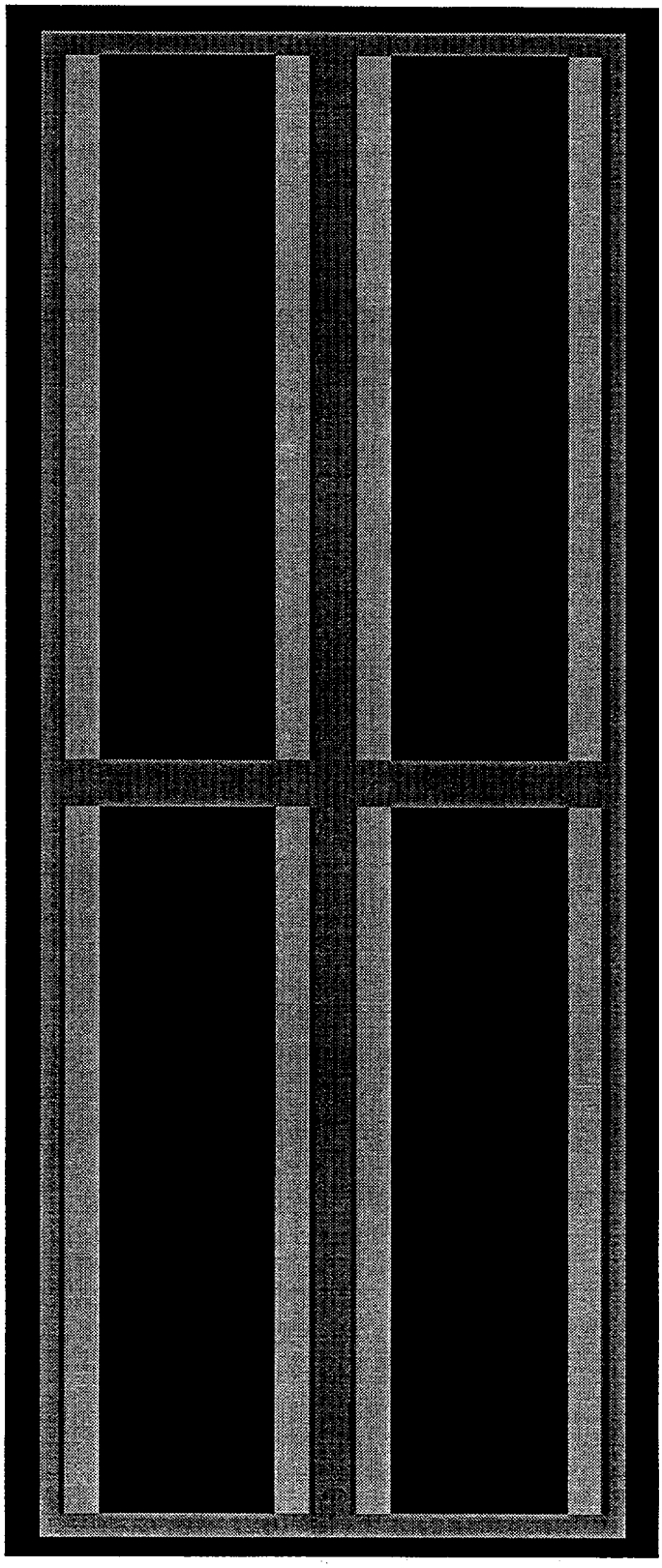


Figure F17.5.44 x-z plot of 2 × 2 × 2 bare array

```
MEDIA 1 1 1
MEDIA 0 1 2 -1
MEDIA 4 1 3 -2 -1
MEDIA 0 1 4 -3 -2 -1
MEDIA 2 1 5 -4
BOUNDARY 5
GLOBAL UNIT 2
CUBOID 1 17.2    0.0   17.2    0.0   64.0    0.0
CUBOID 2 22.20  -5.00 22.20  -5.00 69.00  -5.00
CUBOID 3 27.20 -10.00 27.20 -10.00 74.00 -10.00
CUBOID 4 32.20 -15.00 32.20 -15.00 79.00 -15.00
CUBOID 5 32.44 -15.24 32.44 -15.24 79.24 -15.24
ARRAY 1 1 PLACE 1 1 1 4.3 4.3 16.0
MEDIA 3 2 2 -1
MEDIA 3 3 3 -2
MEDIA 3 4 4 -3
MEDIA 3 5 5 -4
BOUNDARY 5
END GEOM
READ BIAS ID=301 2 5 END BIAS
READ ARRAY ARA=1 NUX=2 NUY=2 NUZ=2 FILL F1 END FILL 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=640 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=640 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=640 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=640 END
END PLOT
END DATA
END
```

The ARRAY record, associated with the first geometry record in global unit 2, specifies the coordinates of the most negative point of the array to be (0.0,0.0,0.0) and the most positive point to be (17.2,17.2,64.0). 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 color plot for this example shows an x-y slice through the array and reflector at the half-height of the bottom layer. A black border is used to verify that the entire reflector has been shown. This border 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. F17.5.45, and the plot is shown in Fig. F17.5.46.

```
 x-y slice at half height of bottom layer.includes refl.

                                        mixture map

mixture  0 1 2 3 4
 symbol    1 2 3 4
      upper  left          lower right
      coordinates          coordinates
x     -1.6240E+01           3.3440E+01
y      3.3440E+01          -1.6240E+01
z      1.6000E+01           1.6000E+01
          u axis       v axis
          (down)       (across)
x         0.00000       1.00000
y        -1.00000       0.00000
z         0.00000       0.00000
nu=  640   nv=  640     delu= 7.7625E-02     delv= 7.7625E-02     lpi=  10.000
```

Figure F17.5.45 Plot data for x-y slice of example 4

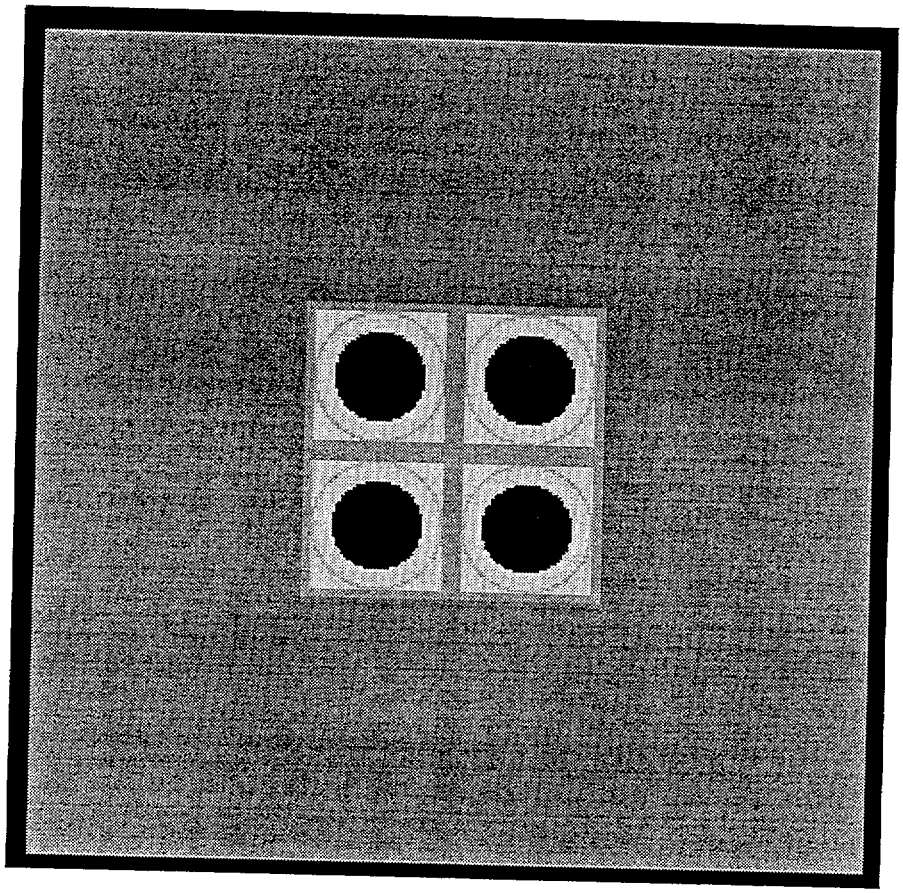


Figure F17.5.46 x-y plot of 2 × 2 × 2 reflected array

The next color 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. F1.5.42, and the plot is shown in Fig. F17.5.48.

The third color 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 black border 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. F17.5.49 and F17.5.50.

```
x-y slice at half height of bottom layer, include 3 cm of refl.

                                        mixture map

 mixture  0 1 2 3 4
  symbol    1 2 3 4
       upper  left          lower right
       coordinates          coordinates
 x     -3.0000E+00           2.0200E+01
 y      2.0200E+01          -3.0000E+00
 z      1.6000E+01           1.6000E+01
          u axis       v axis
          (down)       (across)
 x        0.00000       1.00000
 y       -1.00000       0.00000
 z        0.00000       0.00000
 nu=  640   nv=  640     delu= 3.6250E-02     delv= 3.6250E-02     lpi=  10.000
```

Figure F17.5.47 Plot data for enlarged x-y slice of example 4

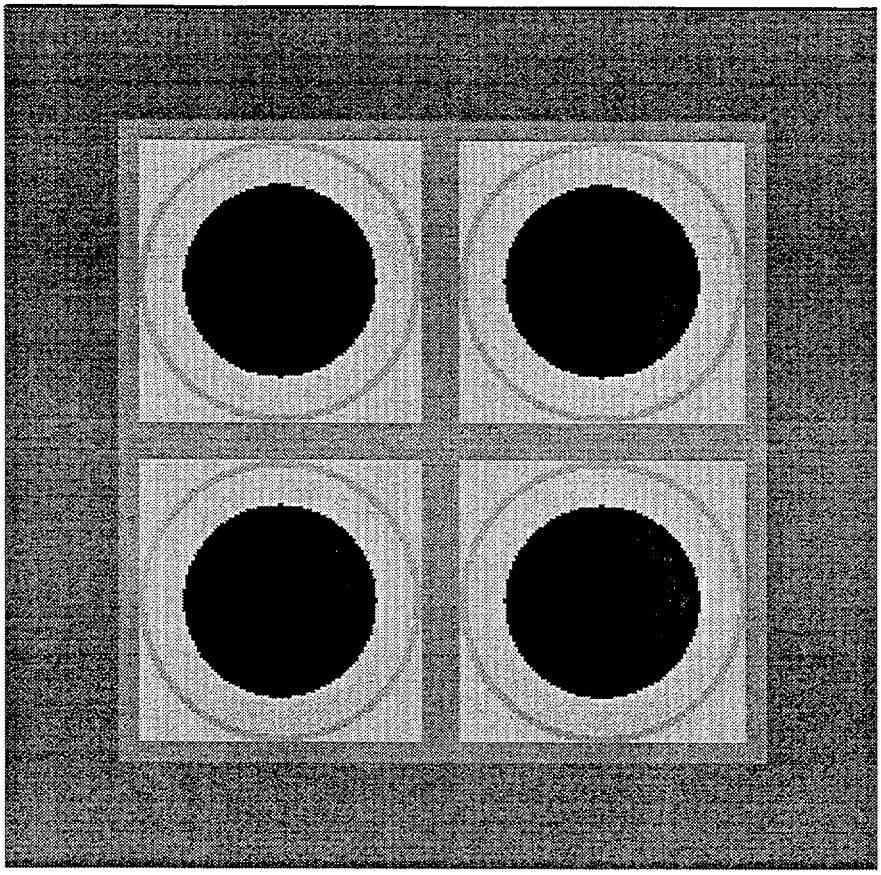


Figure F17.5.48 Enlarged x-y plot of 2 × 2 × 2 reflected array

```
x-z slice through front row, y=12.9. include reflector

                                    mixture map

 mixture  0 1 2 3 4
  symbol    1 2 3 4
       upper  left          lower right
       coordinates          coordinates
 x     -1.6240E+01           3.3440E+01
 y      1.2900E+01           1.2900E+01
 z      8.0240E+01          -1.6240E+01
          u axis       v axis
          (down)       (across)
 x        0.00000       1.00000
 y        0.00000       0.00000
 z       -1.00000       0.00000
 nu= 1242   nv=  640     delu= 7.7625E-02     delv= 7.7625E-02     lpi=  10.000
```

Figure F17.5.49 Plot data for x-z slice of example 4

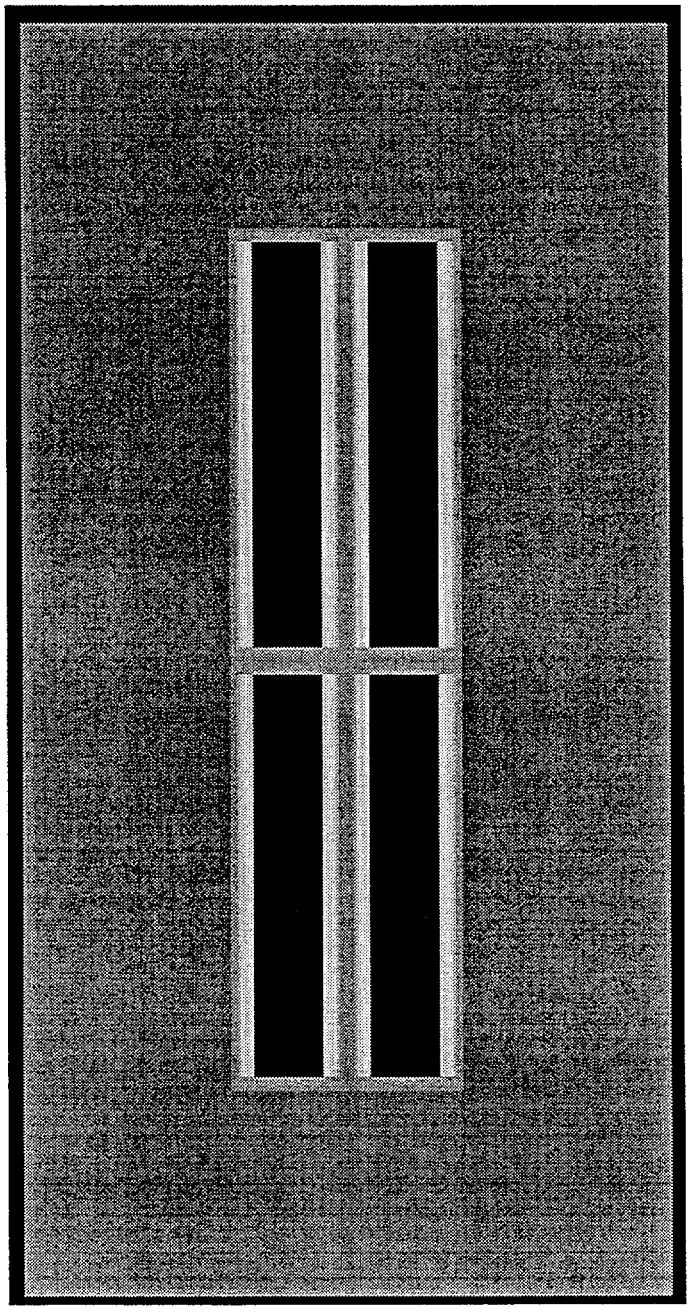


Figure F17.5.50 x-z plot for 2 × 2 × 2 reflected array

The last color 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. F17.5.51 and F17.5.52.

```
x-z slice through front row, y=12.9. include 3 cm of reflector

                                        mixture map

 mixture  0 1 2 3 4
  symbol    1 2 3 4
       upper  left          lower right
       coordinates          coordinates
x      -3.0000E+00           2.0200E+01
y       1.2900E+01           1.2900E+01
z       6.7000E+01          -3.0000E+00
           u axis       v axis
           (down)       (across)
x          0.00000       1.00000
y          0.00000       0.00000
z         -1.00000       0.00000
nu= 1931   nv=  640     delu= 3.6250E-02     delv= 3.6250E-02     lpi=  10.000
```

Figure F17.5.51 Plot data for enlarged x-z slice of example 4

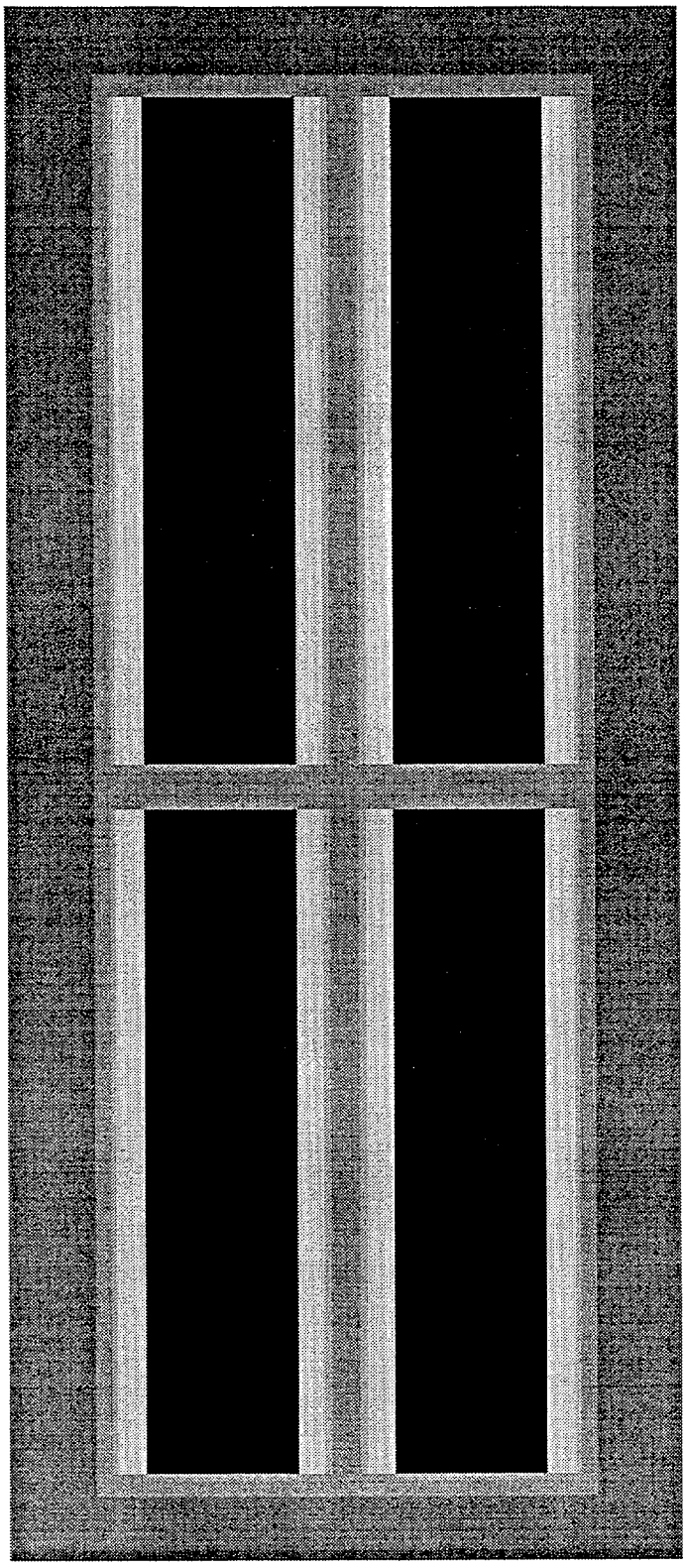


Figure F17.5.52 Enlarged x-z plot of 2 × 2 × 2 reflected array

EXAMPLE 5. A 2 × 2 × 2 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 most negative point of the array as x = -8.6, y = -8.6 and z = -32.0. This positioning is done using the CUBOID card in association with the PLACE section of the ARRAY card. 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.

```
=KENOVI
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
UNIT 1
CYLINDER 1 2.6 2P15.0
CYLINDER 2 3.6 2P15.0
CYLINDER 3 3.8 2P15.0
CUBOID 4 4P3.8 2P15.0
CUBOID 5 4P4.3 2P16.0
MEDIA 1 1
MEDIA 0 1 2 -1
MEDIA 4 1 3 -2 -1
MEDIA 0 1 4 -3 -2 -1
MEDIA 2 1 5 -4
BOUNDARY  5
GLOBAL UNIT 2
CUBOID 1 8.6 -8.6 8.6 -8.6 32.0 -32.0
CUBOID 2 13.6 -13.6 13.6 -13.6 37.0 -37.0
CUBOID 3 18.6 -18.6 18.6 -18.6 42.0 -42.0
CUBOID 4 23.6 -23.6 23.6 -23.6 47.0 -47.0
CUBOID 5 23.84 -23.84 23.84 -23.84 47.24 -47.24
ARRAY 1 1 PLACE 1 1 1 -4.3 -4.3 -16.0
MEDIA 3 2 2 -1
MEDIA 3 3 3 -2
MEDIA 3 4 4 -3
MEDIA 3 5 5 -4
BOUNDARY  5
END GEOM
READ BIAS ID=301 2 5 END BIAS
READ ARRAY ARA=1 NUX=2 NUY=2 NUZ=2I FILL F1 END FILL  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=640 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=640 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=640 END
END PLOT
END DATA
END

The first color 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. F17.5.53 and F17.5.54.

```
x-y slice at half height of bottom layer. includes refl.

                                        mixture map

 mixture  0 1 2 3 4
  symbol    1 2 3 4
       upper  left          lower right
       coordinates          coordinates
 x     -2.4840E+01           2.4840E+01
 y      2.4840E+01          -2.4840E+01
 z     -8.0000E+00          -8.0000E+00
           u axis       v axis
           (down)       (across)
 x         0.00000       1.00000
 y        -1.00000       0.00000
 z         0.00000       0.00000
 nu=  640   nv=  640     delu= 7.7625E-02     delv= 7.7625E-02     lpi=  10.000
```

Figure F17.5.53 Plot data for x-y slice of example 5

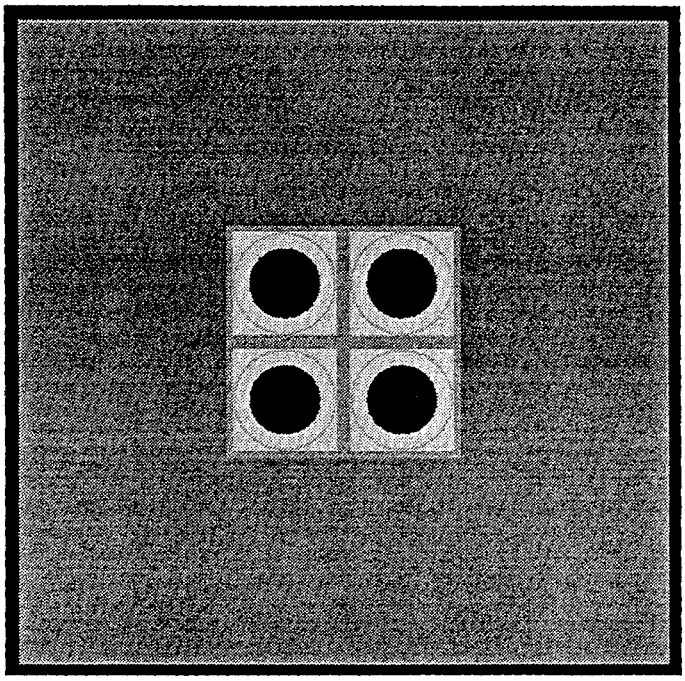


Figure F17.5.54 The x-y plot of 2 × 2 × 2 reflected array with centered origin

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. F17.5.55 through F17.5.60.

```
x-y slice at half height of bottom layer, include 3 cm of refl.

                                    mixture map

 mixture   0 1 2 3 4
  symbol     1 2 3 4
       upper  left          lower right
       coordinates          coordinates
 x      -1.1600E+01           1.1600E+01
 y       1.1600E+01          -1.1600E+01
 z      -8.0000E+00          -8.0000E+00
           u axis       v axis
           (down)       (across)
 x         0.00000       1.00000
 y        -1.00000       0.00000
 z         0.00000       0.00000
 nu=   640    nv=   640     delu= 3.6250E-02     delv= 3.6250E-02     lpi=  10.000
```

Figure F17.5.55 Plot data for an enlarged x-y slice of example 5

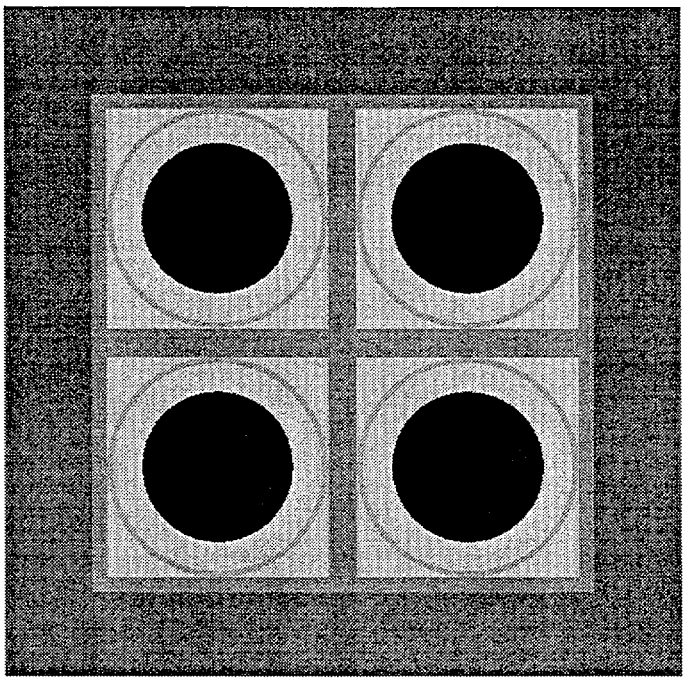


Figure F17.5.56 Enlarged z-y plot of 2 × 2 × 2 reflected array with centered origin

```
x-z slice through front row. y=4.3  include reflector

                                        mixture map

 mixture  0 1 2 3 4
  symbol    1 2 3 4
       upper  left          lower right
       coordinates          coordinates
 x     -2.4840E+01           2.4840E+01
 y      4.3000E+00           4.3000E+00
 z      4.8240E+01          -4.8240E+01
          u axis       v axis
          (down)       (across)
 x        0.00000       1.00000
 y        0.00000       0.00000
 z       -1.00000       0.00000
 nu= 1242   nv=  640     delu= 7.7625E-02     delv= 7.7625E-02     lpi=  10.000
```

Figure F17.5.57 Plot data for x-z slice of example 5

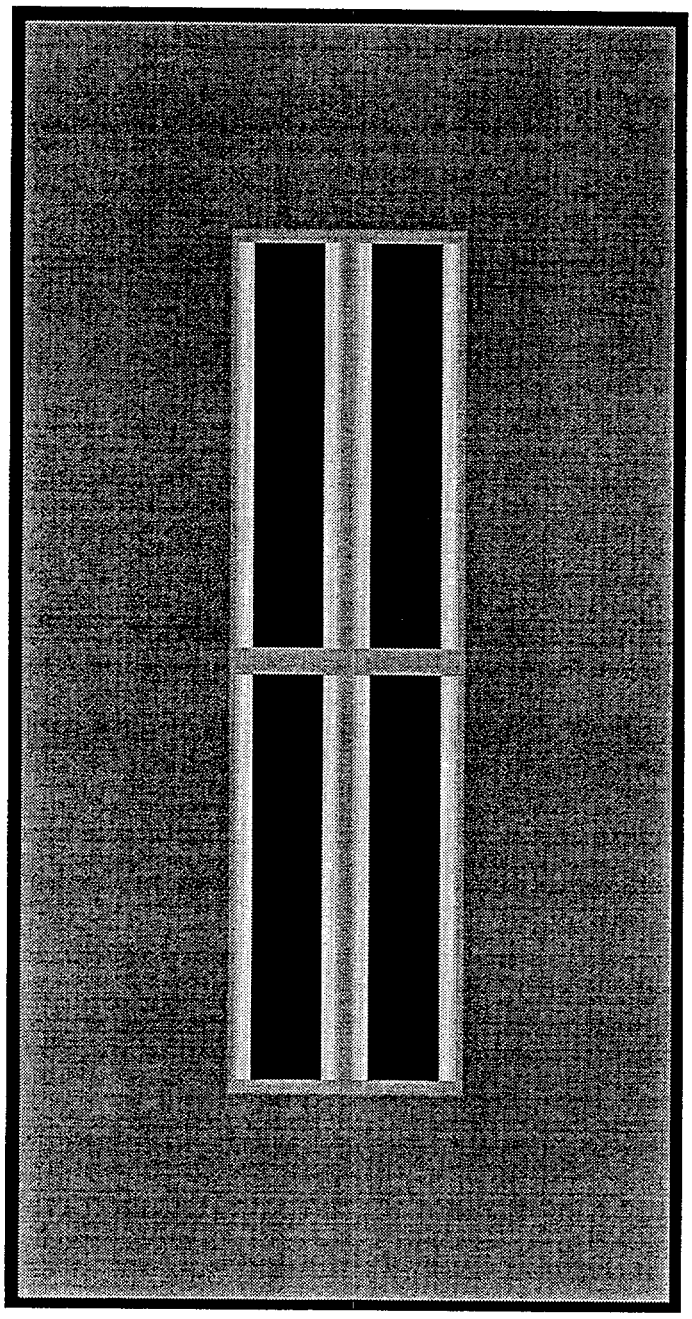


Figure F17.5.58 x-z plot of reflected 2 × 2 × 2 array with centered origin

```
x-z slice through front row, y=4.3  include 3 cm of reflector

                                    mixture map

 mixture  0 1 2 3 4
  symbol    1 2 3 4
       upper  left          lower right
       coordinates          coordinates
 x     -1.1600E+01           1.1600E+01
 y      4.3000E+00           4.3000E+00
 z      3.5000E+01          -3.5000E+01
          u axis      v axis
          (down)      (across)
 x        0.00000      1.00000
 y        0.00000      0.00000
 z       -1.00000      0.00000
 nu= 1931   nv=  640     delu= 3.6250E-02     delv= 3.6250E-02     lpi=  10.000
```

Figure F17.5.59 Plot data for enlarged x-z slice of example 5

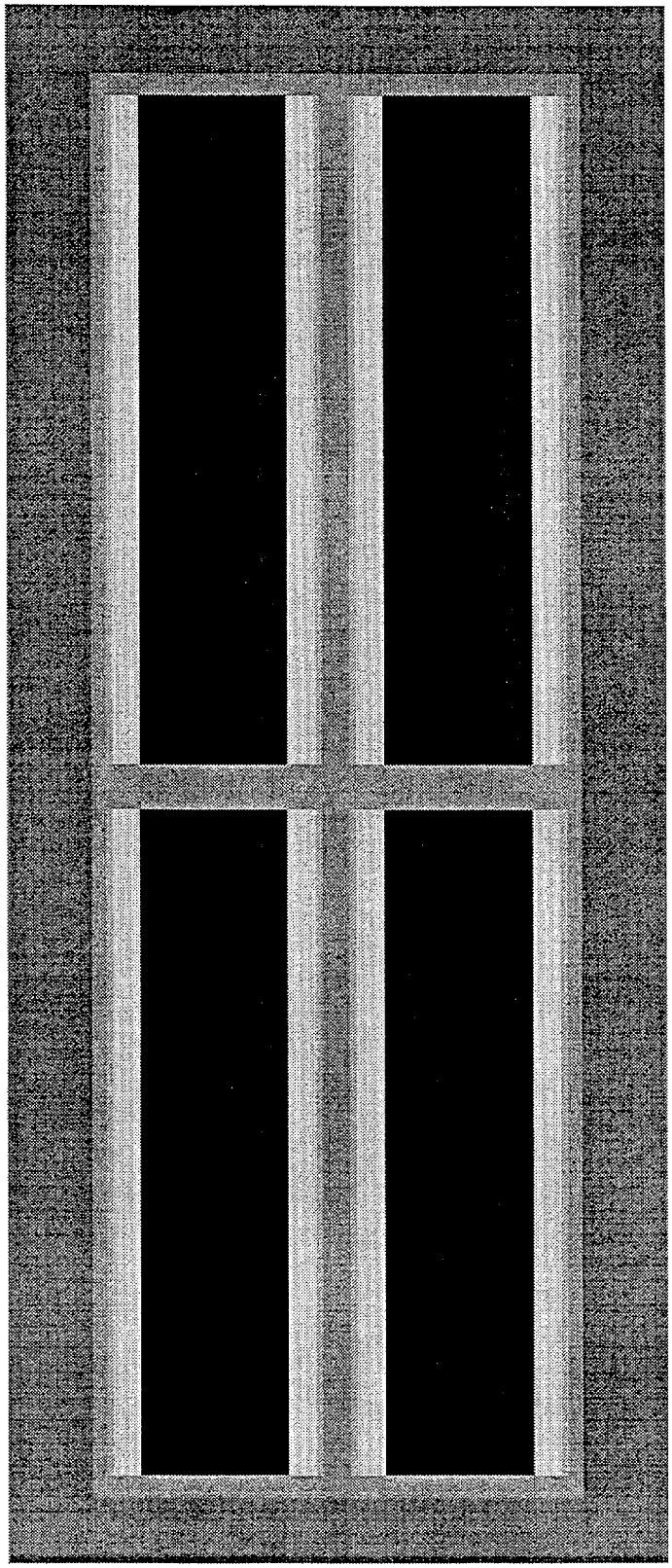


Figure F17.5.60 Enlarged x-z plot of reflected $2 \times 2 \times 2$ array with centered origin

EXAMPLE 6. NESTED HOLES

Refer to the nested hole description of Sect. F17.5.6.2. This example is one that involves a reasonably complicated placement of units; therefore, it might be useful to the user to generate both a mixture map and a unit map for the problem. The resultant mixture map is shown in Fig. F17.5.61. The plot data and actual plot for the unit map are shown in Figs. F17.5.62 and F17.5.63. The data description for Example 6 follows Fig. F17.5.63.

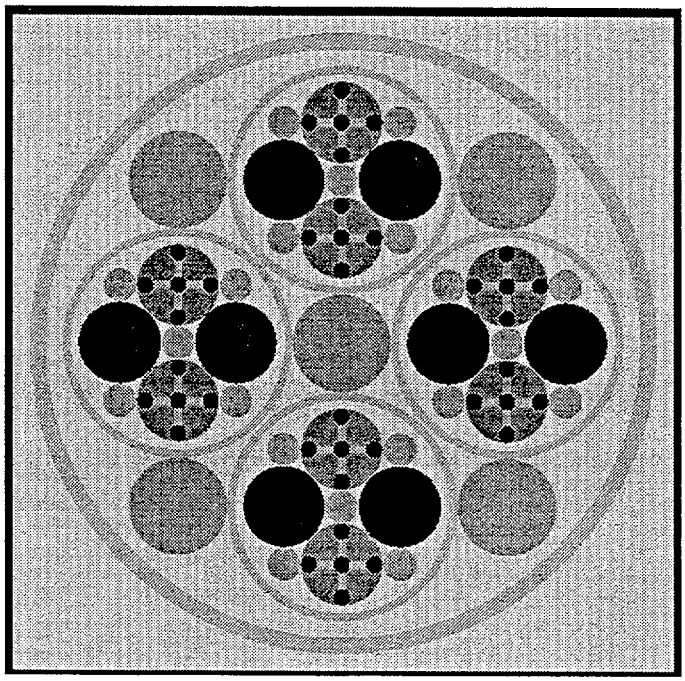


Figure F17.5.61 Mixture map of nested holes problem

```
x-y slice at z midpoint.  nested holes.  unit map

                                        unit map

   unit  1 2 3 4 5 6 7 8 9
 symbol  1 2 3 4 5 6 7 8 9


      overall system coordinates:
       xmin= 0.00000E+00   xmax= 8.00000E+00   ymin= 0.00000E+00
       ymax= 8.00000E+00   zmin= 0.00000E+00   zmax= 3.20000E+01
      upper  left          lower right
      coordinates          coordinates
x     -1.0000E-01           8.1000E+00
y      8.1000E+00          -1.0000E-01
z      1.6000E+01           1.6000E+01
          u axis      v axis
          (down)      (across)
x         0.00000      1.00000
y        -1.00000      0.00000
z         0.00000      0.00000
nu=  640   nv=  640    delu= 1.2813E-02     delv= 1.2813E-02     lpi=  10.000
```

Figure F17.5.62 Plot data for unit map of nested holes

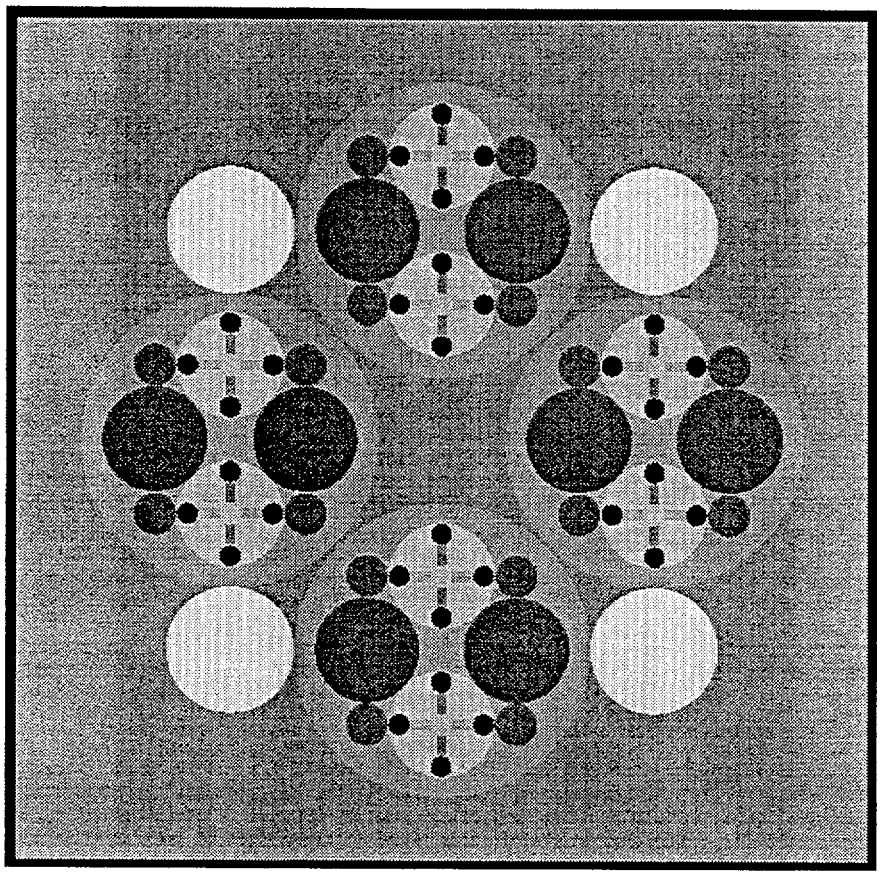


Figure F17.5.63 Unit map of nested holes problem

```
=KENOVI
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 0.1 2P15.0
CYLINDER 2 0.1 2P15.0 ORIGIN Y=-0.4
CYLINDER 3 0.1 2P15.0 ORIGIN X=0.4
CYLINDER 4 0.1 2P15.0 ORIGIN Y=0.4
CYLINDER 5 0.1 2P15.0 ORIGIN X=-0.4
CUBOID 6 2P0.1 2P0.05 2P15.0 ORIGIN X=-0.2
CUBOID 7 2P0.1 2P0.05 2P15.0 ORIGIN X=0.2
CUBOID 8 2P0.05 2P0.1 2P15.0 ORIGIN Y=-0.2
CUBOID 9 2P0.05 2P0.1 2P15.0 ORIGIN Y=0.2
CYLINDER 10 0.5 2P15.0
MEDIA 1 1 1 -6 -7 -8 -9
MEDIA 1 1 2 -8
MEDIA 1 1 3 -7
MEDIA 1 1 4 -9
MEDIA 1 1 5 -6
MEDIA 2 1 6 -1 -5
MEDIA 2 1 7 -1 -3
MEDIA 2 1 8 -1 -2
MEDIA 2 1 9 -1 -4
MEDIA 3 1 -1 -2 -3 -4 -5 -6 -7 -8 -9 10
BOUNDARY  10
UNIT 2
CYLINDER 1 0.2 2P15.0
CYLINDER 2 0.2 2P15.0 ORIGIN X=0.707107 Y=0.707107
CYLINDER 3 0.2 2P15.0 ORIGIN X=-0.707107 Y=0.707107
CYLINDER 4 0.2 2P15.0 ORIGIN X=-0.707107 Y=-0.707107
CYLINDER 5 0.2 2P15.0 ORIGIN X=0.707107 Y=-0.707107
CYLINDER 6 0.5 2P15.0 ORIGIN X=0.707107
CYLINDER 7 0.5 2P15.0 ORIGIN X=-0.707107
CYLINDER 8 0.5 2P15.0 ORIGIN Y=0.707107
CYLINDER 9 0.5 2P15.0 ORIGIN Y=-0.707107
CYLINDER 10 1.3 2P15.0
CYLINDER 11 1.4 2P15.0
MEDIA 2 1 1
MEDIA 2 1 2
MEDIA 2 1 3
MEDIA 2 1 4
MEDIA 2 1 5
MEDIA 1 1 6 -8 -9
MEDIA 1 1 7 -8 -9
HOLE 1 8 -3 -6 ORIGIN Y=0.707107
```

```
HOLE 1 9 -6 -7 ORIGIN Y=-0.707107
MEDIA 0 1 10 -1 -2 -3 -4 -5 -6 -7 -8 -9
MEDIA 4 1 11 -10
BOUNDARY 11
GLOBAL UNIT 3
CYLINDER 1 0.6 2P15.0
CYLINDER 2 0.6 2P15.0 ORIGIN X=2.0 Y=2.0
CYLINDER 3 0.6 2P15.0 ORIGIN X=-2.0 Y=2.0
CYLINDER 4 0.6 2P15.0 ORIGIN X=-2.0 Y=-2.0
CYLINDER 5 0.6 2P15.0 ORIGIN X=2.0 Y=-2.0
CYLINDER 6 1.4 2P15.0 ORIGIN X=2.0
CYLINDER 7 1.4 2P15.0 ORIGIN Y=2.0
CYLINDER 8 1.4 2P15.0 ORIGIN X=-2.0
CYLINDER 9 1.4 2P15.0 ORIGIN Y=-2.0
CYLINDER 10 3.6 2P15.0
CYLINDER 11 3.8 2P15.0
CUBOID 12 4P4.0 2P16.0
MEDIA 2 1 1 -6 -7 -8 -9
MEDIA 2 1 2 -6 -7
MEDIA 2 1 3 -7 -8
MEDIA 2 1 4 -8 -9
MEDIA 2 1 5 -9 -6
HOLE 2 6 -1 -5 -2 ORIGIN X=2
HOLE 2 7 -1 -2 -3 ORIGIN Y=2
HOLE 2 8 -1 -3 -4 ORIGIN X=-2
HOLE 2 9 -1 -4 -5 ORIGIN Y=-2
MEDIA 0 1 10 -1 -2 -3 -4 -5 -6 -7 -8 -9
MEDIA 4 1 11 -10
MEDIA 0 1 12 -11
BOUNDARY 12
END GEOM
READ PLOT
TTL='X-Y SLICE AT Z MIDPOINT. NESTED HOLES'
XUL=-4.1 YUL=4.1 ZUL=0.0 XLR=4.1 YLR=-4.1 ZLR=0
UAX=1.0 VDN=-1.0 NAX=640 END
TTL='X-Y SLICE AT Z MIDPOINT. NESTED HOLES, UNIT MAP.'
PIC=UNIT END
END PLOT
END DATA
END
```

The plot data description for unit map of nested holes shown in Fig. F17.5.62 shows the symbols that represent each unit in the geometry mock-up. The unit map is shown in Fig. F17.5.63. The user can utilize this map to verify the correct placement of the units. Note that the unit map plots the units that are present at the deepest nesting level for the 2-D slice. It does not show any detail within a unit. In the legend of the plot, the material numbers actually refers to the unit number.

EXAMPLE 7. LARGE STORAGE ARRAY

The storage array described in Sect. F17.5.6.3 and Fig. F17.5.15 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. F17.5.6.3, 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.

```
READ PLOT PLT=YES PIC=UNIT
TTL='X-Z SLICE AT Y=30.48 WITH Z ACROSS AND X DOWN'
XUL=624.84 YUL=30.48 ZUL=-15.0 XLR=-30.48 YLR=30.48 ZLR=201.0
WAX=1.0 UDN=-1.0 NAX=320 END
TTL='X-Y SLICE OF ROOM THROUGH SHELF Z=0.3175 WITH X ACROSS AND Y DOWN'
XUL=-30.48 YUL=1341.0 ZUL=0.3175 XLR=624.84 YLR=-30.4 ZLR=0.3175
UAX=1.0 VDN=-1.0 NAX=320 END
END PLOT
```

The plot data and unit map for an x-z slice through the array at y = 30.48 cm are given in Figs. F17.5.64 and F17.5.65. The z direction, which extends from the bottom of the reflector at -30.84 cm to just about the array at 201.0 cm, is printed in 320 pixels. This unit map was created with z across the page and x down the page.

```
x-z slice through storage array room 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      6.2484E+02          -3.0480E+01
y      3.0480E+01           3.0480E+01
z     -4.5720E+01           3.8100E+02
          u axis      v axis
          (down)      (across)
x        -1.00000      0.00000
y         0.00000      0.00000
z         0.00000      1.00000
nu=  491   nv=  320     delu= 1.3335E+00     delv= 1.3335E+00     lpi=  10.000
```

Figure F17.5.64 Plot data for x-z slice of storage array

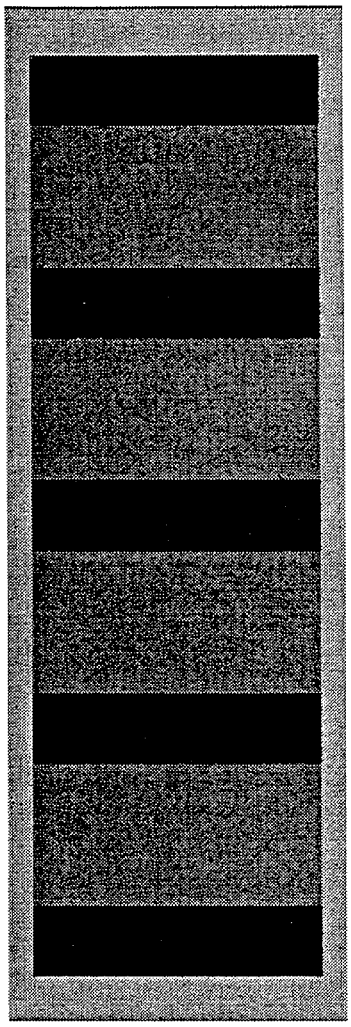


Figure F17.5.65 x-z plot of storage array

The plot data and unit map for an x-y slice through the shelf are given in Figs. F17.5.66 and F17.5.67. This unit map was created with x across the plot and y down the plot. This shows 5 rows of shelves in the x direction.

```
x-y slice through storage array room at 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     -3.0480E+01           6.2484E+02
y      1.3411E+03          -3.0480E+01
z      3.1750E-01           3.1750E-01
         u axis       v axis
         (down)       (across)
x        0.00000       1.00000
y       -1.00000       0.00000
z        0.00000       0.00000
nu=  669   nv=  320     delu= 2.0479E+00     delv= 2.0479E+00     lpi=  10.000
```

Figure F17.5.66 Plot data for x-y slice of storage array

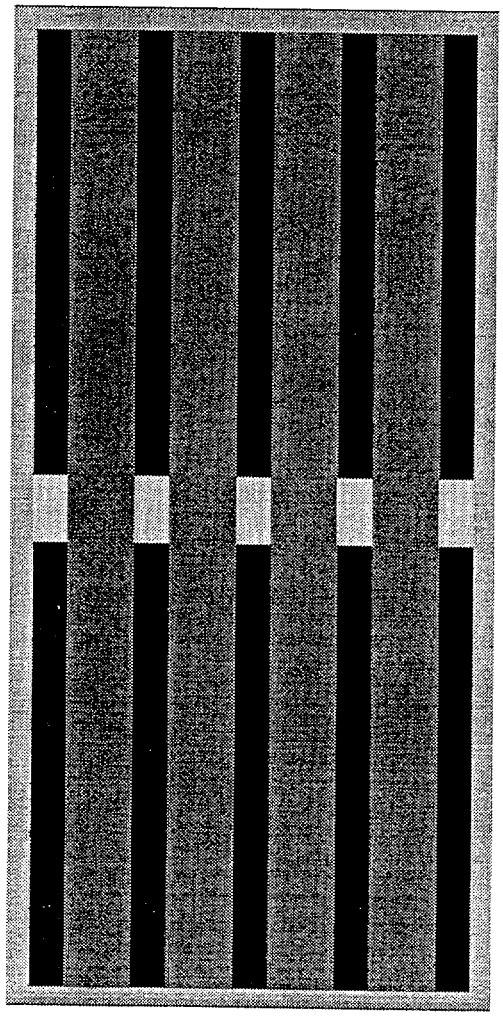


Figure F17.5.67 x-y plot of storage array

F17.5.10 RESTART CAPABILITIES

Restart data can be written and used for restarting a problem. This type of data is saved by specifying file definition data 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 1.

Parameters that can be changed when a problem is restarted at a generation greater than 1 include: RND=, TME=, TBA=, GEN=, RES=, LNG=, BEG=, AMX=, XAP=, XS1=, XS2=, PKI=, P1D=, CKU=, CKP=, CKH=, CKA=, FMU=, FMP=, FMH=, FMA=, BUG=, TRK=, PWT=, PGM=, RUN=, PLT=, NB8=, NL8=, and PAX=. All 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 written to a particular restart file 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 written to a particular restart file 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 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 data 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 data are 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 procedure 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-VI will automatically pull the job before it runs out of time or I/Os. 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.

If the user wishes to run additional histories to improve convergence, BEG should be set to 1 greater than the last generation saved on the restart file. For example, if a problem was originally run with GEN=100, RES=100 WRS=35, then the restart problem should specify BEG=101 GEN=200 RST=35, in order to run an additional 100 generations.

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 1.

Note that if a problem is restarted from one unit and restart data are written to another unit, the new problem description becomes the "original problem" for the new restart dataset.

Parameter data can be changed by entering new values for the desired options, which has the effect of inputting additional data within the parameter data block as explained in Sect. F17.5.3. 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 ARA=1 NUX=11 NUY=5 NUZ=3 END ARRAY and the problem is to be restarted at the first generation as an $11 \times 5 \times 6$ array, the following data block must be entered: READ ARRAY ARA=1 NUX=11 NUY=5 NUZ=6 END ARRAY.

Section F17.2.6 contains some information pertaining to the restart capability. Other information may be found in Sect. F17.4.3. The structure of the restart file is listed in Table F17.5.2, and the variables referenced in that table are listed in Table F17.5.3.

F17.5.11 RANDOM SEQUENCE

The random number package utilized by KENO-VI 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-VI 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 F17.5.2 Structure of RESTART file

Record No.	Contents
(1)	TITLE(20), NBA, NPB, NSKIP, NRSTRT, NBANK, NFBNK, NXBNK, NXFBK, NUMX1D, RFLT(6),RNDNUM, LOG(33) (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 record of data (These records contain the data associated with the specified type of data.) NDX = 1 (geometry data) (b1) LLNGTH, KMAX, NQUAD, MAXSFU, REGTOT, SECTOT, MAXREG, NIAR, NGWRDS, MAXGWD, MAXSEC, NGBLU, NBOXT, MAXMIX, MAXIMP, NUMHOL, NUCOM, EXRFL (b2) MAT(REGTOT), IMP(REGTOT), IGEOM(REGTOT), IGWDPT(0:NGWRDS), ARANDX(REGTOT), NSURFT(NBOXT), NSU(NBOXT), KBNDSQ((NBOXT), KBNDSM(NBOXT), KBNDSX(NBOXT), KBNDSG(NBOXT), NGWPU(NBOXT), IBNDS(NBOXT), BOXBND(3,NBOXT), ARABND(3,NIAR), IRFPOS(3,NIAR), QC(NQUAD,11), REGION(NAXGWD, REGTOT), SECTOR(SECTOT), HOLROT(3,3,NUMHOL), KHOLE(NUMHOL), LHOLU(NUMHOL), HOLX(NUMHOL), HOLY(NUMHOL), HOLZ(NUMHOL), ICOMC(NBOXT), UCOMNT(33,NUCOM) NDX = 2 (array data) (b1) LLNGTH, NGLOBL, MAXARA, NACOM, LSGUN, MBOX, 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.)

Table F17.5.2 (continued)

Record No.	Contents
	(b3) LBA(NBXMAX(I),NBYMAX(I),NBZMAX(I))
	NDX = 3 (mixing table data)
	(b1) LLNGTH, NMIX, NSCT, MIX, MIXT, NPL, PBXS
	(b2) MIXTUR(NMIX), NUC(NMIX), DEN(NMIX)
	NDX = 4 (extra data)
	(b1) LLNGTH, LNEXTR
	(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
	(b2) IF (NTYPST .EQ. 6) NUBANK(LFIN,LBANK0)
	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

Table F17.5.2 (continued)

Record No.	Contents
	(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, NMAT, NGP, MAXANG
	(b2) LXS(MANG,MATT) MANG = 2*NSCT + 4
	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-Ds
	(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, NSTOR(49)

Table F17.5.2 (continued)

Record No.	Contents
	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 (a) IGEN, RND, NPB, NGP, KMAX, LBANK, NBANK, DLIF(15), LIF(20), LOJIC(36), EFFK(I),I=1,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,3) (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.)

Table F17.5.2 (continued)

Record No.	Contents
	(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 F17.5.3 Key of RESTART file variables

TITLE	80-character KENO-VI problem title
NBA	Number of generations in the KENO-VI 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
RFLT	Contains tmax, tbtch, dwtav, wthigh, wtlow, tsigma
TBTCH	Time allowed for each generation
RNDNUM	Random number with which the problem will be started
LOG	First 33 positions of 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
NQUAD	Number of quadratic equations in the problem
MAXSFU	Maximum number of surfaces in a unit
REGTOT	Number of MEDIA type cards read in the geometry data
SECTOT	Number of sectors x number of geometry cards in a unit summed over all units
MAXREG	Maximum number of regions in a unit
NIAR	Number of arrays read in the problem
NGWRDS	Number of geometry words in the problem
MAXGWD	Largest number of geometry words in a unit
MAXSEC	Largest number of sectors in a unit
NGBLU	The global unit number
NBOX	Largest unit number in the geometry data
NBOXT	NBOX + the number extra units generated by KENO-VI
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
NUCOM	The number of units having comments in the geometry region data
EXRFL	Logical flag. Value is TRUE if a reflector is present
MAT	Array containing the mixtures used in the geometry
IMP	Array containing the bias IDs used in the geometry
IGWDPT	Array containing the offset to the last quadratic for each geometry word in QC
ARANDX	Array containing the offset for each array into ARABNDS

Table F17.5.3 (continued)

NSURFT	Array containing the number of quadratic equations in each unit
NSU	Array containing the number of sectors in each unit
KBNDSQ	Array of offsets to the first quadratic of each unit in the QC array
KBNDSM	Array of pointers into the SECTOR array
NKBNDSX	Array of offsets of the first entry for each unit into the SECTOR array
KBNDSG	Array of offsets of the first entry of each unit into the IGWDPT array
NGWPU	Array containing the number of geometry words in each unit
IBNDS	Array of offsets into IGWDPT to first boundary quadratic in a unit
BOXBND	Array containing the most negative boundary for each unit
ARABND	Array containing the most negative boundary for each array
IRFPOS	Array containing the array position of the reference unit used to postion the array
QC	Array containing the coefficients of the quadratic equations for each surface
REGION	Array of sector data based on geometry input cards
SECTOR	Array containing information relating each sector to each geometry card for each unit
HOLROT	Array containing rotation data for each hole
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
ICOMC	The index into the comment array for a unit
UCOMNT	Comments associated with units specified in the geometry region data
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
NACOM	The number of arrays having comments in the array data
LSGUN	Logical flag. Value is TRUE if there is no array present
MBOX	A logical variable that is set .TRUE. if arrays are used.
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, 2 for triangular-pitched arrays
LPT	Pointer to locate the beginning of each unit orientation array
LNG	Length of each unit orientation array
ICOMA	The index into the comment array for arrays
LBA	Contains the unit orientation arrays for all arrays
MAXARA	The largest array number

Table F17.5.3 (continued)

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
MIXTUR	Array of mixture numbers used in the mixing table
NUC	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
LNXTR	Length of the extra data
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
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

Table F17.5.3 (continued)

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
NUBANK	The neutron bank. The variables that are used from the neutron bank are: X(NPB),Y(NPB),Z(NPB),NBX(NPB),NBY(NPB),NBZ(NPB)
LFIN	The last neutron for which start type 6 data was entered
LBANK	The number of positions per neutron in the neutron bank
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
A	Probabilities for the returning angle
LENG	Length of albedo data for a given angle
NDX = 8	Mixed cross sections
LLNGTH	Length of b2 record
NMAT	Number of mixtures used in the problem
NGP	Number of energy groups from the cross-section file

Table F17.5.3 (continued)

MAXANG	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
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
NSTOR	An array that contains PTITL and TABLE. The first 33 words are PTITL and the last 16 are TABLE.
NDX = 11	Biasing data

Table F17.5.3 (continued)

LLNGTH	Length of b2 record
NUMIDS	Number of bias IDs requested
NCS	Number of bias IDs entered from cards
NTSETS	Total number of group structures for all IDs
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
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
Calculated data	
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
DLIF	The double precision portion of COMMON/LIFETM/
LIF	The single precision portion of 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

Table F17.5.3 (continued)

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(3)	Key indicating whether to calculate fluxes Specified by entering FLX= in the parameter data

F17.5.12 MATRIX K-EFFECTIVE

Matrix k-effective calculations provide an alternative 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-VI 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 3-D lattice. For a 2 × 2 × 2 array there are eight unique position indices as follows:

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 4 × 4 × 4 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. It is the number of next-generation neutrons produced in unit J by a neutron born in unit I. This matrix is used to calculate the matrix k-effective, cofactor k-effectives and source vector by unit.

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 calculate matrix information for a single unit the geometry data must include a second global unit that contains a 1 × 1 × 1 array composed of the first unit. An array data block consisting of READ ARRAY ARA=1 NBX=1 NBY=1 NBZ=1 FILL 1 END FILL END ARRAY must also be included. These actions convert the single unit problem into a 1 × 1 × 1 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.

F17.5.13 DEVIATIONS

When a deviation is calculated by KENO-VI, it is the standard deviation of the mean. A large sample and a normal distribution are assumed for this calculation. If these assumptions are not true, the deviations cannot be assumed to be correct. If results are correlated, the calculated deviations will be in error.

F17.5.14 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. The lifetime and generation time calculated by KENO-VI are not kinetics parameters. 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.

F17.5.15 REFERENCES

1. S. K. Fraley, *Users Guide for ICE-II*, ORNL/CSD/TM-9/R1, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., July 1977.

2. J. T. Thomas, "Critical Three-Dimensional Arrays of U(93.2)-Metal Cylinders," *Nucl. Sci. Eng.* **52**, 350 (November 1973).

3. J. T. Thomas, *Critical Three-Dimensional Arrays of Neutron-Interacting Units, Part II*, ORNL/TM-868, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., July 1964.

F17.6 DESCRIPTION OF OUTPUT

This section contains a brief description and explanation of the KENO-VI output. Portions of the printout will not be printed for every problem. Some printout is optional and is so noted in this section.

F17.6.1 HEADER PAGE

The first page of print from KENO-VI is the header page. A sample header page is shown in Fig. F17.6.1.

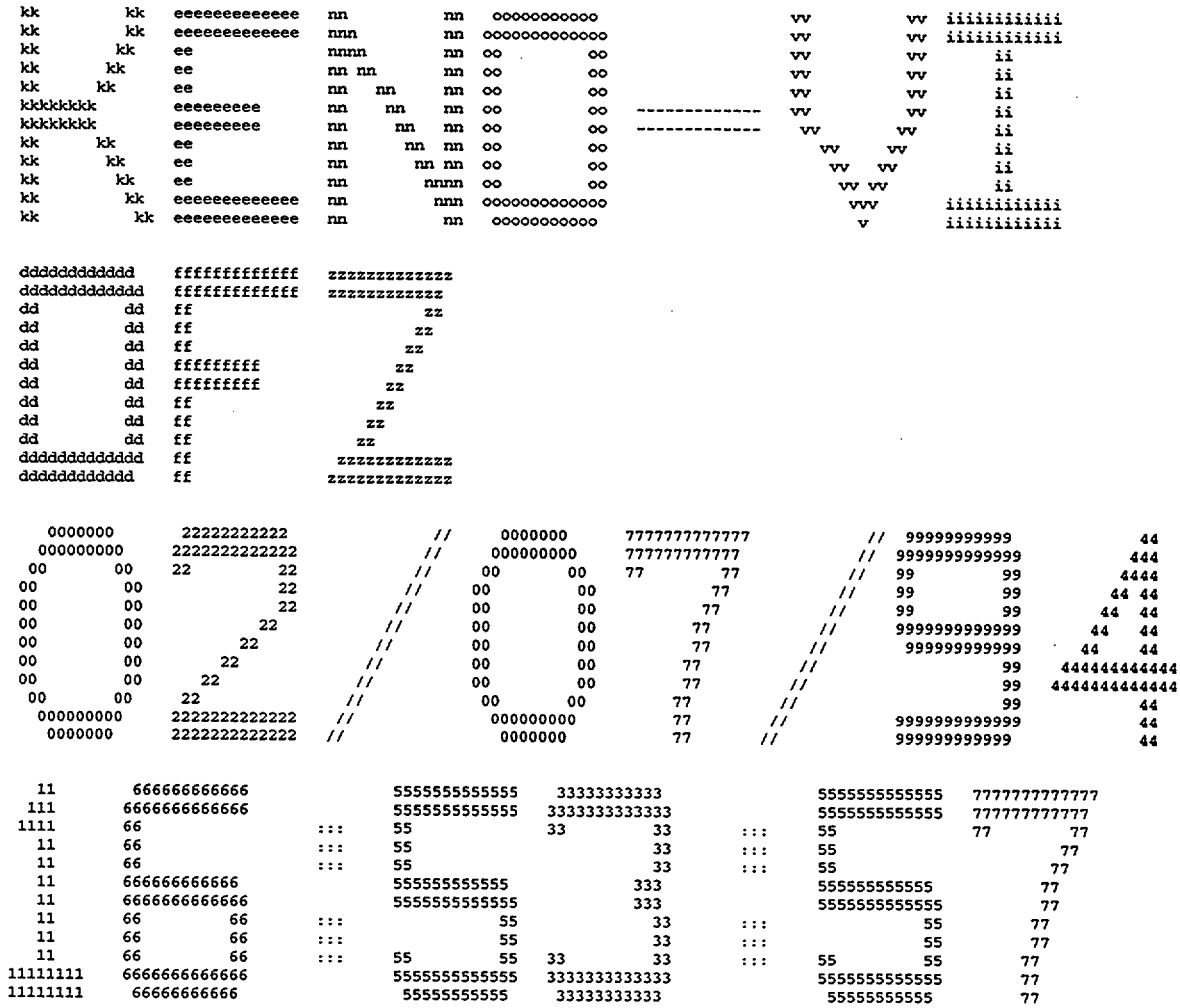


Figure F17.6.1 Sample KENO-VI header page

The first line of block letters prints KENO-VI. 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 when the execution of the problem began (hour/minute/second) in terms of a 24-h clock, with midnight being 2400 h. The header page is printed from subroutine MESAGE.

F17.6.2 PROGRAM VERIFICATION INFORMATION

Program verification information (Fig. F17.6.2) is printed after the header page. It lists the name of the program, the date the load module was created, the library that contains the load module, the computer code name from the configuration control table, and the revision number. The job name, date, and time of execution are also printed. This information may be used for quality assurance purposes.

```
*********************************************************************************
*********************************************************************************
*********************************************************************************
*****                                                                       *****
*****                  program verification information                     *****
*****                                                                       *****
*****                 code system:    scale  version:    4.2               *****
*****                                                                       *****
*********************************************************************************
*********************************************************************************
*****                                                                       *****
*****                                                                       *****
*****               program:  kenovi                                        *****
*****                                                                       *****
*****        creation date:  02/07/94                                       *****
*****                                                                       *****
*****               library:  /home/nfc/kenovi/bin                          *****
*****                                                                       *****
*****                                                                       *****
*****    this is not a   scale  configuration controlled code               *****
*****                                                                       *****
*****               jobname:  dfz                                           *****
*****                                                                       *****
*****    date of execution:  02/07/94                                       *****
*****                                                                       *****
*****    time of execution:  16:53:57                                       *****
*****                                                                       *****
*****                                                                       *****
*********************************************************************************
*********************************************************************************
*********************************************************************************
```

Figure F17.6.2 Sample program verification table

F17.6.3 TABLES OF PARAMETER DATA

The first two tables printed by KENO-VI 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. F17.6.3 and F17.6.4.

sample problem 18 1f27 critical experiment

****** numeric parameters ******

tme	maximum problem time (min)	30.00
tba	time per generation (min)	.50
gen	number of generations	53
npg	number per generation	350
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	375
xnb	extra positions in neutron bank	0
nfb	fission bank size	350
xfb	extra positions in fission bank	0
sig	cut off standard deviation	.0000
wta	default value of weight average	.5000
wth	weight high for splitting	3.0000
wtl	weight low for russian roulette	.3333
rnd	starting random number	00000BB827100001
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

Figure F17.6.3 Sample table of numeric parameter data

```
                                sample problem 18   1f27  critical experiment

                                    ******      logical parameters      ******

  run  execute problem after checking data    yes          plt  plot map(s)                          yes
  flx  compute flux                           yes          fdn  compute fission densities            yes
  smu  compute avg unit self-multiplication    no          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  no
  cku  compute cofactor k-eff by unit number  yes          ckp  compute cofactor k-eff by unit location  no
  fmu  print fiss prod matrix by unit number  yes          fmp  print fiss prod matrix by unit location  no
  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          far  print fis. and abs. by region         yes
  xs1  print 1-d mixture x-sections           yes          pax  print xsec-albedo correlation tables  yes
  xs2  print 2-d mixture x-sections           yes          pwt  print weight average array           yes
  xap  print mixture angles & probabilities   yes          pgm  print input geometry                 yes
  pki  print fission spectrum                 yes          bug  print debug information               no
  pld  print extra 1-d cross sections         yes          trk  print tracking information          no

                                parameter input completed

.....    0 io's were used reading the parameter data      .....
```

Figure F17.6.4 Sample table of logical parameter data

The title of the problem is printed at the top of each table. The first table, shown in Fig. F17.6.3, 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. F17.6.4, 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 the completion of the parameter input and a statement of the number of I/Os used in reading the parameter data.

At this point, the unprocessed input geometry, shown in Fig. F17.6.5, may be printed as described in Sect. F17.6.4. These data are followed by a statement affirming the completion of the data reading and a table listing the geometry parameter summary data, as shown in Fig. F17.6.6.

F17.6.4 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. F17.4.3. Standard KENO-VI use does not require printing these data because the processed geometry that is used in the problem is always printed. See Sects. F17.6.15 and F17.6.16 for examples of the standard printed KENO-VI geometry data. An example of the data printed in the unprocessed geometry input data is shown in Fig. F17.6.5. 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 and a statement affirming the completion of the data input. A summary table of the geometry input data is then printed, as shown in Fig. F17.6.6. This table contains summations of the different geometry parameters used in the problem such as number of units, quadratic equation, geometry words, and holes.

```
                        sample problem 18   1f27  critical experiment
                                                  geometry description input
  1 cylinder          radius =  9.5200     +z=  8.7804     -z= -8.7804
  2 cylinder          radius =  9.5200     +z=  8.9896     -z= -8.7804
  3 cylinder          radius =  10.160     +z=  9.6296     -z= -9.4204
  4 cuboid              +x =  18.450     -x = -18.450      +y =  18.450     -y = -18.450     +z =  17.895     -z = -17.685
sector data associated with media   1 and importance region   1
    1
icnt/irgsec(ii),ii=1,icnt    1    1
sector data associated with media   0 and importance region   1
   -1    2
icnt/irgsec(ii),ii=1,icnt    2   -1    2
sector data associated with media   2 and importance region   1
   -1   -2    3
icnt/irgsec(ii),ii=1,icnt    3   -1   -2    3
sector data associated with media   0 and importance region   1
    4   -2   -3
icnt/irgsec(ii),ii=1,icnt    3    4   -2   -3
  1 cuboid              +x =  18.440     -x = -55.340      +y =  55.340     -y = -18.440     +z =  53.360     -z = -17.780
  2 cuboid              +x =  18.440     -x = -55.340      +y =  55.340     -y = -18.440     +z = -17.800     -z = -53.360
  3 cuboid              +x =  55.340     -x =  18.460      +y =  55.340     -y = -18.440     +z =  53.360     -z = -53.360
  4 cuboid              +x =  55.340     -x = -55.340      +y = -18.460     -y = -55.340     +z =  53.360     -z = -53.360
  5 cuboid              +x =  55.350     -x = -55.350      +y =  55.350     -y = -55.350     +z =  53.370     -z = -53.370
sector data associated with array   1
 the origin of the unit at array position (  1,  1,  1)
           is positioned at x= -36.900     y= 0.00000E+00 z= -.10460     relative to the origin of the unit.
    1
icnt/irgsec(ii),ii=1,icnt    1    1
sector data associated with array   2
 the origin of the unit at array position (  1,  1,  1)
           is positioned at x= -36.900     y= 0.00000E+00 z= -35.685     relative to the origin of the unit.
    2   -1
icnt/irgsec(ii),ii=1,icnt    2    2   -1
sector data associated with array   3
 the origin of the unit at array position (  1,  1,  1)
           is positioned at x=  36.900     y= 0.00000E+00 z= -35.685     relative to the origin of the unit.
    3   -2   -1
icnt/irgsec(ii),ii=1,icnt    3    3   -2   -1
sector data associated with array   4
 the origin of the unit at array position (  1,  1,  1)
           is positioned at x= -36.900     y= -36.900     z= -35.685     relative to the origin of the unit.
    4   -3   -2   -1
icnt/irgsec(ii),ii=1,icnt    4    4   -3   -2   -1
sector data associated with media   0 and importance region   1
    5   -4   -3   -2   -1
icnt/irgsec(ii),ii=1,icnt    5    5   -4   -3   -2   -1
  1 cuboid              +x =  55.350     -x = -55.350      +y =  55.350     -y = -55.350     +z =  53.370     -z = -53.370
  2 cuboid              +x =  58.350     -x = -58.350      +y =  58.350     -y = -58.350     +z =  56.370     -z = -56.370
  3 cuboid              +x =  61.350     -x = -61.350      +y =  61.350     -y = -61.350     +z =  59.370     -z = -59.370
  4 cuboid              +x =  64.350     -x = -64.350      +y =  64.350     -y = -64.350     +z =  62.370     -z = -62.370
  5 cuboid              +x =  67.350     -x = -67.350      +y =  67.350     -y = -67.350     +z =  65.370     -z = -65.370
  6 cuboid              +x =  70.590     -x = -70.590      +y =  70.590     -y = -70.590     +z =  68.610     -z = -68.610
sector data associate with hole   1 containing unit   2
    1
sector data associated with media   3 and importance region   2
   -1    2
icnt/irgsec(ii),ii=1,icnt    2   -1    2
sector data associated with media   3 and importance region   3
   -2    3
icnt/irgsec(ii),ii=1,icnt    2   -2    3
sector data associated with media   3 and importance region   4
   -3    4
icnt/irgsec(ii),ii=1,icnt    2   -3    4
sector data associated with media   3 and importance region   5
   -4    5
icnt/irgsec(ii),ii=1,icnt    2   -4    5
sector data associated with media   3 and importance region   6
    6   -5
icnt/irgsec(ii),ii=1,icnt    2    6   -5
```

Figure F17.6.5 Example of unprocessed geometry input data

```
                                ******      geometry parameters      ******

                niar        number of independent array references          4
                ngblu       global unit number                              3
                nboxt       number of units in the problem                  3
                nquad       number of quadratics in the problem            42
                ngwrds      number of geometry words read                  15
                maxgwd      maximum geometry words in a unit                6
                maxsfu      largest number of surfaces in a unit           18
                maxreg      largest number of media in a unit               6
                regtot      number of spatial volumes defined              15
                sectot      number of entries in the sector array          77
                nucom       number of comments in the geometry data         0
                numhol      number of holes in the problem                  1

               *************** data reading completed ***************
```

Figure F17.6.6 Sample table of geometry input data

```
                          sample problem 18   1f27  critical experiment

      unit                                    volume
     number        data set name              name       unit function

   xsc  14    ft14f001                                   mixed cross sections
   alb  79    /scale/datalib/albedos                     input albedos
   wts  80    /scale/datalib/weights                     input weights
   skt  16    unknown                                    write scratch data
   lib  41    /home/lmp/scale/data/hrxsecs               input ampx working library
         8    ft08f001                                   input data direct access
         9    ft08f001                                   super grouped direct access
        10    ft08f001                                   xsec mixing direct access

               ........    0 io's were used preparing input data    ........
```

Figure F17.6.7 Sample table of data sets used in the problem

F17.6.5 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. For UNIX systems the volume name is left blank. 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/Os used preparing the input data and writing it on the direct access data sets. It should be carefully scrutinized to verify that the desired data set name is associated with the proper unit number and volume. An example of this table is shown in Fig. F17.6.7.

F17.6.6 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. F17.6.8.

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 of each entry, printed in columnar form. After all the entries have been printed, the number of I/Os used in mixing the cross sections is printed. If extra 1-D cross sections were specified in the problem (see X1D=, Sect. F17.4.3), the extra 1-D cross-section IDs will be printed under the heading "1-D Cross-Section Array ID Numbers." If $\bar{\nu}$ is to be calculated (see NUB=, Sect. F17.4.3), six ID numbers will be printed. The ID number for the total cross section, Σ_T, is 1; the ID number for the sum of the transfer array normalized by Σ_T is 2002; the ID number for $\nu\Sigma_f$ is 1452; the normalized absorption cross section, Σ_{abs}/Σ_T, ID number is 27; the normalized fission cross section, Σ_f/Σ_T, ID number is 18; the fission spectrum, χ, ID number is 1018. χ is summed and normalized to 1.0. Other ID numbers that appear in this list have been specified by the user. 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. F17.4.3), a message is printed stating that restart information was written and the restart I/O unit number is specified. This sequence is followed by a statement of the number of I/Os used in preparing the cross sections. 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 IDs and their titles. This information is printed by subroutine MIXMIX. The 1-D cross-section array ID numbers are then printed.

```
                    sample problem 18   1f27  critical experiment
                          mixing table
                           number of scattering angles =  1

                       cross section message threshold =3.0E-05

                       entry   mixture   nuclide      density

                         1        1       92508     9.84350E-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-02
                         8        3        6100     3.96970E-02
                         9        4         502     1.00000E-09

                    cross sections read from the ampx working library on unit 41
  502    de/e water  0.9982g/cc  h=.066742, o=.033371
  602    de/e plexiglas  1.182g/cc  h=.056884, c=.035552, o=.014221
 1102    hydrogen de/e                    hansen roach
 6100    carbon                           hansen roach
 7100    nitrogen                         hansen roach
 8100    oxygen                           hansen roach
92508    u-235-8r    sig p = 1000         hansen roach
92852    u-238 sig p = 20000              hansen roach
                  ........    0 io's were used mixing cross-sections       ........
                   1-d cross section array id numbers
                      1  2002  1452    27    18  1018
                 ........    0 io's were used preparing the cross sections    ........
```

Figure F17.6.8 Example of mixing table data

F17.6.7 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. F17.4.3. Examples of these tables are shown in Figs. F17.6.9 and F17.6.10.

The table shown in Fig. F17.6.9 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. F17.6.10 is the inverse of the table shown in Fig. F17.6.9. 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-VI.

cumulative probabilities for choosing the corresponding albedo group for each xsec group

xsec energy group	cross section group lethargy boundaries		albedo group	albedo probability	albedo group	albedo probability	albedo group	albedo probability
1	.00000E+00	.12040E+01	1	1.00000				
2	.12040E+01	.19661E+01	2	1.00000				
3	.19661E+01	.24079E+01	3	1.00000				
4	.24079E+01	.32189E+01	4	1.00000				
5	.32189E+01	.46052E+01	5	1.00000				
6	.46052E+01	.63771E+01	6	1.00000				
7	.63771E+01	.81117E+01	7	1.00000				
8	.81117E+01	.98082E+01	8	1.00000				
9	.98082E+01	.11513E+02	9	1.00000				
10	.11513E+02	.12717E+02	10	1.00000				
11	.12717E+02	.13816E+02	11	1.00000				
12	.13816E+02	.15019E+02	12	1.00000				
13	.15019E+02	.16118E+02	13	1.00000				
14	.16118E+02	.17034E+02	14	1.00000				
15	.17034E+02	.18421E+02	15	1.00000				
16	.18421E+02	.19807E+02	16	1.00000				

Figure F17.6.9 Cumulative probabilities for correlating the albedo energy group to the cross-section energy group

cumulative probabilities for choosing the corresponding xsec group for each albedo group

albedo energy group	albedo group lethargy boundaries		xsec group	xsec probability	xsec group	xsec probability	xsec group	xsec probability	xsec group	xsec probability
1	-.40547E+00	.12040E+01	1	1.00000						
2	.12040E+01	.19661E+01	2	1.00000						
3	.19661E+01	.24079E+01	3	1.00000						
4	.24079E+01	.32189E+01	4	1.00000						
5	.32189E+01	.46052E+01	5	1.00000						
6	.46052E+01	.63771E+01	6	1.00000						
7	.63771E+01	.81117E+01	7	1.00000						
8	.81117E+01	.98082E+01	8	1.00000						
9	.98082E+01	.11513E+02	9	1.00000						
10	.11513E+02	.12717E+02	10	1.00000						
11	.12717E+02	.13816E+02	11	1.00000						
12	.13816E+02	.15019E+02	12	1.00000						
13	.15019E+02	.16118E+02	13	1.00000						
14	.16118E+02	.17034E+02	14	1.00000						
15	.17034E+02	.18421E+02	15	1.00000						
16	.18421E+02	.23026E+02	16	1.00000						

Figure F17.6.10 Cumulative probabilities for correlating the cross-section energy group to the albedo energy group

F17.6.8 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. F17.4.3. 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. F17.6.11.

When the 1-D mixture cross sections are printed, the problem title is printed at the top of the page. The mixture ID and MIXTURE INDEX numbers are then printed. ID is the mixture number from the mixing table, and MIXTURE INDEX is the index used to reference it, 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, ABP is the absorption 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, $\upsilon\Sigma_f/\Sigma_T$. The nonabsorption probability and the production probability are not true probabilities in that they may be greater than 1. 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.

F17.6.9 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. F17.4.3). Extra 1-D cross sections are not used in KENO-VI 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. F17.6.11. The fission cross-section heading is XSEC ID 18, and it follows the table of 1-D cross sections.

F17.6.10 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. F17.6.12.

F17.6.11 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. F17.4.3. The probabilities and angles are printed for each mixture by subroutine PRT2DS. An example of the probabilities is shown in Fig. F17.6.13, and an example of the angles is shown in Fig. F17.6.14. 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.

sample problem 18 1f27 critical experiment

mixture id = 1 mixture index = 1

group	sgt	nap	abp	nfp	chi	mwa1	mwa2	mwa3	sum
1	1.39944E-01	9.76695E-01	2.33045E-02	2.59767E-02	2.04000E-01	1	6	1	1.00000E+00
2	1.77770E-01	9.91346E-01	8.65391E-03	1.82268E-02	5.48000E-01	7	12	2	1.00000E+00
3	3.19702E-01	9.95623E-01	4.37705E-03	9.53099E-03	7.16000E-01	13	18	3	1.00000E+00
4	3.83997E-01	9.96305E-01	3.69464E-03	7.65954E-03	8.96000E-01	19	24	4	1.00000E+00
5	5.44520E-01	9.96969E-01	3.03092E-03	6.35962E-03	9.86000E-01	25	30	5	1.00000E+00
6	8.24261E-01	9.96191E-01	3.80898E-03	7.31460E-03	1.00000E+00	31	36	6	1.00000E+00
7	9.89720E-01	9.94467E-01	5.53306E-03	1.02342E-02	1.00000E+00	37	42	7	1.00000E+00
8	1.04405E+00	9.89315E-01	1.06849E-02	1.82530E-02	1.00000E+00	43	48	8	1.00000E+00
9	1.06120E+00	9.73740E-01	2.62602E-02	4.24972E-02	1.00000E+00	49	54	9	1.00000E+00
10	1.09463E+00	9.45115E-01	5.48852E-02	8.26186E-02	1.00000E+00	55	60	10	1.00000E+00
11	1.09554E+00	9.44429E-01	5.55713E-02	7.55643E-02	1.00000E+00	61	66	11	1.00000E+00
12	1.10314E+00	9.37344E-01	6.26558E-02	6.09451E-02	1.00000E+00	67	71	12	1.00000E+00
13	1.07433E+00	9.61400E-01	3.85995E-02	6.73441E-02	1.00000E+00	72	75	13	1.00000E+00
14	1.28920E+00	9.35180E-01	6.48206E-02	1.30946E-01	1.00000E+00	76	78	14	1.00000E+00
15	1.83807E+00	8.75203E-01	1.24797E-01	2.42731E-01	1.00000E+00	79	80	15	1.00000E+00
16	3.38272E+00	8.16191E-01	1.83809E-01	3.67874E-01	1.00000E+00	81	81	16	1.00000E+00

xsec id	18 grp.
1	8.85189E-03
2	6.96651E-03
3	3.76749E-03
4	3.07612E-03
5	2.58521E-03
6	2.98555E-03
7	4.17721E-03
8	7.45021E-03
9	1.73458E-02
10	3.37219E-02
11	3.08426E-02
12	2.48756E-02
13	2.74874E-02
14	5.34473E-02
15	9.90739E-02
16	1.50153E-01

Figure F17.6.11 Example of macroscopic 1-D cross sections

scattering transfer array for material 1

from grp / to grp	grp 1	grp 2	grp 3	grp 4	grp 5	grp 6	grp 7	grp 8	grp 9	grp 10
+ 0	2.54209E-01	3.10539E-01	2.98182E-01	3.71084E-01	4.55319E-01	4.72411E-01	4.56823E-01	4.47266E-01	4.48962E-01	3.30557E-01
+ 1	4.48688E-01	2.73107E-01	4.37470E-01	4.85849E-01	4.56102E-01	4.37482E-01	4.45877E-01	4.54113E-01	3.88724E-01	4.51241E-01
+ 2	1.04476E-01	2.31550E-01	1.97693E-01	1.18779E-01	7.31741E-02	7.39104E-02	7.95987E-02	6.85965E-02	1.07909E-01	1.52634E-01
+ 3	1.06680E-01	1.39556E-01	5.49774E-02	2.00378E-02	1.25166E-02	1.32261E-02	1.23322E-02	2.02416E-02	3.82178E-02	4.37863E-02
+ 4	6.46550E-02	3.86576E-02	9.48767E-03	3.33964E-03	2.35355E-03	2.12183E-03	3.59935E-03	6.85965E-03	1.07909E-02	1.30236E-02
+ 5	2.12918E-02	6.59086E-03	2.18946E-03	9.10810E-04	5.34899E-04	8.48732E-04	1.77017E-03	2.92379E-03	5.39546E-03	8.75725E-03

from grp / to grp	grp 11	grp 12	grp 13	grp 14	grp 15	grp 16
+ 0	3.02946E-01	3.19203E-01	2.89136E-01	3.46347E-01	8.10619E-01	1.00000E+00
+ 1	4.94030E-01	4.59900E-01	4.33884E-01	4.94691E-01	1.89381E-01	
+ 2	1.35724E-01	1.32774E-01	2.07145E-01	1.58962E-01		
+ 3	4.03578E-02	6.63872E-02	6.98356E-02			
+ 4	2.02070E-02	2.17359E-02				
+ 5	6.73566E-03					

Figure F17.6.12 Example of 2-D macroscopic cross sections

```
 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    1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00
        transfers from grp+   1 to grp+   4 are the same as above
  +  5    1.00000E+00 1.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00
     from   grp  11     grp  12     grp  13     grp  14     grp  15     grp  16
to grp
  +  0    1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00
  +  1    1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00
  +  2    1.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00
  +  3    1.00000E+00 1.00000E+00 1.00000E+00
  +  4    1.00000E+00 1.00000E+00
  +  5    1.00000E+00
```

Figure F17.6.13 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    2.50717E-02 1.37955E-01 1.34616E-02 7.28156E-02 2.88056E-01 4.28189E-01 4.40454E-01 4.42506E-01 4.43150E-01 3.70214E-01
  +  1    5.38829E-01 6.16596E-01 5.69298E-01 5.77713E-01 5.52605E-01 5.45150E-01 5.49196E-01 5.50241E-01 5.78041E-01 6.17541E-01
  +  2    5.20566E-01 5.65012E-01 4.67921E-01 3.16948E-01 2.26608E-01 2.37321E-01 2.40919E-01 2.54098E-01 3.31771E-01 3.63001E-01
  +  3    3.82358E-01 3.49483E-01 2.25674E-01 1.21212E-01 1.02564E-01 1.01604E-01 1.10048E-01 1.41667E-01 1.88235E-01 2.02564E-01
  +  4    2.36583E-01 1.70493E-01 9.61538E-02 4.54545E-02 4.54545E-02 3.33333E-02 6.55738E-02 8.19672E-02 1.04167E-01 1.20690E-01
  +  5    1.19735E-01 5.00000E-02-2.00000E+00-2.00000E+00-2.00000E+00-2.00000E+00 3.33333E-02 3.84615E-02 6.25000E-02 7.05128E-02
     from   grp  11     grp  12     grp  13     grp  14     grp  15     grp  16
to grp
  +  0    3.37214E-01 3.36073E-01 3.09208E-01 5.56326E-01 4.45358E-01 2.52418E-01
  +  1    6.18647E-01 6.19905E-01 6.40449E-01 6.23199E-01 3.81259E-01
  +  2    3.56907E-01 3.79019E-01 4.00109E-01 1.81818E-01
  +  3    2.11405E-01 2.26734E-01 1.28824E-01
  +  4    1.25000E-01 7.23514E-02
  +  5    4.16667E-02
```

Figure F17.6.14 Example of macroscopic angles

F17.6.12 TABLE OF ADDITIONAL INFORMATION

This table contains information determined by the input data and is printed by subroutine NSUPG. An example of this table is shown in Fig. F17.6.15.

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. F17.6.5, 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 1. 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 1 and may be overridden by specifying the parameter SCT= in the mixing table input as described in Sect. F17.4.10. One scattering angle yields P_1 cross sections, two scattering angles yield P_3 cross sections, three scattering angles yield 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 value 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 IDS USED is the number of different biasing regions used in the problem. This will always be 1 unless a biasing data block is entered in the problem as described in Sect. F17.4.7.

The NUMBER OF DIFFERENTIAL ALBEDOS USED is the number of different differential albedo reflectors used in the problem. This value 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. F17.4.6. 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 F17.4.4 for a list of differential albedo keywords.

The TOTAL INPUT GEOMETRY REGIONS is the number of geometry regions specified in the problem input. This is defined by the MEDIA-type record used in describing the data. The MEDIA records start with the words MEDIA, ARRAY, and HOLE. 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 LARGEST GEOMETRY UNIT NUMBER is the largest unit number used in the problem.) The LARGEST ARRAY NUMBER is the largest array number specified in the array data. (See Sect. F17.4.5.)

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 F17.4.5 describes array data. The GLOBAL ARRAY NUMBER is the number of the array designated as the global array. Any array in a problem can be designated the global array. The global array is used only when matrix data are being collected.

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 that are located along the edge of the array boundaries in the x/y/z directions. For a single unit, all three of these values should be zero. For a simple $1 \times 1 \times 1$ array consisting of one unit type, all three of these numbers should be 1.

```
*****************************************************************************************************
***                                                                                               ***
***          sample problem 18   1f27  critical experiment                                        ***
***                                                                                               ***
*****************************************************************************************************
*****************************************************************************************************
***                                                                                               ***
***                                ****** additional information ******                           ***
***                                                                                               ***
***  number of energy groups                16       use lattice geometry                   yes  ***
***                                                                                               ***
***  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    28       number of units in the global y dir.     0  ***
***                                                                                               ***
***  entries/neutron in the fission bank    15       number of units in the global z dir.     0  ***
***                                                                                               ***
***  number of mixtures used                 3       use a global reflector                 yes  ***
***                                                                                               ***
***  number of bias id's used                6       use nested holes                        no  ***
***                                                                                               ***
***  number of differential albedos used     1       number of holes                          1  ***
***                                                                                               ***
***  total input geometry regions           15       maximum hole nesting level               1  ***
***                                                                                               ***
***  number of geometry regions used        15       use nested arrays                       no  ***
***                                                                                               ***
***  largest geometry unit number            3       number of arrays used                    4  ***
***                                                                                               ***
***  largest array number                    4       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                  h2o   ***
***                                                                                               ***
*****************************************************************************************************
```

Figure F17.6.15 Sample table of additional information

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. F17.4.4).

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. F17.4.5) actually used in the problem description.

MAXIMUM ARRAY NESTING LEVEL is the deepest level of array nesting.

Six BOUNDARY CONDITIONs are 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. F17.4.6. Also refer to the NUMBER OF DIFFERENTIAL ALBEDOS USED, discussed previously in the description of this table of information.

F17.6.13 TABLE OF SPACE AND SUPERGROUP INFORMATION

Another table printed by KENO-VI, as shown in Fig. F17.5.15, summarizes the space requirements for the problem and prints information related to the supergroups. This table is printed by subroutines NSUPG and LIMLN.

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. Following this information are statements concerning the amount of storage available for supergrouped data, the amount of storage available for constructing the supergroups, 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, and the number of words of storage that will be used to run this problem.

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.

F17.6.14 ARRAY SUMMARY

The arrays that are used in the problem are summarized in the table shown in Fig. F17.6.17. This table is printed by subroutine PRTARA whenever more than one array is used in the problem.

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 global array is flagged by the word GLOBAL. The NESTING LEVEL indicates the level of nesting for each array. Arrays can be nested to any level, limited only by the capability of the computer. A nesting level of 1 is the highest or first nesting level. The larger the number in the nesting level column, the deeper the nesting level.

```
*********************************************************************************************
***                                                                                       ***
***          sample problem 18   1f27  critical experiment                              ***
***                                                                                       ***
*********************************************************************************************
*********************************************************************************************
***                                                                                       ***
***                    ****** space and supergroup information ******                     ***
***                                                                                       ***
***        30000 words is the total space available.                                      ***
***                                                                                       ***
***        18899 words were used for non-supergroup storage.                              ***
***                                                                                       ***
***        11101 words of storage are available for supergrouped data.                    ***
***                                                                                       ***
***        29626 words of storage are available for constructing the supergroups.         ***
***                                                                                       ***
***        11062 words of storage are available to each supergroup.                       ***
***                                                                                       ***
***          690 words are needed for the largest group.                                  ***
***                                                                                       ***
***        19733 words of storage is sufficient to run this problem.                      ***
***                                                                                       ***
***        26768 words of storage will allow the problem to run with one supergroup.      ***
***                                                                                       ***
***        26928 words of storage will be used to run this problem.                       ***
***                                                                                       ***
*********************************************************************************************
*********************************************************************************************
***                                                                                       ***
***                    starting      ending       xsec       albedo      total          ***
***   supergroup         group        group       length      length      length         ***
***                                                                                       ***
***                                                                                       ***
***          1            1            16          243         544         7830          ***
***                                                                                       ***
*********************************************************************************************
                  ........    0 io's were used in supergrouping    ........
```

Figure F17.6.16 Sample table of space and supergroup information

```
************************************************************
**                                                        **
**   array      units in    units in    units in   nesting **
**   number      x dir.      y dir.      z dir.     level  **
**                                                        **
**     1           2           2           2         1    **
**                                                        **
**     2           2           2           1         1    **
**                                                        **
**     3           1           2           3         1    **
**                                                        **
**     4           3           1           3         1    **
**                                                        **
************************************************************

 ........    0 io's were used loading the data    ........
```

Figure F17.6.17 Examples of array summary

F17.6.15 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. F17.6.18.

The problem title and a heading are printed at the top of each page. The unit number followed by the GEOMETRY data for that unit is then printed. Each geometry record type used in the unit, numbered in the order they appear in the unit, is printed out. Following each geometry record type is the set of quadratic equations that describe the input geometry for that geometry record. The MEDIA data, consisting of four columns, is then listed in the order it appears in the problem. The first column contains the media-type record. The second column contains the media/hole/array number. The third column contains the bias ID number if the media-type record is MEDIA and is otherwise blank. The fourth column contains the sector definition array, which describes the region location relative to the GEOMETRY records listed above for that unit.

The data consisting of the unit number, GEOMETRY data, and MEDIA data are repeated for each unit used in the problem. The global unit has a heading printed before the unit number to identify it as the global unit.

F17.6.16 UNIT ORIENTATION DESCRIPTION

Each unit orientation description defines the location of units in the 3-D 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 ensure the proper placement of the units in each lattice. A sample unit orientation description is shown in Fig. F17.6.19.

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 that indicate the portion of each lattice being printed. The unit orientation descriptions are printed in subroutine PRTLBA.

F17.6.17 VOLUME INFORMATION

Three tables of volumes are printed by subroutine VOLUME and cannot be suppressed. Currently, the volume data contained in these tables are meaningless. Future work on KENO-VI includes calculating the volumes of each region as is done in KENO-V.a. The first table printed will include the volume of each individual region and the cumulative volumes for each region in the unit. The second table will list the number of times each unit is used in the problem and the total volume of each region throughout the entire problem description. The third table will list the total volume of each mixture used in the problem. An example of the volume printout is given in Fig. F17.6.20.

```
                    sample problem 18   1f27  critical experiment
                geometry description for those units utilized in this problem

                                                    -----   unit     1   -----
  1           cylinder                                          quadratic surfaces
   -1.000x**2     -1.000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z     90.630
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z     77.095
  2           cylinder                                          quadratic surfaces
   -1.000x**2     -1.000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z     90.630
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .209z     78.932
  3           cylinder                                          quadratic surfaces
   -1.000x**2     -1.000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z    103.226
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .209z     90.715
  4 boundary  cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z    340.402
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z    340.402
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .209z    316.473
                        sector
                imp       definitions
media     1      1          1
media     0      1         -1   2
media     2      1         -1  -2   3
media     0      1          4  -2  -3

  1 boundary  cuboid                                    -----   unit     2   -----
                                                               quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz    -36.900x        .000y        .000z   1020.470
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x     36.900y        .000z   1020.470
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y     35.580z    948.741
  2           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz    -36.900x        .000y        .000z   1020.470
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x     36.900y        .000z   1020.470
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y    -71.160z   -949.808
  3           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz     73.800x        .000y        .000z  ********
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x     36.900y        .000z   1020.470
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   2847.290
  4           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3062.516
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x    -73.800y         .000z  ********
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   2847.290
  5           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3063.622
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3063.622
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   2848.357
                        sector
                imp       definitions
array     1                  1
array     2                  2  -1
array     3                  3  -2  -1
array     4                  4  -3  -2  -1
media     0      1           5  -4  -3  -2  -1

                                  ********************   global   ********************
                                                    -----   unit     3   -----
  1           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3063.622
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3063.622
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   2848.357
  2           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3404.723
1.000
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3404.723
1.000
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3177.577
1.000  3           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3763.823
1.000
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3763.823
1.000
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3524.797
1.000
  4           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4140.922
1.000
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4140.922
1.000
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   3890.017
1.000
  5           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4536.022
1.000
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4536.022
1.000
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4273.237
1.000
  6           cuboid                                           quadratic surfaces
   -1.000x**2        .000y**2       .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4982.948
1.000
     .000x**2     -1.000y**2        .000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4982.948
1.000
     .000x**2        .000y**2    -1.000z**2       .000xy       .000xz       .000yz       .000x        .000y        .000z   4707.332
1.000
                        sector
                imp       definitions
hole      1                  1
media     3      2           -1   2
media     3      3           -2   3
media     3      4           -3   4
media     3      5           -4   5
media     3      6            6  -5
```

Figure F17.6.18 Example of geometry region data

```
                    sample problem 18   1f27  critical experiment
                -------  unit orientation description for array   1        -------
 z layer   1, x column    1 to    2 left to right    y row    1 to    2  bottom to top
  1 1
  1 1
-z layer   2, x column    1 to    2 left to right    y row    1 to    2  bottom to top
  1 1
  1 1
```

Figure F17.6.19 Example of unit orientation description

sample problem 18 1f27 critical experiment
volumes for those units utilized in this problem

unit	region	geometry region	volume	cumulative volume
1	1	1	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	2	1.00000E+00 cm**3	1.00000E+00 cm**3
	3	3	1.00000E+00 cm**3	1.00000E+00 cm**3
	4	4	1.00000E+00 cm**3	1.00000E+00 cm**3
2	1	5	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	6	1.00000E+00 cm**3	1.00000E+00 cm**3
	3	7	1.00000E+00 cm**3	1.00000E+00 cm**3
	4	8	1.00000E+00 cm**3	1.00000E+00 cm**3
	5	9	1.00000E+00 cm**3	1.00000E+00 cm**3
3	1	10	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	11	1.00000E+00 cm**3	1.00000E+00 cm**3
	3	12	1.00000E+00 cm**3	1.00000E+00 cm**3
	4	13	1.00000E+00 cm**3	1.00000E+00 cm**3
	5	14	1.00000E+00 cm**3	1.00000E+00 cm**3
	6	15	1.00000E+00 cm**3	1.00000E+00 cm**3

unit	uses	region	mixture	total volume
1	27	1	1	2.70000E+01 cm**3
		2	0	2.70000E+01 cm**3
		3	2	2.70000E+01 cm**3
		4	0	2.70000E+01 cm**3
2	1	1		1.00000E+00 cm**3
		2		1.00000E+00 cm**3
		3		1.00000E+00 cm**3
		4		1.00000E+00 cm**3
		5	0	1.00000E+00 cm**3
3	1	1	-1	1.00000E+00 cm**3
		2	3	1.00000E+00 cm**3
		3	3	1.00000E+00 cm**3
		4	3	1.00000E+00 cm**3
		5	3	1.00000E+00 cm**3
		6	3	1.00000E+00 cm**3

total mixture volumes

mixture	total volume
0	5.50000E+01 cm**3
1	2.70000E+01 cm**3
2	2.70000E+01 cm**3
3	5.00000E+00 cm**3

Figure F17.6.20 Sample volume information

The unit number is printed under the heading UNIT. The REGION refers to the number of the geometry region within the unit determined by the order of the MEDIA records in the unit. The geometry regions within a unit are numbered sequentially, starting with 1. GEOMETRY REGION refers to the entry number for the individual geometry region. These regions are numbered sequentially, starting with 1, through the TOTAL INPUT GEOMETRY REGIONS defined in Sect. F17.6.12. The VOLUME is the net volume of the individual region. The CUMULATIVE VOLUME is calculated from the dimensions of the region. In the second table, USES indicates the number of times the unit is used in the problem. MIXTURE is the mixture number used in the region. A minus sign indicates an array or unit, placed in a hole, in that region. TOTAL VOLUME is the total volume of that region, determined by multiplying the VOLUME of the region by the number of times the unit containing that region is used in the problem. In the third table, the mixtures used in the problem are listed along with their total volumes. 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. 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.

F17.6.18 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. F17.6.21.

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. F17.6.19.

```
*****************************************************************************************************
***                                                                                               ***
***                                        biasing information                                    ***
***                                                                                               ***
***    weighting intervals 1 to 5 for paraffi ,mat id=400 will be used for bias id's 2 to 6       ***
***                                                                                               ***
***    a default weight of .500 will be used for all other bias id's.                             ***
***                                                                                               ***
*****************************************************************************************************
```

Figure F17.6.21 Biasing information

F17.6.19 GROUP-DEPENDENT WEIGHTS

Printing the group-dependent weights is optional. They are printed by subroutine PRTWTS if PWT=YES (Sect. F17.4.3) is entered in the parameter data. An example of the printed group-dependent weights is shown in Fig. F17.6.22.

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.

sample problem 18 1f27 critical experiment

group dependent weights

energy group	bias id 1	bias id 2	bias id 3	bias id 4	bias id 5	bias id 6
1	5.00000E-01	5.96000E-01	8.46000E-01	1.28000E+00	2.08000E+00	3.51000E+00
2	5.00000E-01	5.89000E-01	8.73000E-01	1.50000E+00	2.87000E+00	5.88000E+00
3	5.00000E-01	5.83000E-01	9.15000E-01	1.80000E+00	4.17000E+00	1.07000E+01
4	5.00000E-01	5.77000E-01	9.72000E-01	2.17000E+00	5.80000E+00	1.73000E+01
5	5.00000E-01	5.94000E-01	1.14000E+00	2.99000E+00	9.42000E+00	3.28000E+01
6	5.00000E-01	6.40000E-01	1.44000E+00	4.46000E+00	1.61000E+01	6.19000E+01
7	5.00000E-01	6.71000E-01	1.63000E+00	5.38000E+00	2.02000E+01	7.86000E+01
8	5.00000E-01	6.84000E-01	1.75000E+00	6.02000E+00	2.30000E+01	9.01000E+01
9	5.00000E-01	6.87000E-01	1.84000E+00	6.57000E+00	2.54000E+01	9.97000E+01
10	5.00000E-01	6.73000E-01	1.88000E+00	6.95000E+00	2.71000E+01	1.07000E+02
11	5.00000E-01	6.48000E-01	1.89000E+00	7.12000E+00	2.79000E+01	1.10000E+02
12	5.00000E-01	6.31000E-01	1.95000E+00	7.48000E+00	2.94000E+01	1.16000E+02
13	5.00000E-01	7.68000E-01	2.65000E+00	1.03000E+01	4.05000E+01	1.59000E+02
14	5.00000E-01	8.18000E-01	2.97000E+00	1.16000E+01	4.56000E+01	1.80000E+02
15	5.00000E-01	8.24000E-01	3.05000E+00	1.19000E+01	4.70000E+01	1.85000E+02
16	5.00000E-01	8.91000E-01	3.52000E+00	1.39000E+01	5.46000E+01	2.15000E+02

1f27 xy plot at z=0.0

Figure F17.6.22 Example of biasing data

F17.6.20 PLOT REPRESENTATION

Plots representing 2-D slices through the geometrical description of the problem are optional. They are created if plot data are entered as specified in Sect. F17.4.11 unless PLT=NO is specified either in the plot data or the parameter data (Sect. F17.4.3). A plot or plots can be generated and displayed as (1) alphanumeric characters to represent mixture numbers, unit numbers, or bias ID numbers or (2) color plots displayed to the screen or plotting device using colors to represent mixture numbers, unit numbers, or bias ID numbers. Color plots generate a GIF file and require an independent program to be displayed using a plotting device.

An example of the output generated using the character plot method is given in Figs. F17.6.23 and F17.6.24. An example of the output generated using the color plot method is given in Figs. F17.6.25 and F17.6.26.

Figure F17.6.23 summarizes the data used to generate the character plot. Figure F17.6.24 is an example of a character plot of the 2-D slice specified through the geometrical description of the problem.

In Figure F17.6.23, the plot title is printed at the top of the page, followed by a statement that "THE FOLLOWING WILL BE A CHARACTER PLOT." 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 plot (MIXTURE MAP, BIAS ID MAP, or UNIT MAP). This is followed by a table that correlates the symbols to be used in the character plot with the mixture numbers, bias ID numbers, or unit numbers that were used in the problem. If the problem is a bare array, the overall system coordinates are printed. Then the coordinates of the upper-left corner and lower-right corner of the plot are printed. This is followed by the direction cosines down and across the plot. The remaining plot parameters (including both input data and calculated values) are then printed. NU is the number of characters printed in the "U" (down) direction, NV is the number of characters printed in the "V" (across) direction, DELU is the incremental distance, in cm, represented by each character in the "U" (down) direction, DELV is the incremental distance, in cm, represented by each character in the "V" (across) direction, and LPI is the vertical to horizontal scaling factor for plot proportionality.

```
 1f27 xy plot at z=0.0

                    the following will be a character plot

                                        mixture map

mixture  0 1 2 3 4
 symbol  . * - 3

      upper  left          lower right
      coordinates          coordinates

x     -7.1000e+01           7.1000e+01
y      7.1000e+01          -7.1000e+01
z      0.0000e+00           0.0000e+00

          u axis      v axis
          (down)      (across)

x          .00000     1.00000
y        -1.00000      .00000
z          .00000      .00000

nu=   78   nv=  130     delu= 1.8205e+00     delv= 1.0923e+00     lpi=   6.000
```

Figure F17.6.23 Summary of character plot symbols, coordinates, and data

Figure F17.6.24 shows a character plot of a 2-D 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 in one problem.

Figure F17.6.25 summarizes the data used to generate the color plot. Figure F17.6.26 is an example of a color plot of the 2-D slice specified through the geometrical description of the problem. This plot does not appear in the KENO V.a printout. It is generated from a GIF file that is created when a color plot has been specified in the KENO V.a input data and requires special processing by the user.

In Figure F17.6.25, the plot title is printed at the top of the page, followed by a statement that "THE FOLLOWING WILL BE A COLOR PLOT." 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 plot (MIXTURE MAP, BIAD ID MAP, or UNIT MAP). If the problem is a bare array, the overall system coordinates are printed. Then the coordinates of the upper-left corner and lower-right corner of the plot are printed. This is followed by the direction cosines down and across the plot. The remaining plot parameters (including both input data and calculated values) are then printed. NU is the number of characters printed in the "U" (down) direction, NV is the number of characters printed in the "V" (across) direction, DELU is the incremental distance, in cm, represented by each character in the "U" (down) direction, DELV is the incremental distance, in cm, represented by each character in the "V" (across) direction, and LPI is the vertical to horizontal scaling factor for plot proportionality.

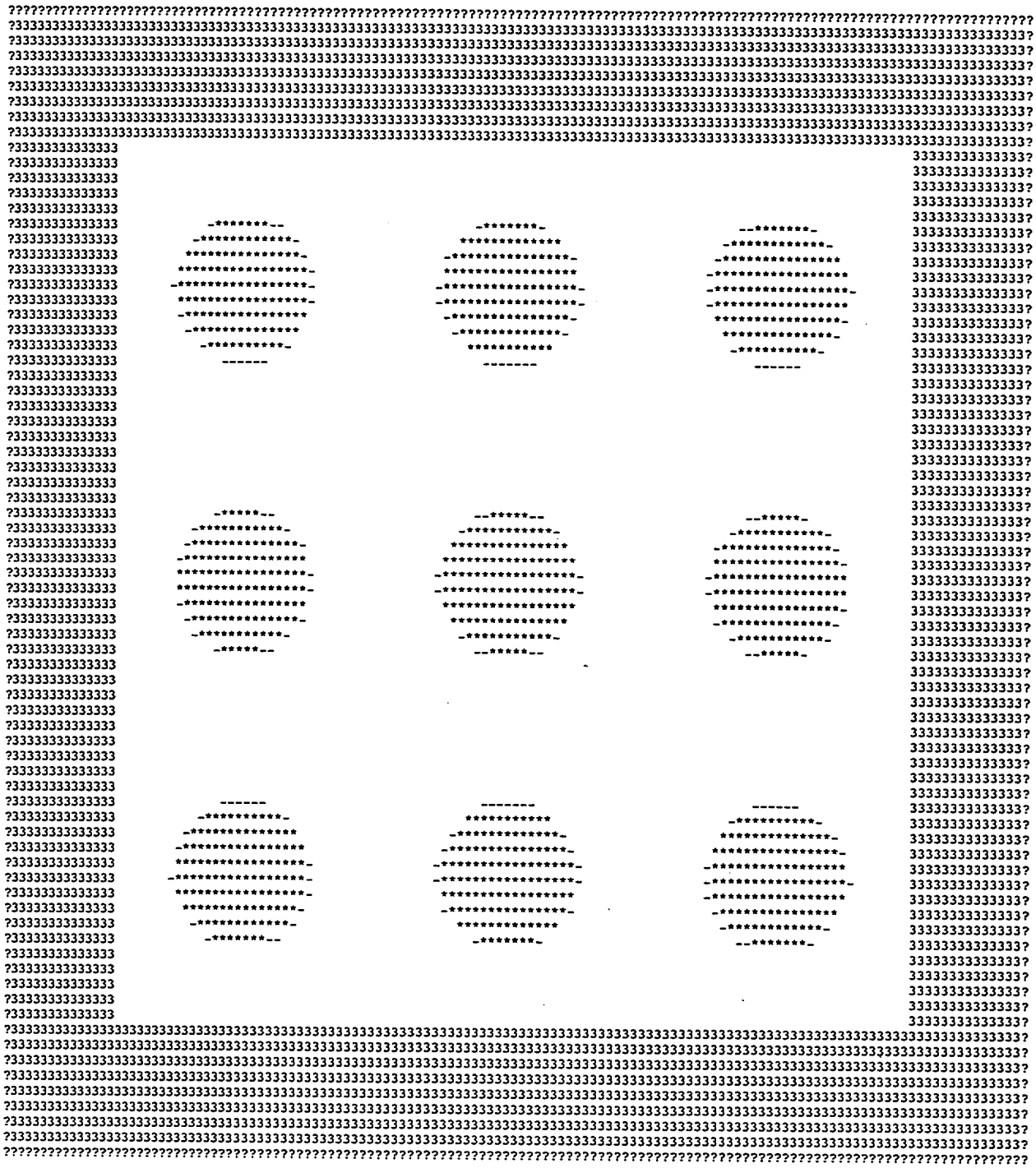


Figure F17.6.24 Sample character plot representation

```
 1f27 xy plot at z=0.0

                  the following will be a color plot

                                        mixture map

mixture  0 1 2 3 4
 symbol    1 2 3 4

      upper  left          lower right
      coordinates          coordinates

x     -7.1000e+01           7.1000e+01
y      7.1000e+01          -7.1000e+01
z      0.0000e+00           0.0000e+00

          u axis      v axis
          (down)      (across)

x          .00000      1.00000
y        -1.00000       .00000
z          .00000       .00000

nu= 640   nv= 640     delu= 2.2187e-01     delv= 2.2187e-01     lpi=  10.000
```

Figure F11.6.25 Summary of plot symbols, coordinates, and data

Figure F17.6.26 shows an example of a color plot of a 2-D slice specified through the geometrical description of the problem. The color plots do not appear in the KENO V.a printout. They are generated from GIF files that are created when a color plot has been specified in the KENO V.a input data. Color plots require special processing by the user. Any number of plots can be made in one problem. The color plots can be a valuable tool to assist the user in verifying that a problem is described correctly.

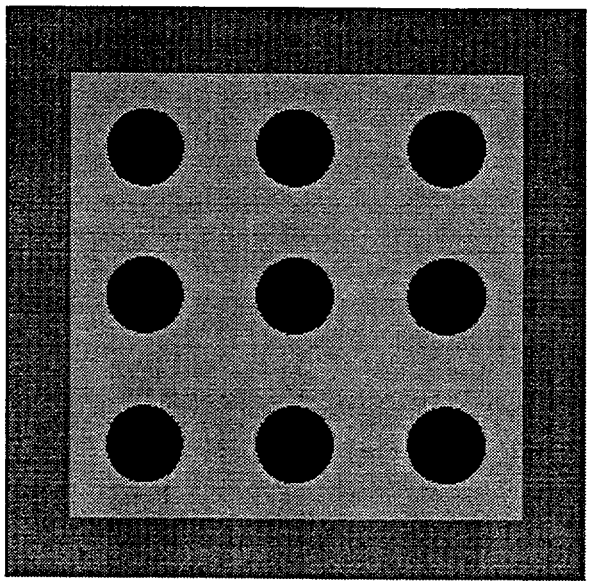


Figure F17.6.26 Sample color plot representation

F17.6.21 CONCLUDE PRETRACKING EDITS

Prior to calculating the k-effectives for each generation, KENO-VI prints a final edit as shown in Fig. F17.6.27

```
                                ........    0 io's were used in keno-v before tracking        ........
                                ........   .06400minutes were used processing data.         ........
volume fraction of fissile material in the system=  .23684E+00
start type 0 was used.
the neutrons were started with a flat distribution in a cuboid defined by:
 +x= 7.05900E+01  -x=-7.05900E+01  +y= 7.05900E+01  -y=-7.05900E+01  +z= 6.86100E+01  -z=-6.86100E+01
 .02133 minutes were required for starting.   total elapsed time is  .08533 minutes.
```

Figure F17.6.27 Pretracking edits

The information printed in this edit includes the number of I/Os 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 system. The volume fraction of fissile material in the core is currently meaningless. Future plans for KENO-VI include the ability to calculate volumes and volume fractions. If RFL=NO (Sect. F17.4.8) was specified in the start data, the volume fraction message is changed to: VOLUME FRACTION OF FISSILE MATERIAL IN THE CORE =.

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. Start type 0, which generates a flat, random distribution over all fissile material in the problem, is the default source distribution.

F17.6.22 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. F17.6.28.

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.

WT is the current weight (WT is always 1.0 for a neutron when it is started).

K is the region number that contains the point X, Y, Z.

NBX, NBY, NBZ are the coordinates of the unit within the array.

where	neutron random number	x u	y v	z w	wt etausd	k now	nbx kcor	nby llc	nbz kcol	lstc lsth	nhol ig	ll nsig	kr sg	ki
start	350	-2.93389E+00	4.55407E+00	4.68300E+00	1.00000E+00	1	2	1	3	0	1	1	1	0
	0000334768P44690	3.78162E-01	4.60779E-01	8.02917E-01	0.00000E+00	4	8	0	0	3	3	1		
start	349	-4.00210E+00	6.15422E+00	5.93966E+00	1.00000E+00	1	3	1	2	0	1	1	1	0
	0000334768P44690	2.08141E-01	2.01154E-01	-9.57191E-01	0.00000E+00	4	8	0	0	1	1	1		
start	348	-2.80915E+00	2.29649E+00	3.72475E+00	1.00000E+00	1	2	1	1	0	1	1	1	0
	0000334768P44690	-7.35305E-01	1.49356E-01	6.61074E-01	0.00000E+00	4	6	0	0	1	1	1		
start	347	3.94031E+00	-6.48469E+00	1.03994E+00	1.00000E+00	1	1	1	1	0	1	1	1	0
	0000334768P44690	3.18453E-01	-2.49655E-01	-9.14473E-01	0.00000E+00	4	6	0	0	4	4	1		
start	346	5.68195E+00	-4.41095E-01	-3.78734E+00	1.00000E+00	1	2	2	2	0	1	1	1	0
	0000334768P44690	3.83747E-01	-8.02679E-01	4.56557E-01	0.00000E+00	4	5	0	0	4	4	1		
start	345	-1.29827E+00	-1.33806E+00	-5.75073E+00	1.00000E+00	1	1	1	2	0	1	1	1	0
	0000334768P44690	-3.05448E-01	-9.31668E-01	-1.96713E-01	0.00000E+00	4	7	0	0	2	2	1		
start	344	3.16313E-02	8.62766E+00	3.12925E+00	1.00000E+00	1	2	1	1	0	1	1	1	0
	0000334768P44690	-8.19429E-01	2.34186E-01	5.23157E-01	0.00000E+00	4	8	0	0	4	4	1		
start	343	2.31603E+00	-3.41755E+00	8.47753E+00	1.00000E+00	1	3	1	1	0	1	1	1	0
	0000334768P44690	2.50993E-01	-9.44899E-01	-2.10163E-01	0.00000E+00	4	8	0	0	2	2	1		
start	342	7.92013E+00	-3.31669E+00	-6.54100E+00	1.00000E+00	1	1	1	1	0	1	1	1	0
	0000334768P44690	1.75156E-02	9.23284E-01	3.83718E-01	0.00000E+00	4	8	0	0	3	3	1		
start	341	8.17152E+00	7.80831E-01	4.88753E+00	1.00000E+00	1	2	1	2	0	1	1	1	0
	0000334768P44690	-4.58036E-01	8.74804E-01	-1.57864E-01	0.00000E+00	4	8	0	0	1	1	1		
start	340	2.25991E+00	-7.97285E+00	1.27218E+00	1.00000E+00	1	1	2	1	0	1	1	1	0
	0000334768P44690	4.04771E-01	4.06303E-01	8.19193E-01	0.00000E+00	4	5	0	0	1	1	1		
start	339	-7.11336E+00	-5.01182E+00	-1.35331E-01	1.00000E+00	1	1	2	2	0	1	1	1	0
	0000334768P44690	-8.35758E-01	4.92574E-01	-2.42650E-01	0.00000E+00	4	5	0	0	2	2	1		
start	338	-4.42523E+00	1.73051E+00	4.17569E+00	1.00000E+00	1	1	2	1	0	1	1	1	0
	0000334768P44690	2.08661E-01	-9.69024E-01	1.32109E-01	0.00000E+00	4	7	0	0	2	2	1		

Figure F17.6.28 Example of initial source points

LSTC is the last unit where the neutron resided

NHOL is the hole where the neutron currently resides.

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.

RANDOM NUMBER is the current random number.

U, V, and W are the direction cosines defining the direction the history is traveling.

ETAUSD is the amount of the distance to collision already traversed.

NOW is the array where the neutron currently resides.

KCOR is the core boundary region of the array where the neutron currently resides.

LLC is the unit surrounding the array the unit is in.

KCOL is the region number in which the last collision occurred.

IG is the energy group.

NSIG is the position of the group IG in the supergroup.

SG is the supergroup 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.

F17.6.23 K-EFFECTIVES BY GENERATION

At the completion of each generation, KENO-VI prints the k-effective for that generation and associated information. An example of this printout is given in Fig. F17.6.29.

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 *nskip-1* generations), (5) the deviation associated with the average k-effective, (6) the matrix k-effective for the generation, and (7) 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.

After the last generation, a message is printed to indicate why execution was terminated. If matrix k-effectives were calculated, this information is followed by a message stating the method used to determine the matrix k-effectives printed under the heading "MATRIX K-EFFECTIVE," as shown in Fig. F17.6.29.

sample problem 18 1f27 critical experiment

generation	generation k-effective	elapsed time minutes	average k-effective	avg k-eff deviation	matrix k-effective	matrix k-eff deviation
1	9.55981E-01	1.92000E-01	1.00000E+00	0.00000E+00	9.55979E-01	0.00000E+00
2	9.85743E-01	3.20000E-01	1.00000E+00	0.00000E+00	9.70861E-01	1.48842E-02
3	9.81019E-01	4.26667E-01	9.81019E-01	0.00000E+00	9.74246E-01	9.23697E-03
4	9.75411E-01	5.33333E-01	9.78215E-01	2.80419E-03	9.74537E-01	6.53871E-03
5	1.02908E+00	6.61333E-01	9.95171E-01	1.70332E-02	9.85446E-01	1.20268E-02
6	1.00068E+00	7.89333E-01	9.96547E-01	1.21227E-02	9.87984E-01	1.01431E-02
7	1.01941E+00	8.96000E-01	1.00112E+00	1.04444E-02	9.92473E-01	9.67729E-03
8	1.06116E+00	1.00267E+00	1.01113E+00	1.31479E-02	1.00106E+00	1.19984E-02
9	1.09419E+00	1.10933E+00	1.02299E+00	1.62569E-02	1.01141E+00	1.48002E-02
10	9.53752E-01	1.23733E+00	1.01434E+00	1.65265E-02	1.00564E+00	1.44391E-02
11	9.85589E-01	1.36533E+00	1.01114E+00	1.49210E-02	1.00382E+00	1.31874E-02
12	9.68067E-01	1.49333E+00	1.00684E+00	1.40237E-02	1.00084E+00	1.24016E-02
13	9.81856E-01	1.60000E+00	1.00457E+00	1.28866E-02	9.99378E-01	1.15009E-02
14	1.02367E+00	1.72800E+00	1.00616E+00	1.18711E-02	1.00111E+00	1.07885E-02
15	1.04387E+00	1.85600E+00	1.00906E+00	1.12986E-02	1.00396E+00	1.04404E-02
16	9.42642E-01	1.98400E+00	1.00431E+00	1.14860E-02	1.00013E+00	1.04915E-02
17	9.97040E-01	2.09067E+00	1.00383E+00	1.07038E-02	9.99949E-01	9.85665E-03
18	1.07760E+00	2.19733E+00	1.00844E+00	1.10231E-02	1.00426E+00	1.02454E-02
19	9.80882E-01	2.30400E+00	1.00682E+00	1.04805E-02	1.00303E+00	9.76909E-03
20	9.02059E-01	2.43200E+00	1.00100E+00	1.14678E-02	9.97983E-01	1.05537E-02
21	9.91126E-01	2.53867E+00	1.00048E+00	1.08599E-02	9.97657E-01	1.00440E-02
22	9.39241E-01	2.64533E+00	9.97418E-01	1.07479E-02	9.95001E-01	9.93809E-03
23	9.94304E-01	2.77333E+00	9.97269E-01	1.02244E-02	9.94971E-01	9.49623E-03
24	1.00598E+00	2.90133E+00	9.97665E-01	9.75662E-03	9.95429E-01	9.10327E-03
25	9.81224E-01	2.98667E+00	9.96950E-01	9.35014E-03	9.94861E-01	8.75014E-03
26	9.91740E-01	3.11467E+00	9.96733E-01	8.95471E-03	9.94741E-01	8.40740E-03
27	9.90335E-01	3.22133E+00	9.96477E-01	8.59287E-03	9.94578E-01	8.09182E-03
28	9.46905E-01	3.32800E+00	9.94571E-01	8.47306E-03	9.92875E-01	7.98133E-03
29	9.27620E-01	3.45600E+00	9.92091E-01	8.52194E-03	9.90625E-01	8.02325E-03
30	9.06063E-01	3.56267E+00	9.89019E-01	8.76789E-03	9.87806E-01	8.24758E-03
31	1.00100E+00	3.69067E+00	9.89432E-01	8.47023E-03	9.88232E-01	7.98861E-03
32	9.72119E-01	3.79733E+00	9.88855E-01	8.20334E-03	9.87728E-01	7.75137E-03
33	9.65558E-01	3.92533E+00	9.88103E-01	7.96981E-03	9.87056E-01	7.54264E-03
34	9.48828E-01	4.03200E+00	9.86876E-01	7.81373E-03	9.85932E-01	7.40311E-03
35	1.00807E+00	4.16000E+00	9.87518E-01	7.60043E-03	9.86564E-01	7.21605E-03
36	9.70213E-01	4.26667E+00	9.87009E-01	7.39105E-03	9.86110E-01	7.02736E-03
37	1.01056E+00	4.41600E+00	9.87682E-01	7.20825E-03	9.86771E-01	6.86649E-03
38	9.90428E-01	4.54400E+00	9.87758E-01	7.00558E-03	9.86867E-01	6.68391E-03
39	1.05722E+00	4.65067E+00	9.89635E-01	7.06747E-03	9.88671E-01	6.75549E-03
40	1.01137E+00	4.75733E+00	9.90207E-01	6.90272E-03	9.89238E-01	6.60886E-03
41	9.93761E-01	4.88533E+00	9.90299E-01	6.72401E-03	9.89349E-01	6.44657E-03
42	1.12885E+00	4.99200E+00	9.93762E-01	7.41284E-03	9.92670E-01	7.11412E-03
43	9.71069E-01	5.09867E+00	9.93209E-01	7.25093E-03	9.92168E-01	6.96495E-03
44	9.99904E-01	5.22667E+00	9.93368E-01	7.07798E-03	9.92343E-01	6.80706E-03
45	1.08103E+00	5.35467E+00	9.95407E-01	7.20582E-03	9.94314E-01	6.93963E-03
46	1.00679E+00	5.50400E+00	9.95666E-01	7.04490E-03	9.94585E-01	6.79251E-03
47	1.00537E+00	5.61067E+00	9.95881E-01	6.88994E-03	9.94815E-01	6.65035E-03
48	1.02227E+00	5.73867E+00	9.96455E-01	6.76288E-03	9.95387E-01	6.53554E-03
49	1.07336E+00	5.86667E+00	9.98091E-01	6.81669E-03	9.96978E-01	6.59562E-03
50	9.79966E-01	5.97333E+00	9.97714E-01	6.68384E-03	9.96638E-01	6.47120E-03
51	1.03354E+00	6.08000E+00	9.98445E-01	6.58671E-03	9.97361E-01	6.38403E-03
52	9.79903E-01	6.20800E+00	9.98074E-01	6.46428E-03	9.97026E-01	6.26887E-03
53	1.02787E+00	6.33600E+00	9.98658E-01	6.36314E-03	9.97608E-01	6.17675E-03

keno message number k6-123 execution terminated due to completion of the specified number of generations.
the matrix k-effective is the largest eigenvalue of the fission production by unit number matrix.

Figure F17.6.29 Example of k-effectives by generation

The user should examine this portion of the printed results to ensure 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. F17.6.29.

F17.6.24 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 and generation time calculated by KENO-VI are not kinetics parameters. 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. F17.4.3) 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. F17.6.30.

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.

F17.6.25 PLOT OF AVERAGE K-EFFECTIVE BY GENERATION RUN

This plot consists of average k-effectives plotted vs the number of generations run. The limits of 1 standard deviation are plotted on each side of each average k-effective. These average k-effectives are not necessarily the same as the average k-effectives described in Sect. F17.6.23. The code omits the k-effectives of the first *nskip* generations when the average k-effectives for this plot are calculated. Although the k-effective of the *nskip + 1* 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 fifth generation run. 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*+2 through the total number of generations. Figure F17.6.31 is an example of this type of plot. The primary use for this plot is to determine if the problem has source convergence difficulties.

sample problem 18 1f27 critical experiment

lifetime = 1.38892E-04 + or - 2.38902E-05
nu bar = 2.45311E+00 + or - 4.42297E-05
generation time = 3.91864E-05 + or - 8.51522E-07
average fission group = 1.45973E+01 + or - 6.94393E-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	deviation of variance (per cent)
3	.99901	+ or - .00648	.99253 to 1.00549	.98605 to 1.01197	.97957 to 1.01846	17500	23.2217
4	.99949	+ or - .00660	.99290 to 1.00609	.98630 to 1.01269	.97970 to 1.01928	17150	23.2287
5	.99888	+ or - .00671	.99217 to 1.00558	.98546 to 1.01229	.97876 to 1.01900	16800	23.5308
6	.99884	+ or - .00685	.99199 to 1.00569	.98514 to 1.01254	.97829 to 1.01939	16450	23.4550
7	.99839	+ or - .00699	.99140 to 1.00538	.98442 to 1.01236	.97743 to 1.01935	16100	23.5944
8	.99700	+ or - .00700	.99000 to 1.00400	.98300 to 1.01099	.97600 to 1.01799	15750	24.8216
9	.99479	+ or - .00679	.98799 to 1.00158	.98120 to 1.00838	.97440 to 1.01517	15400	26.6241
10	.99574	+ or - .00689	.98886 to 1.00263	.98197 to 1.00951	.97508 to 1.01640	15050	26.9694
11	.99598	+ or - .00705	.98894 to 1.00303	.98189 to 1.01008	.97484 to 1.01712	14700	26.8762
12	.99666	+ or - .00719	.98948 to 1.00385	.98229 to 1.01104	.97510 to 1.01823	14350	26.9732
17	.99650	+ or - .00790	.98860 to 1.00440	.98071 to 1.01230	.97281 to 1.02020	12600	28.9327
22	.99946	+ or - .00799	.99147 to 1.00744	.98349 to 1.01543	.97550 to 1.02341	10850	34.3414
27	1.00076	+ or - .00950	.99125 to 1.01026	.98175 to 1.01976	.97224 to 1.02927	9100	33.7045
32	1.01266	+ or - .00947	1.00319 to 1.02214	.99371 to 1.03161	.98424 to 1.04109	7350	38.8785
37	1.02267	+ or - .01080	1.01187 to 1.03347	1.00108 to 1.04426	.99028 to 1.05506	5600	42.2001
42	1.01646	+ or - .01091	1.00556 to 1.02737	.99465 to 1.03827	.98375 to 1.04918	3850	40.1501
47	1.01948	+ or - .01450	1.00499 to 1.03398	.99049 to 1.04848	.97599 to 1.06298	2100	56.3393

Figure F17.6.30 Example of the final k-effective edit

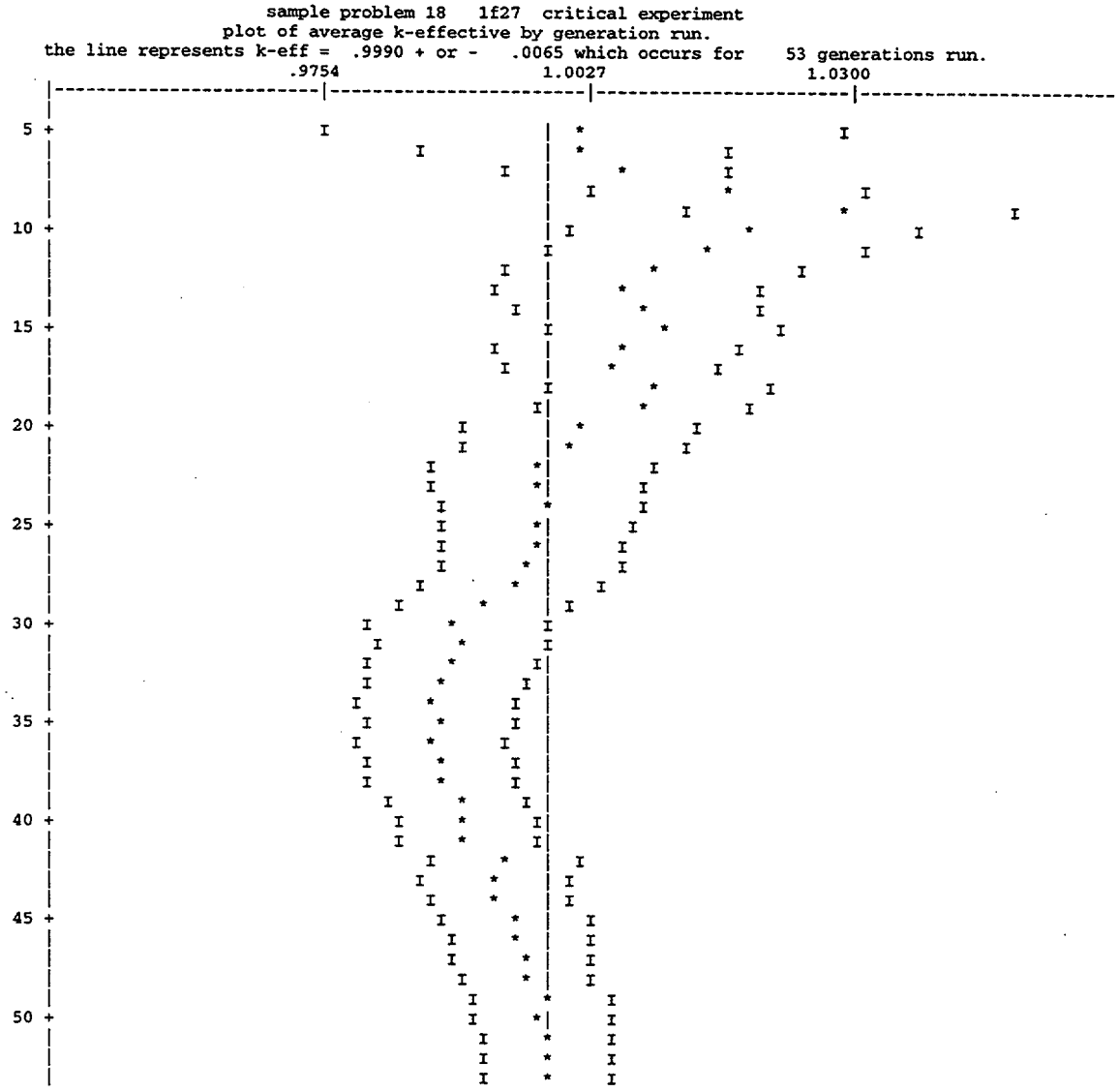


Figure F17.6.31 Sample plot of average k-effective by generation run

F17.6.26 PLOT OF AVERAGE K-EFFECTIVE BY GENERATION SKIPPED

This plot illustrates the average k-effective vs the number of generations skipped, as shown in Fig. F17.6.32. The limits of 1 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. The average k_{eff} and the minimum standard deviation (σ) of the system are listed at the top of the plot. These are determined by computing an average k_{eff} and σ for each number of generations skipped over the range of *nskip*+1 to *ngen*–2. The average k_{eff} and σ for a given number of generations skipped is calculated by batching the generation k_{eff}s differently and keeping the largest σ calculated. The plot is essentially a plot of the data described in Sect. F17.6.24 plotted over the range of *nskip + 1* to *ngen - 2*, where ngen is the number of generations run. It is useful for determining if source convergence has been achieved.

F17.6.27 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. Examples of the final edits of fissions, absorptions, and leakage are shown in Figs. F17.6.33 and F17.6.34.

If FAR=NO is specified, region-dependent fissions and absorptions are not printed. Figure F17.6.33 demonstrates the printout if FAR=NO.

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. F17.6.34. 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 percentage deviation for the fission fraction is the same as that of the fissions in the same group. The heading UNIT refers to the unit and REGION to the region number within the specified unit. The geometry regions are numbered sequentially within each unit, starting with 1. 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 percentage deviation is printed for each.

The sum of the leakage and absorptions printed for the system total should be close to 1. 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. F17.6.24. 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 weight loss is due to absorptions in the albedo reflector and leakage from the albedo reflector. No leakage is 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.

F17.6.28 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. F17.4.3). 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 3-D lattice. An example of the matrix k-effective by unit locations is given in Fig. F17.6.35. It is contained within two rows of asterisks to draw attention to it.

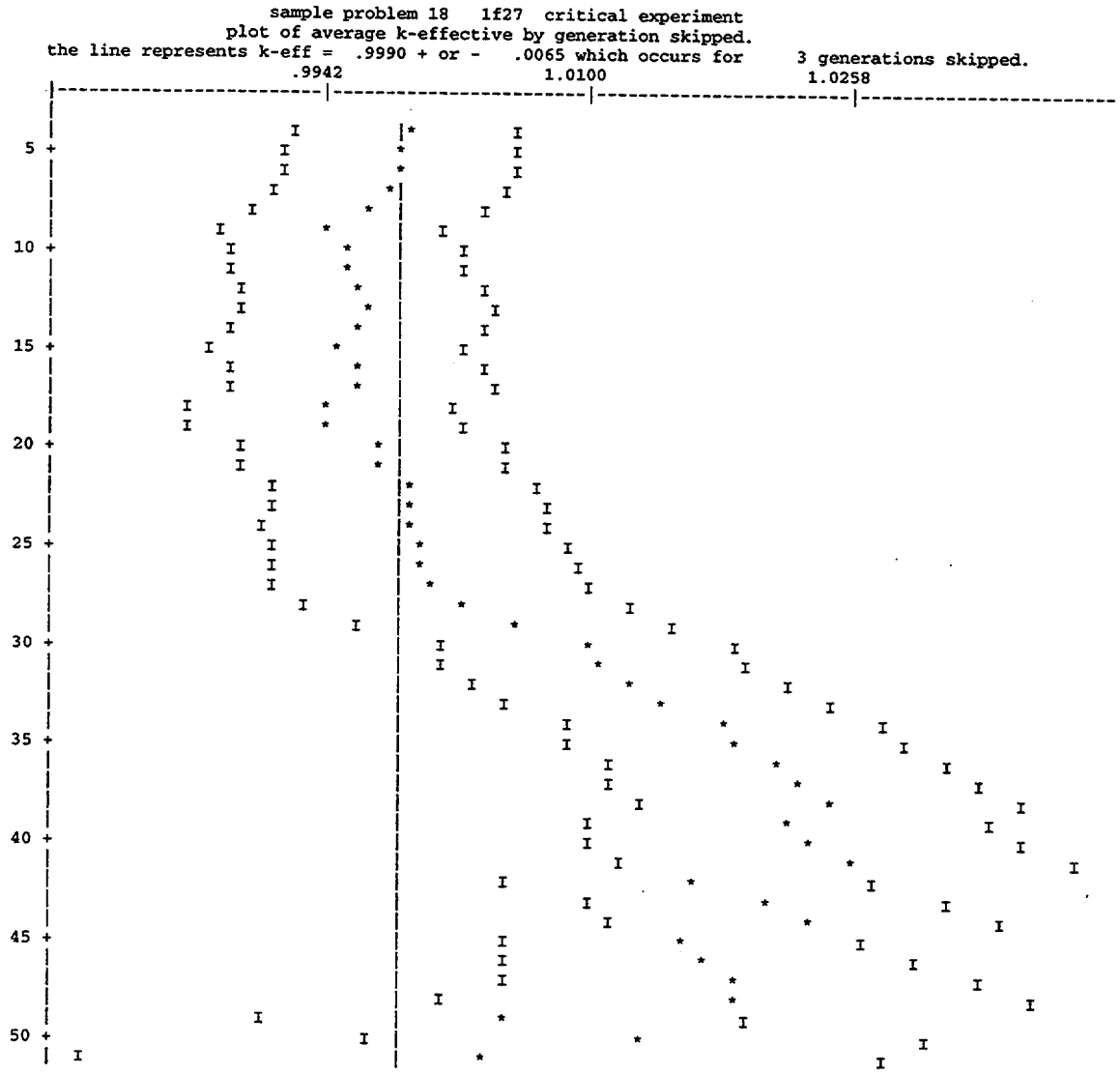


Figure F17.6.32 Sample plot of average k-effective by generation skipped

kenovi sample problem 2 case 2c8 bare with 8 unit types matrix cal

group	fission fraction	unit	region	fissions	percent deviation	absorptions	skipping 3 generations percent deviation	leakage	percent deviation
1	.1430			1.43511E-01	1.5000	5.06782E-02	1.5000	7.58911E-02	1.9424
2	.2406			2.41572E-01	.9401	9.84058E-02	.9401	1.47103E-01	1.4153
3	.1402			1.40755E-01	1.3565	6.09022E-02	1.3565	8.95706E-02	1.5378
4	.2313			2.32167E-01	1.1095	1.05587E-01	1.1095	1.46626E-01	1.0692
5	.1974			1.98175E-01	1.3537	9.41517E-02	1.3537	9.50593E-02	1.3858
6	.0467			4.69053E-02	2.7491	2.43717E-02	2.7491	1.18674E-02	3.8742
7	.0008			8.05573E-04	20.0775	4.34537E-04	20.0775	1.21626E-04	31.4931
8	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
9	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
10	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
11	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
12	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
13	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
14	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
15	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
16	.0000			0.00000E+00	.0000	0.00000E+00	.0000	0.00000E+00	.0000
system total =				1.00389E+00	.4635	4.34531E-01	.4709	5.66239E-01	.3376

elapsed time .83200 minutes
random number= 0000273130731F14

Figure F17.6.33 Sample of the final edit of fissions, absorptions, and leakage with region-dependent information suppressed

sample problem 18 1f27 critical experiment

group	fission fraction	unit	region	fissions	percent deviation	absorptions	percent deviation	skipping leakage	3 generations percent deviation
15	.1850			1.84769E-01	1.2445	1.05365E-01	1.2592	0.00000E+00	.0000
		1	1	1.84769E-01	1.2445	9.49967E-02	1.2445		
			2	0.00000E+00	.0000	0.00000E+00	.0000		
			3	0.00000E+00	.0000	7.06826E-04	3.2280		
			4	0.00000E+00	.0000	0.00000E+00	.0000		
		2	1	0.00000E+00	.0000	0.00000E+00	.0000		
			2	0.00000E+00	.0000	0.00000E+00	.0000		
			3	0.00000E+00	.0000	0.00000E+00	.0000		
			4	0.00000E+00	.0000	0.00000E+00	.0000		
			5	0.00000E+00	.0000	0.00000E+00	.0000		
		3	1	0.00000E+00	.0000	0.00000E+00	.0000		
			2	0.00000E+00	.0000	4.89387E-03	2.2892		
			3	0.00000E+00	.0000	2.50874E-03	5.0362		
			4	0.00000E+00	.0000	1.02395E-03	15.1271		
			5	0.00000E+00	.0000	5.30936E-04	43.4492		
			6	0.00000E+00	.0000	7.03715E-04	69.8467		
16	.6063			6.05662E-01	.8630	7.29098E-01	7.2582	8.84374E-03	100.0000
		1	1	6.05662E-01	.8630	3.02620E-01	.8630		
			2	0.00000E+00	.0000	0.00000E+00	.0000		
			3	0.00000E+00	.0000	8.38855E-03	2.0907		
			4	0.00000E+00	.0000	0.00000E+00	.0000		
		2	1	0.00000E+00	.0000	0.00000E+00	.0000		
			2	0.00000E+00	.0000	0.00000E+00	.0000		
			3	0.00000E+00	.0000	0.00000E+00	.0000		
			4	0.00000E+00	.0000	0.00000E+00	.0000		
			5	0.00000E+00	.0000	0.00000E+00	.0000		
		3	1	0.00000E+00	.0000	0.00000E+00	.0000		
			2	0.00000E+00	.0000	1.57178E-01	2.2761		
			3	0.00000E+00	.0000	1.20358E-01	3.8172		
			4	0.00000E+00	.0000	6.92781E-02	16.3281		
			5	0.00000E+00	.0000	4.42942E-02	51.5053		
			6	0.00000E+00	.0000	2.69816E-02	67.5988		
system total =				9.99011E-01	.6488	9.73385E-01	5.5176	2.10446E-02	42.6261

the weight lost in the albedo portion of the problem = 1.3472E-03 + or - .0003
elapsed time 6.33600 minutes
random number= 000035AB2AB34C1B

Figure F17.6.34 Sample of the final edit of fissions, absorptions, and leakage by region

```
kenovi sample problem 2  case 2c8 bare with 8 unit types matrix cal
*****************************************************************************************************
 position k-effective=  1.00224E+00  + or -  7.92895E-03
 the position k-effective is the largest eigenvalue of the fission production by position index matrix.
*****************************************************************************************************
```

Figure F17.6.35 Example of matrix k-effective by unit location

F17.6.29 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 array size + 1 squared. Thus for a $2 \times 2 \times 2$ array there are 81 entries, and for a $4 \times 4 \times 4$ array there are 4225 entries in the fission production matrix by position index. The 0th position in the matrix contains information for all neutrons outside the array. An example of the fission production matrix by position index for a $2 \times 2 \times 2$ array is shown in Fig. F17.6.36.

The position index definition is given in Sect. F17.6.31. 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.

```
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.

(  0,  0) 0.00E+00  (  0,  1) 0.00E+00  (  0,  2) 0.00E+00  (  0,  3) 0.00E+00  (  0,  4) 0.00E+00  (  0,  5) 0.00E+00
(  0,  6) 0.00E+00  (  0,  7) 0.00E+00  (  0,  8) 0.00E+00

(  1,  0) 0.00E+00  (  1,  1) 7.30E-01  (  1,  2) 5.43E-02  (  1,  3) 5.68E-02  (  1,  4) 2.92E-02  (  1,  5) 6.78E-02
(  1,  6) 2.52E-02  (  1,  7) 2.51E-02  (  1,  8) 1.85E-02

(  2,  0) 0.00E+00  (  2,  1) 5.38E-02  (  2,  2) 7.20E-01  (  2,  3) 2.97E-02  (  2,  4) 5.34E-02  (  2,  5) 2.54E-02
(  2,  6) 6.98E-02  (  2,  7) 1.80E-02  (  2,  8) 2.53E-02

(  3,  0) 0.00E+00  (  3,  1) 5.30E-02  (  3,  2) 2.96E-02  (  3,  3) 7.20E-01  (  3,  4) 5.42E-02  (  3,  5) 2.53E-02
(  3,  6) 1.97E-02  (  3,  7) 6.84E-02  (  3,  8) 2.40E-02

(  4,  0) 0.00E+00  (  4,  1) 2.93E-02  (  4,  2) 5.53E-02  (  4,  3) 5.35E-02  (  4,  4) 7.32E-01  (  4,  5) 1.88E-02
(  4,  6) 2.62E-02  (  4,  7) 2.49E-02  (  4,  8) 6.77E-02

(  5,  0) 0.00E+00  (  5,  1) 6.82E-02  (  5,  2) 2.44E-02  (  5,  3) 2.51E-02  (  5,  4) 1.98E-02  (  5,  5) 7.23E-01
(  5,  6) 5.78E-02  (  5,  7) 5.30E-02  (  5,  8) 3.11E-02

(  6,  0) 0.00E+00  (  6,  1) 2.36E-02  (  6,  2) 6.78E-02  (  6,  3) 2.03E-02  (  6,  4) 2.52E-02  (  6,  5) 5.31E-02
(  6,  6) 7.30E-01  (  6,  7) 2.88E-02  (  6,  8) 5.40E-02

(  7,  0) 0.00E+00  (  7,  1) 2.56E-02  (  7,  2) 2.02E-02  (  7,  3) 6.80E-02  (  7,  4) 2.62E-02  (  7,  5) 5.41E-02
(  7,  6) 2.89E-02  (  7,  7) 7.20E-01  (  7,  8) 5.47E-02

(  8,  0) 0.00E+00  (  8,  1) 2.12E-02  (  8,  2) 2.28E-02  (  8,  3) 2.53E-02  (  8,  4) 7.02E-02  (  8,  5) 2.92E-02
(  8,  6) 5.64E-02  (  8,  7) 5.67E-02  (  8,  8) 7.13E-01
```

Figure F17.6.36 Sample fission production matrix by position index

F17.6.30 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 (3-D) lattice representing the physical problem being analyzed. The position index is defined in Sect. F17.6.31. An example of the source vector by position index is shown in Fig. F17.6.37. The average self-multiplication by array position is the overall average of the self-multiplication of all units used in the problem.

```
kenovi sample problem 2  case 2c8 bare with 8 unit types matrix cal
  source vector by position index

                    index     vector
                      1    1.30419E-01
                      2    1.21533E-01
                      3    1.21295E-01
                      4    1.30846E-01
                      5    1.21621E-01
                      6    1.24876E-01
                      7    1.26689E-01
                      8    1.22721E-01
                      9    0.00000E+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 7.28552E-01 + or - 2.55906E-03
```

Figure F17.6.37 Example of source vector by position index

F17.6.31 COFACTOR K-EFFECTIVE BY POSITION INDEX

These data are printed by subroutine MATRIX only if MKP=YES and CKP=YES is specified in the parameter data. This input means that the fission production matrix is collected by position index. An example of the cofactor k-effective by position index is shown in Fig. F17.6.38. See Appendix F17. D for a description of the problem used for the example.

kenovi sample problem 2 case 2c8 bare with 8 unit types matrix cal

position index	position x	y	z	unit	cofactor k-effective	deviation
0	0	0	0	0	1.00003E+00	3.28099E-03
1	1	1	1	1	9.59766E-01	3.42705E-03
2	2	1	1	2	9.60815E-01	3.48681E-03
3	1	2	1	3	9.61649E-01	3.52555E-03
4	2	2	1	4	9.63183E-01	3.50292E-03
5	1	1	2	5	9.60190E-01	3.44069E-03
6	2	1	2	6	9.61163E-01	3.51237E-03
7	1	2	2	7	9.62489E-01	3.52625E-03
8	2	2	2	8	9.61997E-01	3.43740E-03

Figure F17.6.38 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 (3-D) lattice. POSITION is the x, y, and z location within the lattice. UNIT is the unit located at the specified location in the lattice. Thus in Fig. F17.6.36, Unit 1 is located at the lower left-hand front corner of the array or 3-D 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.

F17.6.32 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. F17.4.3). An example of the matrix k-effective by unit is given in Fig. F17.6.39.

```
          kenovi sample problem 2  case 2c8 bare with 8 unit types matrix cal

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

 unit k-effective=  1.00003E+00  + or -  3.28103E-03
 the unit k-effective is the largest eigenvalue of the fission production by unit number matrix.

*****************************************************************************************************************
```

Figure F17.6.39 Example of matrix k-effective by unit number

F17.6.33 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 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. F17.6.40.

```
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) 7.30E-01  ( 1, 2) 5.43E-02  ( 1, 3) 5.68E-02  ( 1, 4) 2.92E-02  ( 1, 5) 6.78E-02  ( 1, 6) 2.52E-02
( 1, 7) 2.51E-02  ( 1, 8) 1.85E-02  ( 1, 9) 0.00E+00

( 2, 1) 5.38E-02  ( 2, 2) 7.20E-01  ( 2, 3) 2.97E-02  ( 2, 4) 5.34E-02  ( 2, 5) 2.54E-02  ( 2, 6) 6.98E-02
( 2, 7) 1.80E-02  ( 2, 8) 2.53E-02  ( 2, 9) 0.00E+00

( 3, 1) 5.30E-02  ( 3, 2) 2.96E-02  ( 3, 3) 7.20E-01  ( 3, 4) 5.42E-02  ( 3, 5) 2.53E-02  ( 3, 6) 1.97E-02
( 3, 7) 6.84E-02  ( 3, 8) 2.40E-02  ( 3, 9) 0.00E+00

( 4, 1) 2.93E-02  ( 4, 2) 5.53E-02  ( 4, 3) 5.35E-02  ( 4, 4) 7.32E-01  ( 4, 5) 1.88E-02  ( 4, 6) 2.62E-02
( 4, 7) 2.49E-02  ( 4, 8) 6.77E-02  ( 4, 9) 0.00E+00

( 5, 1) 6.82E-02  ( 5, 2) 2.44E-02  ( 5, 3) 2.51E-02  ( 5, 4) 1.98E-02  ( 5, 5) 7.23E-01  ( 5, 6) 5.78E-02
( 5, 7) 5.30E-02  ( 5, 8) 3.11E-02  ( 5, 9) 0.00E+00

( 6, 1) 2.36E-02  ( 6, 2) 6.78E-02  ( 6, 3) 2.03E-02  ( 6, 4) 2.52E-02  ( 6, 5) 5.31E-02  ( 6, 6) 7.30E-01
( 6, 7) 2.88E-02  ( 6, 8) 5.40E-02  ( 6, 9) 0.00E+00

( 7, 1) 2.56E-02  ( 7, 2) 2.02E-02  ( 7, 3) 6.80E-02  ( 7, 4) 2.62E-02  ( 7, 5) 5.41E-02  ( 7, 6) 2.89E-02
( 7, 7) 7.20E-01  ( 7, 8) 5.47E-02  ( 7, 9) 0.00E+00

( 8, 1) 2.12E-02  ( 8, 2) 2.28E-02  ( 8, 3) 2.53E-02  ( 8, 4) 7.02E-02  ( 8, 5) 2.92E-02  ( 8, 6) 5.64E-02
( 8, 7) 5.67E-02  ( 8, 8) 7.13E-01  ( 8, 9) 0.00E+00

( 9, 1) 0.00E+00  ( 9, 2) 0.00E+00  ( 9, 3) 0.00E+00  ( 9, 4) 0.00E+00  ( 9, 5) 0.00E+00  ( 9, 6) 0.00E+00
( 9, 7) 0.00E+00  ( 9, 8) 0.00E+00  ( 9, 9) 0.00E+00
```

Figure F17.6.40 An example of the fission probability matrix by unit

F17.6.34 SOURCE VECTOR BY UNIT NUMBER

These data are printed by subroutine MATRIX only if MKU=YES and 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. F17.6.41. 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. F17.6.24 if the problem utilizes multiple units, the system is reflected, or a differential albedo is used in the problem.

```
kenovi sample problem 2  case 2c8 bare with 8 unit types matrix cal

   source vector by unit
     unit      vector
      1     1.26606E-01
      2     1.23931E-01
      3     1.24397E-01
      4     1.28599E-01
      5     1.23625E-01
      6     1.29880E-01
      7     1.22250E-01
      8     1.20713E-01
      9     0.00000E+00

                    average self multiplication by unit
 the number of next generation neutrons produced in a unit by
 a neutron born in that same unit is    7.23506E-01 + or -  9.95285E-04
```

Figure F17.6.41 Example of the source vector by unit

F17.6.35 COFACTOR K-EFFECTIVE BY UNIT NUMBER

Cofactor k-effectives are printed by subroutine MATRIX only if MKU=YES and CKU=YES is specified 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 calculation is accomplished 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. F17.6.42.

```
kenovi sample problem 2  case 2c8 bare with 8 unit types matrix cal

                  cofactor
     unit       k-effective      deviation
       1        9.59766E-01    3.42705E-03
       2        9.60815E-01    3.48681E-03
       3        9.61649E-01    3.52555E-03
       4        9.63183E-01    3.50292E-03
       5        9.60190E-01    3.44069E-03
       6        9.61163E-01    3.51237E-03
       7        9.62489E-01    3.52625E-03
       8        9.61997E-01    3.43740E-03
       9        1.00003E+00    3.28099E-03
```

Figure F17.6.42 Example of cofactor k-effective by unit number

F17.6.36 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. F17.4.3. 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. F17.6.43.

```
sample problem 22  case 2c8 bare with 3 nested, equal volume holes

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

 hole k-effective=  9.97666E-01  + or -  2.13237E-03
 the hole k-effective is the largest eigenvalue of the fission production by hole number matrix.

*****************************************************************************************************************
```

Figure F17.6.43 Example of matrix k-effective by hole number

F17.6.37 FISSION PRODUCTION BY HOLE NUMBER MATRIX

This fission production matrix is collected by hole number. It is printed only if MKH=YES and FMH=YES were specified in the parameter data, Sect. F17.4.3. An example of this fission production matrix is given in Fig. F17.6.44. 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.
  (  0,  0) 3.85E-01  (  0,  1) 1.65E-01  (  0,  2) 1.96E-01  (  0,  3) 1.08E-01
  (  1,  0) 1.73E-01  (  1,  1) 5.51E-01  (  1,  2) 2.61E-01  (  1,  3) 1.43E-01
  (  2,  0) 2.10E-01  (  2,  1) 2.74E-01  (  2,  2) 3.49E-01  (  2,  3) 1.90E-01
  (  3,  0) 2.47E-01  (  3,  1) 1.87E-01  (  3,  2) 2.36E-01  (  3,  3) 2.48E-01
```

Figure F17.6.44 Example of fission production matrix by hole

F17.6.38 SOURCE VECTOR BY HOLE NUMBER

This information is printed by subroutine MATRIX only if MKH=YES is specified in the parameter data, Sect. F17.4.3. 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. F17.6.45. 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  critical experiment assembled using holes

   source vector by hole

     hole      vector
      0     2.48086E-01
      1     3.22897E-01
      2     2.64550E-01
      3     1.64466E-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    4.06887E-01  + or -  1.10484E-03
```

Figure F17.6.45 Example of source vector by hole number

F17.6.39 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. F17.4.3. The cofactor k-effective for a given hole is the k-effective of the system calculated without the fission source of that hole. This calculation 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. F17.6.46.

```
sample problem 22  case 2c8 bare with 3 nested, equal volume holes
                                   cofactor
          hole          unit      k-effective      deviation
            0             0      8.46517E-01     2.36886E-03
            1             1      7.20519E-01     2.11586E-03
            2             2      7.58170E-01     2.31495E-03
            3             3      8.70905E-01     2.22666E-03
```

Figure F17.6.46 Example of cofactor k-effective by hole number

F17.6.40 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. F17.4.3. It is the largest eigenvalue of the fission production matrix collected by array number. An example is given in Fig. F17.6.47. 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  critical experiment

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

 array k-effective=  1.01345E+00  + or -  8.24387E-03
 the array k-effective is the largest eigenvalue of the fission production by array number matrix.

*****************************************************************************************************************
```

Figure F17.6.47 Example of matrix k-effective by array number

F17.6.41 FISSION PRODUCTION BY ARRAY NUMBER MATRIX

The fission production matrix collected by array number is shown in Fig. F17.6.86. It is printed only if MKA=YES and FMA=YES are specified in the parameter data (Sect. F17.4.3).

```
 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.
  (  0,  0) 0.00E+00  (  0,  1) 0.00E+00  (  0,  2) 0.00E+00  (  0,  3) 0.00E+00  (  0,  4) 0.00E+00  (  0,  5) 0.00E+00
  (  1,  0) 0.00E+00  (  1,  1) 7.46E-01  (  1,  2) 6.45E-02  (  1,  3) 9.03E-02  (  1,  4) 1.23E-01  (  1,  5) 0.00E+00
  (  2,  0) 0.00E+00  (  2,  1) 1.43E-01  (  2,  2) 6.71E-01  (  2,  3) 9.07E-02  (  2,  4) 1.22E-01  (  2,  5) 0.00E+00
  (  3,  0) 0.00E+00  (  3,  1) 1.15E-01  (  3,  2) 5.97E-02  (  3,  3) 7.12E-01  (  3,  4) 1.12E-01  (  3,  5) 0.00E+00
  (  4,  0) 0.00E+00  (  4,  1) 1.08E-01  (  4,  2) 5.29E-02  (  4,  3) 7.85E-02  (  4,  4) 7.67E-01  (  4,  5) 0.00E+00
  (  5,  0) 0.00E+00  (  5,  1) 0.00E+00  (  5,  2) 0.00E+00  (  5,  3) 0.00E+00  (  5,  4) 0.00E+00  (  5,  5) 0.00E+00
```

Figure F17.6.48 An example of the fission production matrix by array number

F17.6.42 SOURCE VECTOR BY ARRAY NUMBER

This information is printed by subroutine MATRIX only if MKA=YES is specified in the parameter data (Sect. F17.4.3). 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. F17.6.49. The average self-multiplication by array number is the overall self-multiplication of all the arrays in the problem.

```
sample problem 18   1f27  critical experiment
  source vector by array
   array      vector
     0     0.00000E+00
     1     3.05788E-01
     2     1.46445E-01
     3     2.21002E-01
     4     3.26765E-01
     5     0.00000E+00

                    average self multiplication by array
the number of next generation neutrons produced in  an array by
a neutron born in that same array is   7.34530E-01  + or -  3.77283E-03
```

Figure F17.6.49 Example of source vector by array number

F17.6.43 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. F17.4.3). The cofactor k-effective for a given array is the k-effective of the system calculated without the fission source of that array. This calculation 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 F17.6.50 is an example of the cofactor k-effective by array number.

sample problem 18 1f27 critical experiment

array index	array number	cofactor k-effective	deviation
0	0	1.00487E+00	7.13587E-03
1	1	8.80707E-01	8.55838E-03
2	2	9.49413E-01	7.38712E-03
3	3	9.19747E-01	7.68909E-03
4	4	8.89557E-01	8.30549E-03
5	5	1.00487E+00	7.13587E-03

Figure F17.6.50 Example of cofactor k-effective by array number

F17.6.44 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. F17.6.51.

The UNIT is the unit 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.

kenovi sample problem 2 case 2c8 bare with 8 unit types matrix cal
**** fission densities ****

unit	region	fission density	percent deviation	total fissions
1	1	1.316E-01	2.44	1.316E-01
	2	0.000E+00	.00	0.000E+00
2	1	1.217E-01	1.88	1.217E-01
	2	0.000E+00	.00	0.000E+00
3	1	1.222E-01	2.41	1.222E-01
	2	0.000E+00	.00	0.000E+00
4	1	1.311E-01	2.06	1.311E-01
	2	0.000E+00	.00	0.000E+00
5	1	1.239E-01	1.79	1.239E-01
	2	0.000E+00	.00	0.000E+00
6	1	1.256E-01	2.14	1.256E-01
	2	0.000E+00	.00	0.000E+00
7	1	1.268E-01	2.38	1.268E-01
	2	0.000E+00	.00	0.000E+00
8	1	1.209E-01	2.63	1.209E-01
	2	0.000E+00	.00	0.000E+00

Figure F17.6.51 Example of the fission density edit

F17.6.45 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. F17.6.52.

sample problem 18 1f27 critical experiment

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	4.591E-02	2.60	3.484E-04	7.80	1.058E-02	3.55	1.269E-01	2.78
2	8.954E-02	1.72	6.445E-04	7.34	2.227E-02	3.16	2.575E-01	2.07
3	4.254E-02	2.28	2.735E-04	7.96	1.274E-02	4.16	1.144E-01	3.34
4	5.984E-02	1.70	4.034E-04	10.34	1.739E-02	3.77	1.498E-01	3.23
5	5.898E-02	1.42	3.564E-04	7.11	2.026E-02	3.75	1.646E-01	2.74
6	4.006E-02	1.88	2.456E-04	7.87	1.735E-02	4.18	1.216E-01	3.32
7	3.146E-02	1.92	2.252E-04	12.05	1.414E-02	4.05	1.065E-01	3.94
8	2.766E-02	1.84	1.745E-04	10.03	1.243E-02	4.62	9.311E-02	4.29
9	2.543E-02	1.81	1.783E-04	9.70	1.232E-02	5.31	8.840E-02	4.62
10	1.659E-02	2.51	1.266E-04	9.97	9.890E-03	5.67	6.730E-02	3.91
11	1.421E-02	2.48	9.449E-05	12.27	7.372E-03	5.50	5.593E-02	5.57
12	1.416E-02	2.76	8.552E-05	13.27	7.395E-03	5.24	5.707E-02	4.43
13	1.272E-02	2.54	8.081E-05	19.55	6.840E-03	5.59	5.130E-02	5.54
14	9.517E-03	2.39	6.425E-05	12.73	5.822E-03	7.08	4.114E-02	5.88
15	1.821E-02	2.14	1.947E-04	12.16	1.761E-02	4.15	1.094E-01	4.50
16	3.013E-02	1.98	7.287E-04	6.26	1.586E-01	2.06	9.496E-01	2.42

Figure F17.6.52 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 percentage deviation are printed for every energy group and every geometry region. The flux is in units of neutrons/cm^2/source neutron.

F17.6.46 FREQUENCY DISTRIBUTIONS

Subroutine FREAK is responsible for printing the frequency distributions that 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. F17.6.53.

```
kenovi sample problem 2  case 2c8 bare with 8 unit types matrix cal

                              frequency for generations    4 to  103
 0.8803 to 0.9034     *
 0.9034 to 0.9265     *****
 0.9265 to 0.9496     *****
 0.9496 to 0.9727     **************
 0.9727 to 0.9958     ************
 0.9958 to 1.0189     ************************
 1.0189 to 1.0420     ***************
 1.0420 to 1.0651     ***********
 1.0651 to 1.0882     *******
 1.0882 to 1.1113     *****
 1.1113 to 1.1344     *

                              frequency for generations   29 to  103
 0.8803 to 0.9034
 0.9034 to 0.9265     ***
 0.9265 to 0.9496     ****
 0.9496 to 0.9727     **********
 0.9727 to 0.9958     **********
 0.9958 to 1.0189     ****************
 1.0189 to 1.0420     ***********
 1.0420 to 1.0651     *********
 1.0651 to 1.0882     *******
 1.0882 to 1.1113     ****
 1.1113 to 1.1344     *

                              frequency for generations   54 to  103
 0.8803 to 0.9034
 0.9034 to 0.9265     ***
 0.9265 to 0.9496     ***
 0.9496 to 0.9727     *******
 0.9727 to 0.9958     *******
 0.9958 to 1.0189     ********
 1.0189 to 1.0420     ******
 1.0420 to 1.0651     *******
 1.0651 to 1.0882     *****
 1.0882 to 1.1113     ***
 1.1113 to 1.1344     *

                              frequency for generations   79 to  103
 0.8803 to 0.9034
 0.9034 to 0.9265     *
 0.9265 to 0.9496
 0.9496 to 0.9727     ****
 0.9727 to 0.9958     ***
 0.9958 to 1.0189     *****
 1.0189 to 1.0420     *****
 1.0420 to 1.0651     ***
 1.0651 to 1.0882     **
 1.0882 to 1.1113     **
 1.1113 to 1.1344

*****************************************************************************************************
   congratulations!  you have  successfully traversed the perilous path through keno vi in
0.40533 minutes

*****************************************************************************************************
```

Figure F17.6.53 Example of frequency distributions

F17.7 WARNING MESSAGES AND ERROR MESSAGES

KENO-VI prints warning and error messages that are identified by K6-, followed by a unique number, such as K6-2. 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.

F17.7.1 MESSAGES

K6-2 *** ERROR *** 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 INITAL 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.

K6-3 *** WARNING *** THE CROSS SECTION LIBRARY ON UNIT ____ HAS ____ ENERGY GROUPS.

This message is printed in subroutine INITAL after the call to PARAM. It is activated if the cross-section library has more than 300 energy groups. The largest standard cross-section library in the SCALE system contains 238 energy groups.

K6-4 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 F17.4.1 in the KENO-VI input outline.

K6-5 ***** 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.

K6-6 ***** 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.

K6-7 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.

K6-8 ***** 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.

K6-9 *** 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 PARTBL. 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-VI output.

K6-10 *** 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 PARTBL. It is basically 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-VI output.

K6-11 *** ERROR *** A BOUNDARY GEOMETRY WORD WAS NOT SPECIFIED FOR UNIT___.

This self-explanatory message is from subroutine KENOG. A BOUNDARY card must be associated with each unit identifying the outermost region of the unit.

K6-12 *** WARNING *** 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 PARTBL. NBK should not be entered as input data unless it is known that the default value is inadequate.

K6-13 *** ERROR *** MEDIA DATA MUST BE ENTERED FOR UNIT ____.

This message is from subroutine KENOG. Either another UNIT card or an END GEOM card was encountered prior to inputting MEDIA data for the current unit.

K6-14 ***** 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. F17.4.10 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 K6-14 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.

K6-15 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.

K6-16 *** 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 situation 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.

K6-17 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. F17.4.4 for a list of accepted geometry words.

K6-18 ***** ERROR ***** INVALID INDEX INTO KEYWRD ARRAY. INP=_____, CHECK SUBROUTINE KENOG FOR A CODE ERROR.

This message is from subroutine KENOG. This message usually means that a code error was introduced when changes were made to the program. Check subroutine KENOG, function LCOMPR, and KEYWRD data block.

K6-19 ***** ERROR ***** THE GLOBAL UNIT HAS NOT BEEN SPECIFIED.

This self-explanatory message from subroutine KENOG indicates that a global unit has not been specified. The outermost unit in the problem must be specified as the GLOBAL UNIT. See Sect. F17.4.4 to determine the correct specification.

K6-20 ***** ERROR ***** GEOMETRY WORD INDEX OUT OF RANGE ON SCRATCH UNIT _ _ FOR GEOMETRY WORD NUMBER ___ INP=___ GWRD=___.

This message from subroutine READGM means that INP is negative or larger than NP 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.

K6-21 ____ 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. F17.4.7 for assistance in determining the proper procedure for entering biasing data.

K6-22 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.

K6-23 *** 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 F17.4.5 for the weights that are available on the standard data set. The procedure for entering weights from cards is explained in Sect. F17.4.7.

K6-24 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. F17.4.7 for assistance.

K6-25 THE FIRST NUMBER IS THE STORAGE NEEDED; THE SECOND, 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.

K6-26 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 ten numbers as shown in the companion message K6-27. 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.

K6-27 LTYPE=_____ IX1=_____ IX2=_____ INCX=_____ IY1=_____ IY2=_____ INCY=_____ IZ1=_____ IZ2=_____ INCZ=_____

This message is a companion message for K6-26. It indicates how the unit orientation data description for the set named in K6-26 was entered. See Sect. F17.4.5 for information pertaining to unit orientation data.

K6-28 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. F17.4.5 for input instructions.

K6-29 *** ERROR *** THE ARRAY SIZE HAS BEEN SPECIFIED INCORRECTLY.
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. F17.4.5. If a unit orientation data description is to be entered, NBXMAX, NBYMAX, and NBZMAX must all be greater than zero.

K6-30 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.

K6-31 ____ 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. F17.4.5 for additional assistance. A STOP 101 is executed when this message is printed.

K6-32 *** 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. K6-33 is a companion message.

K6-33 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. F17.4.5 for assistance in correcting the error(s).

K6-34 ***** 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.

K6-35 ***** 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 F17.4.3 in Sect. F17.4.6 for a list of acceptable face code names.

K6-36 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.

K6-37 ***** 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).

K6-38 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.

K6-39 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. F17.4.8.

K6-40 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. F17.4.8 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.

K6-41 *** 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. F17.4.8 for assistance concerning start data.

K6-42 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 _____.

K6-43 ***** AN END OF FILE WAS ENCOUNTERED BEFORE AN END DATA WAS FOUND. THE PROBLEM WILL NOT RUN. *****

This message is from subroutine DATAIN. It occurs when an end of file is encountered while reading data.

K6-44 ***** ERROR ***** SECTOR DATA WAS NOT FOUND FOR ARRAY

This self-explanatory message is from subroutine ARRAY. This message usually means that sector data was left off the ARRAY media care. See Sect. F17.4.4 for assistance is specifying an array.

K6-45 ***** ILLEGAL DATA BLOCK IDENTIFIER _____ _____ *****

This message from subroutine DATAIN is printed whenever an invalid data block identifier is encountered. This situation 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 F17.4.1 (Sect. F17.4.1).

Consider the following examples:

ERROR MESSAGE EXAMPLE 1

```
READ PARAM TME=2.9 FLX=YES XSC=38 END PARAM
READ GEOM  GLOBAL UNIT 1
CYLINDER  5.0 5.0 -5.0  MEDIA 1 1 1  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   5.0 5.0 -5.0  MEDIA 1 1 1  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.

K6-46 ***** 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.
A STOP 152 is executed when this message is printed.

K6-47 *** _____ TRANSFERS FOR MIXTURE _____ WERE CORRECTED FOR BAD MOMENTS.

This message from subroutine MAKANG indicates moments were corrected to eliminate negative probabilities for calculated angles. If the moment was changed by more than e ps times the moment, a K6-60 or K6-64 message is printed. If eps is very small, any change without an accompanying message is trivial. The K6-60 or K6-64 messages can be used to determine if the affected transfer and correction are significant. Most messages are caused by the cross-sections being in single precision and the moments calculations being done in double precision. Generally, as the number of energy groups increase so does the number of corrected transfers for a given mixture.

K6-48 *** 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.

K6-49 _____ 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.

K6-50 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.

K6-51 ***** 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 F17.4.4 (Sect. F17.4.6).

K6-52 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.

K6-53 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.

K6-54 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.

K6-55 **** 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 it was not in the cross-section library.

K6-56 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.

K6-57 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.

K6-58 This message appears in two forms, listed below:

K6-58 ** ERROR **** ERROR ** INVALID BIAS ID ____ IN REGION ____ OF UNIT ____.

This form of message K6-58 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 K6-58 will also be printed.

K6-58 *** ERROR *** INVALID BIAS ID IN REGION ABOVE.

This form of message K6-58 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 K6-58 is printed. Review Sect. F17.4.4 for correct geometry data specification information.

K6-59 THE CALCULATION WAS TERMINATED BECAUSE OF EXCESSIVE SPLITTING

This message from subroutine GUIDE is printed only if message K6-128 is printed 50 or more times for a given generation. This message 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. F17.4.7), it should be checked carefully.

K6-60 THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE ____ HAS BAD MOMENTS
FOR THE TRANSFER FROM GROUP ____ TO GROUP ____.
____ MOMENTS WERE ACCEPTED.

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)

THE WEIGHTS/ANGLES FOR THIS DISTRIBUTION ARE
(W_1) (W_2) ... (Wm)
(A_1) (A_2) ... (Am)
THE MOMENTS CORRESPONDING TO THIS DISTRIBUTION ARE
(M_1) (M_2) ... (M_n)

This message from subroutine BADMOM indicates that the moments from the cross-section data are incorrect for the group transfer shown. The code replaces the incorrect moments with acceptable moments and proceeds normally. The moments printed in the last line of the message should match those printed in the eighth line. 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. F17.4.10. See Sect. F17.5.4.4 for assistance in determining an appropriate value.

K6-61 *****ERROR***** _____ IS NOT A VALID UNIT NUMBER, A UNIT NUMBER MUST BE A POSTIVE INTEGER.

This message from subroutine KENOG is self-explanatory. A unit number must be an integer greater than 0.

K6-62 ***** ERROR ***** INSUFFICIENT DATA FOLLOWING THE KEYWORD ARRAY

This self-explanatory message is from subroutine ARRAY. The data following the keyword ARRAY are misspelled or incomplete. See Sect. F17.4.5 for assistance in determining the correct method for inputting array data.

K6-63 NOT ENOUGH IOs ARE LEFT TO PRINT THE FLUXES.

This message from subroutine FITFLX is self-explanatory. The number of IOs specified in the job control language must be increased, and the problem must be rerun if printed fluxes are required. All other results are unaffected by the failure to print the fluxes.

K6-64 THE ANGULAR SCATTERING DISTRIBUTION FOR MIXTURE ____ HAS BAD MOMENTS FOR THE TRANSFER FROM GROUP _____ TO GROUP _____. MOMENTS WERE ACCEPTED. THE P0 COEFFICIENT IS ____.

This message from subroutine BADMOM is printed to inform the user that the cross sections were altered by the code because the moments from the cross-section data were incorrect for the group transfer shown. The P_0 coefficient was larger than the "cross-section message cutoff parameter," EPS, but the relative change in the moments was smaller than EPS.

K6-65 ** WARNING ** AN AMPX WORKING LIBRARY WAS SPECIFIED ON UNIT ____ BUT NO MIXING DATA WAS READ. MIXED CROSS SECTIONS FORM UNIT __ WILL BE USED. ** WARNING**

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. F17.4.3 for parameter data and Sect. F17.4.10 for mixing table information.

K6-66 ** WARNING ** ____ MIXTURES WERE REQUESTED IN THE GEOMETRY, BUT ONLY ____ MIXTURES ARE ON THE MIXED CROSS SECTION LIBRARY. ** WARNING **

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. F17.4.3 for the specification of the unit number of the mixed cross-section library (XSC=), Sect. F17.4.10 for the specification of the mixing table, and Sect. F17.4.4 to determine the mixtures used in the geometry region data.

K6-67 ***** ERROR ***** 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.

K6-68 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-VI calculation. A STOP 106 is executed when this message is printed.

K6-69 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.

K6-70 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 K6-46 and K6-48. 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.

K6-71 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.

K6-72 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.

K6-73 ***** ERROR ***** ____ MIXTURES WERE REQUESTED IN THE GEOMETRY DATA, BUT ONLY ____ OF THESE WERE FOUND IN THE MIXED CROSS SECTIONS. ***** ERROR *****

This self-explanatory message is from subroutine BOOK3. Either the wrong mixed cross-section data set (XSC= from the parameter data, Sect. F17.4.3) is being used, or one or more mixture numbers are in error in the geometry region data (see Sect. F17.4.4).

K6-74 ***** _____ WORDS OF STORAGE WERE ALLOCATED TO RUN THIS PROBLEM, BUT _ ___ WORDS ARE REQUIRED FOR THE INPUT DATA. *****

This self-explanatory message is from subroutine BOOK3. Additional storage is required to run this problem. Alter the job control language to request sufficient storage.

K6-75 *** 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.

K6-76 *** NOT ENOUGH STORAGE TO PRINT TWO DIMENSIONAL CROSS SECTIONS. _ ___ LOCATIONS 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.

K6-77 *** 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.

K6-78 ***** 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.

K6-79 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.

K6-80 _____ 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.

K6-81 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.

K6-82 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.

K6-83 ***** 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 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 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. F17.4.4 and F17.4.5.

K6-84 ********** 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 that was not defined in the geometry region data. Verify the geometry region data and the array definition data as described in Sects. F17.4.4 and F17.4.5.

K6-85 UNIT ___ AT POSITION X=___ Y=___ Z=___ IN UNIT ORIENTATION ARRAY NUMBER _ _ IS INVALID BECAUSE IT IS LESS THAN ZERO OR LARGER THAN THE LARGEST UNIT NUMBER IN THE INPUT DATA.

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, misspelled data and omitted data.) See Sect. F17.4.5 for additional information.

K6-86 GEOMETRY WORD INDEX OUT OF RANGE. 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 27 to be valid. If the geometry words (see *fgeom* in Sect. F17.4.4) are correct, this message is due to a code error that has been introduced when changes were made to the code.

K6-87 ***** ERROR ***** INSUFFICIENT DATA FOLLOWED THE KEYWORD "MEDIA".

This message from subroutine MEDIA is self-explanatory. Either the material, bias ID or sector data were not included on the media card, or noninteger data were inadvertently entered. See Sect. F17.4.4 for additional information on the type of data required on a MEDIA card.

K6-88 ***** ERROR ***** SECTOR DATA WAS NOT FOUND FOR MEDIA ___.

This message from subroutine MEDIA is self-explanatory. Either the material, bias ID or sector data were not included on media card ___ or noninteger data were inadvertently entered. See Sect. F17.4.4 for additional information on the type of data required on a MEDIA card.

K6-89 *** ERROR *** VOLUME HAS BEEN MULTIPLY DEFINED. NEUTRON ___ UNIT ___ AT POINT X1= ___ Y1= ___ Z1= ___ MOVING IN DIRECTION U= ___ V= ___ W= ___ THIS POINT WAS LOCATED INSIDE THE FOLLOWING REGIONS.
REGION NUMBER = ___
. . .
. . .
REGION NUMBER = ___.

This message is from subroutine TRACK. It is caused by incorrectly specifying the vector definition arrays in the media cards of a unit. It can be caused by incorrectly specifying regions that share a common boundary so the boundary between regions is contained in only one region. See Sect. F17.4.4 for more information concerning vector definition arrays on MEDIA cards.

K6-90 THE UNIT ORIENTATION DESCRIPTION MAY APPEAR TO BE INCORRECT BECAUSE THE ARRAY SIZE WAS INCORRECTLY SPECIFIED.

This warning message from subroutine PRTLBA occurs if NBXMAX, NBYMAX or NBZMAX (NUX=, NUY=, NUZ=) from the array data, Sect. F17.4.5, was incorrectly specified.

K6-95 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 cylinder or sphere is larger than the radius. See Sect. F17.4.4 for assistance in specifying the geometry correctly.

K6-96 THE VOLUME DEFINED BY GEOMETRY CARD ____ IS NEGATIVE.

This message from subroutine VOLUME is printed whenever a negative volume is calculated. This can result from the positive dimension being smaller than the negative dimension on a face of a geometry region. It can also 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. F17.4.4.

K6-97 ERROR ERROR. THE VOLUME FOR UNIT ____ IS NEGATIVE.

This message is from subroutine VOLUME. A negative volume for a unit can be caused by having a unit consisting of one region and having a positive dimension smaller than the negative dimension on one or more faces. Message K6-95 or K6-96 may accompany this message. See Sect. F17.4.4 for assistance in specifying the geometry data correctly.

K6-99 This message appears in three forms, listed below:

K6-99 *** ERROR IN UNIT ____ *** THE LAST GEOMETRY REGION OF A UNIT UTILIZED IN THE UNIT ORIENTATION DESCRIPTION OF THE ARRAY DATA MUST BE A CUBOID.

This message is from subroutine ARASIZ. It can occur when a single unit problem whose outer region is not a cuboid is specified as a 1 × 1 × 1 array. To eliminate this message, add a cuboidal outer region containing void, or remove the array data. If the problem is an array problem with the array type either defaulted or specified as cuboidal, be sure each unit used in the unit orientation description ends with a cuboid. See Sects. F17.4.4 and F17.4.5.

K6-99 *** ERROR IN UNIT ____ *** THE LAST GEOMETRY REGION OF A UNIT UTILIZED IN THE UNIT ORIENTATION DESCRIPTION OF THE ARRAY DATA MUST BE A HEXPRISM.

This message is from subroutine ARASIZ. It can occur when a single unit problem whose outer region is not a hexprism is specified as a 1x1x1 array. To eliminate this message, add a hexagonal outer region containing void, or remove the array data. If the problem is an array problem with the array type specified as triangular, be sure each unit used in the unit orientation description ends with a hexprism. See Sects. F17.4.4 and F17.4.5.

K6-99 *** ERROR IN UNIT _____ *** THE ARRAY TYPE SPECIFIED IN THIS UNIT IS UNDEFINED. SPECIFY IF THE ARRAY TYPE IS CUBOID OR HEXPRISM.

This message is from subroutine ARASIZ. It can occur if the array type, TYP=, is incorrectly specified in the array data. Check the spelling of the array type. See Sect. F17.4.5.

K6-100 THIS PROBLEM WILL NOT BE RUN BECAUSE ERRORS WERE ENCOUNTERED IN THE INPUT DATA.

This message is from subroutine BOOK5 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.

K6-101 *** 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.

K6-102 *** ERROR *** 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. XSM, XSP, YSM, YSP, ZSM AND ZSP MUST FIT WITHIN THE OUTER REGION OF THE REFLECTOR. *** ERROR ***

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 cuboid, data must be input to specify an imaginary cube or cuboid within this outer region. See Sect. F17.4.8 for assistance in specifying these data.

K6-103 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 F17.4.6 (Sect. F17.4.8).

K6-104 A POSIT ERROR INDICATES THAT THE POINT X=_____ Y=_____ Z=_____ DOES NOT OCCUR WITHIN UNIT _____. IF XSM, XSP, YSM, YSP, ZSM AND ZSP ARE ENTERED IN THE START DATA, VERIFY THAT THEY FIT WITHIN THE OVERALL COORDINATES OF THE SYSTEM.
THE OVERALL COORDINATES MAY NOT BE PRINTED FOR A BARE ARRAY.

If XSM, XSP, YSM, YSP, ZSM, ZSP were entered in the start data. Verify that they fit within the overall coordinates of the system. The overall coordinates may not be printed for a bare array. 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.

K6-105 ***** WARNING, ONLY _____ INDEPENDENT STARTING POSITIONS WERE GENERATED. *****

This message is from subroutine START. KENO-VI must have *npb* (NPG=, see parameter data, Sect. F17.4.3) 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. F17.4.8 for assistance.

K6-106 POSIT ERROR --- UNIT ___ HAS MULTIPLY DEFINED SPACE
X= ____ Y= ____ Z= ____ SECTOR INSIDE ____

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 improperly specifying media records. Check the MEDIA records in the specified units for multiply defined volumes.

X, Y, Z is the location of the neutron and SECTOR INSIDE is the sectors which contain that point. See Sect. F17.4.4 for correct geometry words.

K6-107 POSIT ERROR ***UNIT ____ HAS AN UNDEFINED SPACE. X=_____ Y=_____ Z=_____ SECTOR INSIDE _____

This message from subroutine POSIT may result from not specifying all volumes in the unit using media records. It could also occur from start data if a particle is attempting to start outside the global unit or in an undefined space. Check your geometry for the specified unit.

K6-108 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. F17.4.8 for the correct start data specification.

K6-109 ***** INVALID GEOMETRY TYPE IN START. IGEO=_____ *****

This message from subroutine STRTSU is probably the result of a code error.

K6-110 ***** 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.

K6-111 ** 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. F17.4.3) 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. F17.4.3). The value of *nbas* should be 1 greater than one of these generations.

K6-112 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. F17.4.3). This message can also be caused by a code error.

K6-113 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. F17.4.3). A code error can also cause this message.

K6-114 *** 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. F17.4.3). 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.

K6-115 *** 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. F17.4.3). Remove FAR=YES from the parameter data of the restarted problem or add FAR=NO later in the parameter data. Section F17.5.3 illustrates methods of changing the parameter input data.

K6-116 *** ERROR ***** ERROR *** EXECUTION IS TERMINATED *** ERROR ***
*** ERROR ***

This message from subroutine RDCALC is a companion to messages K6-112 through K6-115. A STOP 121 is executed when this message is printed.

K6-117 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 K6-101 through K6-104 or K6-106 through K6-110. A STOP 130 is executed when this message is printed.

K6-118 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.

K6-119 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. F17.4.3) 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.

K6-120 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 IOs to allow the problem to run.

K6-121 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.

K6-122 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. F17.4.3) to allow computation of the desired number of histories.

K6-123 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. F17.4.3).

K6-124 ***** 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-Ds are specified in the parameter data (X1D=, Sect. F17.4.3), extra 1-D IDs must be entered as data. See Sect. F17.4.9. A STOP 107 is executed when this message is printed.

K6-125 EXCEEDED NEUTRON BANK SIZE

This message from subroutine BANKER indicates that the number of banked particles exceeds the bank size. This situation can be corrected by increasing *nbank* (NBK=, Sect. F17.4.3).

K6-128 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. F17.4.3) is too small, if the biasing or weighting data are incorrect (Sect. F17.4.7), of if the biasing data are incorrectly utilized in the geometry description (Sect. F17.4.4).

K6-129 *** ERROR IN SUBROUTINE ALBEDO *** FACE NUMBER _____ USES _____
ALBEDO NUMBER=_____ INCIDENT XSEC ENERGY GROUP=_____
INCIDENT ALBEDO ENERGY GROUP=_____
INCIDENT ANGLE INDEX=_____
RANDOM NUMBER=_____.

This message from subroutine ALBEDO indicates that a code error was encountered when trying to determine the output energy group during the albedo treatment.

K6-130 *** ERROR IN SUBROUTINE ALBEDO *** 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 ALBEDO indicates that a code error was encountered while trying to compute the returning angle in the albedo treatment.

K6-131 NO FISSIONS

This message from subroutine NSTART indicates that none of the generations encountered a fissile material so no fission points were generated.

K6-132 WARNING....ONLY _____ INDEPENDENT FISSION POINTS WERE GENERATED

This message from subroutine NSTART indicates that less than *npb* (NPG=, Sect. F17.4.3) 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.

K6-133 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. F17.4.3). In this instance, the summaries for the problem cannot be printed.

K6-134 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.

K6-137 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.

K6-138 ********** 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.

K6-139 ***** 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. F17.4.4) 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.

K6-140 ***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.

K6-141 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.

K6-142 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.

K6-143 UNIT ORIENTATION DATA IS REQUIRED IF MORE THAN ONE UNIT TYPE IS SPECIFIED IN THE GEOMETRY DATA.

This self-explanatory message is from subroutine DATAIN. Enter an array i or unit orientation data block as described in Sects. F17.4.5 and F17.5.6. A STOP 136 is executed in conjunction with this message, and a traceback may be printed from subroutine STOP.

K6-144 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 PARTBL. 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.

K6-145 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 PARTBL. 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.

K6-146 *** 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 are 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.

K6-147 *** 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 are 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 if 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.

K6-149 A CROSS SECTION LIBRARY WAS SPECIFIED FOR A RESTARTED PROBLEM, BUT MIXING TABLE DATA WERE 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 IDs 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.

K6-151 *** ERROR *** ____ IS AN INVALID ARRAY TYPE IN THE ARRAY DEFINITION DATA

This message from subroutine ARAYIN indicates that a parameter name was misspelled or the data were out of order. See Sect. F17.4.5 for a list of the array parameter names. A stop 137 is executed in conjunction with this message.

K6-152 *** 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. F17.4.5 and F17.5.6 for assistance.

K6-153 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 these 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 Kbytes, necessary to hold these 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. A STOP 138 is executed in conjunction with this message.

K6-154 *** ERROR *** LEVELA=___ IS LARGER THAN ___ THE ___ NUMBER OF ARRAYS.

This message from subroutine SORTA indicates that the array data specified in the problem are recursively nested. An example of this is the following:

array I contains array 3 array 2 contains array I array 3 contains array 2

Thus the definition of array I and array 3 are intertwined in a never-ending loop. Correct the array data (Sect. F17.4.5) and resubmit the problem. If the input data did not specify recursive nesting, a code error has occurred. A STOP 139 is executed in conjunction with this message.

K6-155 *** 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.

K6-156 *** ERROR *** MIXTURE ___ CONTAINS AT LEAST ONE ZERO VALUE FOR THE TOTAL CROSS SECTION.

This message is from subroutine XSECID. 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.

K6-157 *** ERROR *** THE FIRST HOLE IN A UNIT MUST FOLLOW A VALID GEOMETRY REGION.

This message is from subroutine READGM. If holes are to be utilized in the geometry region data (Sect. F17.4.4), 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 an ARRAY description. Correct the geometry region data and resubmit the problem.

K6-158 *** 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.

K6-159 *** 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 GEOMETRY data (Sect. F17.4.4) did not have the corresponding UNIT ORIENTATION DATA entered in the ARRAY DATA (Sect. F17.4.5). A STOP 142 is executed in conjunction with this message when it is printed from subroutine SORTA. Correct the data and resubmit the problem.

K6-160 THE HOLES ARE RECURSIVELY NESTED.

This message from subroutine HOLE indicates that the geometry region data description (Sect. F17.4.4) specifies holes that are recursively nested. This situation 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 follows:

```
UNIT 1 CUBOID  6PI0.0 BOUNDARY HOLE 2 1  3*0.0
UNIT 2 CUBOID  6PI0.0 BOUNDARY HOLE 1 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.

K6-163 *** 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 situation 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.

K6-164 *** 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.

K6-168 INSUFFICIENT DATA FOLLOWING THE KEYWORD "HOLE".

This self-explanatory message is from subroutine RDARA. Check the data following the keyword "HOLE." Either the data are missing or noninteger data were inadvertently entered. See Sect. F17.4.4.

K6-170 ***** ERROR ***** SECTOR DATA WAS NOT FOUND FOR HOLE ___

This self-explanatory message is from subroutine HOLEIN. Check the sector data following the HOLE media card corresponding to hole number ___. Either the data are missing or noninteger data were inadvertently entered. See Sect. F17.4.4.

K6-171 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. F17.4.11 for information concerning direction cosines down the page.

K6-172 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 message is an error because a zero value vector does not define a direction. See Sect. F17.4.11 for assistance in defining direction cosines across the page.

K6-173 ERRORS WERE DETECTED IN THE INPUT DATA FOR THIS PLOT. IT WILL NOT BE DRAWN.

This message from subroutine RDPLOT is a companion to messages K6-174, K6-180, K6-171 and K6-172. Correct the error that triggered the companion message, and resubmit the problem.

K6-174 ERROR IN KENO PLOT 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. F17.4.11 for assistance.

K6-175 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 PARTBL. 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.

K6-176 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 PARTBL. 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.

K6-177 ***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.

K6-178 *** 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 are 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 if 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.

K6-180 ERROR IN KENO PLOT DATA - KEYWORD _____ IS NOT VALID.

This message from subroutine RDPLOT indicates that the plot or input data are out of order or a keyword is incorrectly spelled. See Sect. F17.4.11 for a list of correct keywords.

K6-181 ***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=l 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.

K6-182 INSUFFICIENT SPACE FOR PLOTTING ARRAYS. FIRST NOS ARE STORAGE NEEDED AND ALLOCATED. LAST IS REQUIRED ADDITIONAL REG. SIZE.

This message from subroutine BOOK5 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 these 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 Kbytes, 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. F17.4.3).

K6-184 ****** _____ 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.

K6-185 *** 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.

K6-186 *** 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.

K6-187 ***** 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, which 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. F17.5.8 for additional assistance.

KS-188 *** ERROR *** HOLE NUMBER _____ REFERENCES UNDEFINED UNIT NUMBER _____.

This message is printed by subroutine HOLE and READGM if the unit number referenced by the hole is less than 1 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 otherwise it is printed by subroutine READGM. Specify a valid unit number (lhole - see GEOMETRY DESCRIPTION, Sect. F17.4.4) and resubmit the problem.

K6-190 ERROR IN PLOT DATA - OPTION _____ IS NOT VALID FOR KEYWORD PIC=.

This message from subroutine RDPLOT indicates an incorrect option associated with the keyword PIC=. See Sect. F17.4.11. Acceptable options include MAT, MIX, MIXT, MEDI, UNT, UNIT, IMP, BIAS, WTS, WGT, WGTS or WEIGH.

K6-191 ***** 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. F17.4.8) is 2, 3, 4, or 5. A global array is required in order to use the specified start type.

K6-192 MATRIX INFORMATION BY ARRAY WAS SPECIFIED AS YES IN THE PARAMETER DATA (MKA=), BUT IS NOT OF USE UNLESS ARRAYS ARE 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.

K6-193 MATRIX INFORMATION BY HOLE WAS SPECIFIED AS YES IN THE PARAMETER DATA (MKH=), BUT IS NOT OF USE UNLESS HOLES ARE 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.

K6-194 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.

K6-195 ***** 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. F17.4.11) or for COM= (unit comment, Sect. F17.4.4, or array comment, Sect. F17.4.5). A STOP 153 is executed when this message is printed.

K6-196 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. F17.4.3) was smaller than the beginning generation number

(BEG=, Sect. F17.4.3). 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.

K6-198 *** 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 K6-32, K6-33, and/or K6-85 may also print as a result of this error.

K6-199 *** 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. A STOP 156 is executed in conjunction with this message. Enter the appropriate array data and resubmit the problem.

K6-200 *** ERROR *** START TYPE 6 WAS SPECIFIED IN THE START DATA, BUT THE STARTING POINTS WERE NOT SPECIFIED.

This message from subroutine RDSTRT indicates that start type 6 was specified but the corresponding starting points were not included in the START data block. Enter the corresponding starting points or change the start type. See Sect. F17.4.8.

K6-201 *** WARNING *** NEUTRON ____ SPECIFIED A POSITION IN THE GLOBAL ARRAY. THE GLOBAL UNIT DID NOT CONTAIN AN ARRAY SO THE POSITION WAS IGNORED.

This warning message from subroutine START6 indicates that extraneous data were specified in the start type 6 data. Verify that the correct global unit is specified.

K6-202 *** ERROR *** ONLY START TYPES 0, 1, OR 6 ARE VALID FOR A PROBLEM WITHOUT AN ARRAY IN THE GLOBAL UNIT.

This message from subroutine START indicates that the specified start type is not valid for the problem. Choose an appropriate start type. A STOP 157 is executed when this message is printed.

K6-204 EXECUTION TERMINATED DUE TO ACHIEVING THE STANDARD DEVIATION SPECIFIED FOR THE PROBLEM.

This message from subroutine GUIDE states that the requested standard deviation has been achieved. If a lower standard deviation is desired decrease the requested standard deviation (SIG=, Sect. F17.4.3).

K6-206 *** ERROR *** THE UNIT SPECIFIED FOR STARTING IS NOT USED IN THE PROBLEM OR IS UNDEFINED.

This message from subroutine START indicates that the unit in which neutrons are to be started is undefined. Verify that the global unit or array is correctly specified. Check the start data for start types 4 and 5 (Sect. F17.4.8) to be sure NBX= is correctly specified. A STOP 158 is executed when this message is printed.

K6-207 *** ERROR *** THE STARTING ARRAY POSITION IS INVALID.

This message from subroutine START indicates that the array position NXS, NYS, or NZS is not valid for start types 3 or 6. NXS, NYS, and NZS must be larger than zero and no larger than NBXMAX, NBYMAX, and NBZMAX of the global array, respectively. Verify that the global unit or array is correctly specified. Correct the start data (Sect. F17.4.8) and resubmit the problem. A STOP 159 is executed when this message is printed.

K6-208 *** WARNING *** THE FRACTION OF NEUTRONS STARTED AS A SPIKE WAS LESS THAN ZERO. IT HAS BEEN RESET TO ZERO.

This message from subroutine START indicates that FCT= was incorrectly specified in the start data for start type 2. The resultant starting distribution is a cosine distribution throughout the volume of a cuboid defined by XSM, XSP, YSM, YSP, ZSM, and ZSP (see Sect. F17.4.8). If a "spike" was desired, set FCT= to a positive number between 0.0 and 1.0. If FCT=0.0 is specified, a cosine distribution without a "spike" is used as the starting distribution. If FCT=1.0 is specified, all the neutrons are started as a "spike" (i.e., they are started uniformly in the unit located at position NXS, NYS, NZS in the global array, as indicated in Sect. F17.4.8).

K6-209 *** WARNING *** THE FRACTION OF NEUTRONS STARTED AS A SPIKE WAS GREATER THAN ONE. IT HAS BEEN RESET TO ONE.

This message from subroutine START indicates that FCT= was incorrectly specified in the start type 2 data. The code reset FCT=1 so all of the neutrons are started as a "spike" (i.e., they are started uniformly in the unit located at NXS, NYS, NZS in the global array, as indicated in Sect. F17.4.8).

K6-210 *** ERROR *** THE UNIT SPECIFIED FOR STARTING IS NOT IN THE GLOBAL ARRAY.

This message from subroutine START indicates that the unit specified by NBX= does not occur in the global array. Verify that the global array is correctly specified and that the unit specified by NBX= is correct (see Sect. F17.4.8).

K6-211 *** ERROR *** THE NUMBER OF SETS OF BIAS FACTORS FROM CARDS 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 indicative of a code error. A STOP 161 is executed when this message is printed.

K6-212 *** ERROR *** THE BIASING DATA SPECIFIED IBGN= __ AND IEND= __. IBGN MUST BE LARGER THAN ZERO AND IEND MUST BE AT LEAST AS LARGE AS IBGN. THE PROBLEM WILL NOT BE RUN.

This message from subroutine RDBIAS indicates an error in the biasing data. The biasing correlation data are order dependent. The order of data entry is: ID=nn ibgn iend, where nn is an ID number from Table F17.4.5 and ibgn is the beginning BIAS ID and iend is the ending BIAS ID (see Sects. F17.4.7 and F17.5.8). In order to continue checking the input data, if ibgn is less than or equal to zero, it is set to 1. Similarly, if iend is less than ibgn, it is set to ibgn.

K6-213 *** ERROR *** INSUFFICIENT SPACE ALLOCATION. MASTER WAS NOT CALLED.

This error message is printed from subroutine KENOVI when KENO-VI is executed in the "stand-alone mode" and from subroutine O0O019 when executed as part of a CSAS analytical sequence. The message indicates that the job control language must be altered to provide more computer memory to be utilized by the problem. A STOP 162 is executed when this message is printed.

K6-214 *** ERROR *** THE BASE OF A TRIANGULAR FACE OF A WEDGE CANNOT BE ZERO.

This self-explanatory message is from subroutine WEDGE. The input data for the wedge following the keyword WEDGE are incorrect. See Sect. F17.4.4.

K6-215 *** ERROR *** THE Y COORDINATE OF A TRIANGULAR FACE OF A WEDGE CANNOT BE ZERO.

This self-explanatory message is from subroutine WEDGE. The input data for the wedge following the keyword WEDGE are incorrect. See Sect. F17.4.4.

K6-216 ***** ERROR ***** XNB= __ IS THE MAXIMUM VALUE THAT CAN BE SPECIFIED IN THE PARAMETER DATA. XNB MUST BE SMALL ENOUGH TO FIT IN THE EXTRA ARRAY OF COMMON NUTRON.

This message from subroutine FLDATA indicates that the value of XNB was set too large. Decrease the size of XMB and retry the problem.

K6-217 *** ERROR *** DIFFERENTIAL ALBEDOS CANNOT BE USED IN AN ADJOINT PROBLEM.

This self-explanatory message is from subroutine FLDATA. Describe reflector material in the mixing table and the geometry instead of using differential albedos or run the problem in the forward mode.

K6-218 ***** ERROR ***** TO START IN A GLOBAL QUADRATIC GEOMETRY, IGEO = __ VALUES FOR XSM, XSP, YSM, YSP, ZSM, AND ZSP MUST BE ENTERED AS START DATA.

This self-explanatory error message is from subroutine STRTSU. A GLOBAL QUADRATIC GEOMETRY implies an infinite media. The boundary of the starting points must be entered. See Sect. F17.4.8.

K6-219 ***ERROR***THE START DATA SPECIFIED __ STARTING POINTS CHOSEN FROM A COSINE DISTRIBUTION BUT NONE WERE FOUND.

This message from subroutine START is printed if start type 2 was specified and the code was unable to start any neutrons in the cuboid defined by XSM, XSP, YSM, YSP, ZSM, ZSP. Verify that fissile material exists within that cuboid. If it doesn't, respecify the starting cuboid to contain fissile material or choose a different start type. If message K6-105 states that only 0 independent starting points were generated, it indicates that the code was unable to start any neutrons in the spike specified by start type 2. Verify that the unit specified for the spike contains fissile material. If only a very small fraction of the volume of this unit is fissile, it may be necessary to input a larger value for the KENO-VI parameter "TMAX=" or choose a different start type. The problem will not be run if message K6-219 is printed.

K6-220 ERROR IN PLOT DATA.
IF THE COORDINATE OF THE LOWER RIGHT-HAND CORNER IS ENTERED, ONE OF THE PLOT PARAMETERS DLX, DLD, NAX, OR NDN MUST BE ENTERED. CURRENT VALUES ARE LISTED BELOW. TITLE:

(Problem title is listed here)

	UPPER-LEFT COORDINATES	UPPER-RIGHT COORDINATES
X	xu	xl
Y	yu	yl
Z	zu	zl

	U AXIS (DOWN)	V AXIS (ACROSS)
X	ud	ua
Y	vd	va
Z	wd	wa

NDN= 0 NAX= 0 DLD= 0.0000E+00 DLX= 0.0000E+00

This message from subroutine RDPLOT indicates that the coordinates of the lower-right-hand corner of the plot were specified in the input data without specifying one of the following plot parameters: (1) NDN, the number of characters down the plot, (2) NAX, the number of characters across the plot, (3) DLD, the vertical spacing between points, or DLX, the horizontal spacing between points. The problem will not be run.

To correct the error, specify NDN, NAX, DLD, or DLX in the plot data and resubmit the problem. See Sect. F17.4.11 and F17.5.9 for assistance.

K6-221 ***** ERROR ***** THE LENGTH OF THE EDGE ALONG THE BASE OF THE X-AXIS FOR A RHOMBOID MUST BE GREATER THAN 0.0. CHECK GEOMETRY WORD ___.

This self-explanatory message is from subroutine RHOMB. Check the data following the keyword RHOMB. See Sect. F17.4.4.

K6-222 ***** ERROR ***** THE ANGLE BETWEEN THE Y-AXIS AND THE Y-EDGE OF THE BASE MUST BE BETWEEN 0 AND 90 DEGREES. CHECK GEOMETRY WORD ___.

This self-explanatory message is from subroutine RHOMB. Check the data following the keyword RHOMB. See Sect. F17.4.4.

K6-223 ***** ERROR ***** THE NUMBER OF SCATTERING ANGLES, SCT, ENTERED IN THE MIXING TABLE MUST NOT EXCEED 13. SCT = ___WAS ENTERED.

This self-explanatory message is from subroutine MIXIT. Adjust the data following the keyword SCT= in the parameter data. See Sect. F17.4.3.

K6-224 *** ERROR *** THE HEIGHT OF A WEDGE CANNOT BE ZERO.

This self-explanatory message is from subroutine WEDGE. The input data for the wedge following the keyword WEDGE are incorrect. See Sect. F17.4.4.

K6-225 ***** ERROR ***** TO START IN A GLOBAL PLANE GEOMETRY, IGEO = ___ VALUES FOR XSM, XSP, YSM, YSP, ZSM, AND ZSP MUST BE ENTERED AS START DATA.

This self-explanatory error message is from subroutine STRTSU. A GLOBAL PLANE GEOMETRY implies an infinite media. The boundary of the starting points must be entered. See Sect. F17.4.8.

K6-226 ***** ERROR ***** XDIST, YDIST, AND ZDIST SPECIFIED FOR A PPIPED MUST BE GREATER THAN 0.0.

This self-explanatory message is from subroutine PPIPED. Check the data following the keyword PPIPED. See Sect. F17.4.4.

K6-227 ***** ERROR ***** PSI, THETA, AND PHI SPECIFIED FOR A PARALLELEPIPED MUST BE GREATER THAN OR EQUAL TO 0.0 AND LESS THAN 90.0.

This self-explanatory message is from subroutine PPIPED. Check the data following the keyword PPIPED. See Sect. F17.4.4.

K6-228 ***** ERROR ***** THE CALCULATION WAS TERMINATED BECAUSE THE SAME SPACE IS DEFINED IN MULTIPLE REGIONS

This message from subroutine GUIDE is printed only if message K6-89 is printed 10 or more times for a given generation. This indicates that the problem defines the same space in multiple regions. If changes have been made in the code, they should be carefully scrutinized. It is probably caused by not correctly specifying the vector definition arrays in the media cards of a unit. It can be caused by not correctly specifying regions that share a common boundary so the boundary between regions is contained in only one region. See Sect. F17.4.4 for more information concerning vector definition arrays on MEDIA cards.

K6-229 ***** ERROR ***** IN UNIT ___ , ___ GEOMETRY CARDS WERE SPECIFIED BUT ___ WAS SPECIFIED IN A SECTOR DEFINITION ARRAY.

This message from subroutine KENOG is printed if a geometry card, larger than the total number of geometry specified in a unit, is referenced in the sector definition array of a media card for that unit.

K6-230 ***** ERROR ***** THE CALCULATION WAS TERMINATED BECAUSE A GEOMETRY WORD IN UNIT ____ PRODUCES AN IMAGINARY BOUNDARY.

This message from subroutine PNTCHK should only be produced by the geometry word QUADRATIC when used as part of an array boundary. The boundary produces imaginary results. Check the coefficients of the geometry word.

K6-231 ***** ERROR ***** THE CALCULATION WAS TERMINATED BECAUSE ARRAY ____ IS NOT PROPERLY POSITIONED IN REGION ____ OF UNIT ____.
THE ARRAY BOUNDARY IS OUTSIDE THE _____ SIDE BY ____ .

This message from subroutine PNTCHK is printed only if an array is placed in a region such that there is space in the region not occupied by the array. Reposition your array.

K6-232 ***** ERROR ***** IN UNIT _____ MORE THAN ONE GEOMETRY RECORD IS SPECIFIED AS NUMBER _____.

This error message from subroutine READGM means more than one geometry record in the specified unit was given the same label.

K6-233 ***** ERROR ***** A VECTOR DEFINITION ARRAY IN UNIT _____ REFERENCES GEOMETRY RECORD ____ WHICH HAS NOT BEEN SPECIFIED.

This error message from subroutine KLANGA means a geometry record was specified in a vector definition array that has not been defined in the specified unit.

K6-234 ***** ERROR ***** IN UNIT _____ THE USE OF GEOMETRY LABEL _____ IN THE BOUNDARY DEFINITION VECTOR CONFLICTS WITH VECTOR DEFINITION GEOMETRY LABEL _____.

This error message from subroutine KLANGA means a geometry label used in the unit boundary definition vector has the opposite sign of the same geometry label used in a media, array, or hole definition vector in the same unit. The geometry labels used in the unit boundary definition vector do not need to be repeated in the other definition vectors in the unit. If they are repeated they need to have the same sign as those used in the boundary definition vector.

K6-235 ***** ERROR ***** BOUNDARY SECTOR DATA WAS NOT BOUND FOR UNIT _____.

This error message from subroutine BNDRY means a boundary record is present in the specified unit but does not contain sector data.

K6-236 ***** ERROR ***** THE UNIT _____ BOUNDARY RECORD CONTAINS MORE THAN ONE POSITIVE GEOMETRY RECORD LABEL.

This error message from subroutine JOMCHK is printed when the boundary record of a unit used in an array contains more than one positive geometry record label. When a unit is used in an array, the boundary record must contain only one positive geometry record label that is consistent with the type of array it is used in.

K6-237 ***** ERROR ***** THE GLOBAL UNIT BOUNDARY IS NOT DEFINED BY A SINGLE CUBOID BUT CONTAINS MULTIPLE ALBEDOS.

This error message from subroutine JOMCHK indicates more than one albedo type was specified for the problem but a single cuboid was not used to define the global unit boundary. Multiple albedo types are only allowed if a single cuboid is used to define the global unit boundary.

K6-238 ***** ERROR ***** THE GLOBAL UNIT BOUNDARY USES MULTIPLE GEOMETRY WORDS BUT NOT THE VOID ALBEDO.

This error message from subroutine JOMCHK indicates an albedo other than void was specified with a global unit boundary definition vector containing more than one geometry record label. Multiple geometry record labels are allowed in the global unit boundary definition vector only for the void albedo on all surfaces. Void is the default boundary condition.

K6-239 ***** ERROR ***** ALBEDO OTHER THAN VOID WHITE OR REFLECTED USED WITH NONPAIRED PLANE BOUNDARY.

This error message from subroutine JOMCHK indicates an albedo other than void, white, or mirror is specified but the global unit boundary definition vector is specified by a nonpaired plane geometry record. Only void, mirror, or white albedo conditions may be used when the global boundary uses nonpaired planes.

K6-240 *****ERROR***** GEOMETRY WORD "UNIT" IS REQUIRED BUT _____ WAS READ INSTEAD.

This error message is from subroutine KENOG. Prior to specifying geometry or media data in the GEOMETRY DATA BLOCK, a unit must be specified.

K6-241 *****ERROR*****UNDEFINED SPACE IN THE REGION CONTAINING HOLE _____.

This error message from subroutine TRACK is caused by a unit improperly being placed in a hole, leaving a gap between crossing into the hole and entering the unit in the hole. Check the hole geometry.

K6-242 ***ERROR***NEUTRON _____ IN GENERATION ____ HAS LEAKED FROM UNIT____ BUT IS STILL WITHIN THE VOLUME DEFINED BY THE UNIT BOUNDARY RECORD.

This error message from subroutine TRACK indicates that the specified particle is inside the volume defined by the units boundary record but is not in any region. Check the media record descriptions for the specified unit. Also, specifying a screen plot of the problem may help identify the problem.

K6-243 ***ERROR***IN UNIT____ THE BOUNDARY RECORD CONTAINS LABEL ____ WHICH DOES NOT CORRESPOND TO A GEOMETRY RECORD.

This error message from subroutine KLANGA indicates that in the specified unit the boundary record contains a geometry record label that is not linked to a geometry record in the unit.

F17.7.2 STOP CODES

The STOP codes that are encountered in KENO V.a 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, as shown in the following:

STOP NO.	SUBROUTINE	TRACEBACK	ASSOCIATED MESSAGE
20			See Sect. F17.7.3
100	ARAYIN	Yes	K6-25
101	ARAYIN	No	K6-31
105	WRTALB	No	K6-71
106	ICEMIX	No	K6-68
107	INDX	No	K6-124
108	INITAL	No	K6-2
109	INITAL	No	K6-2
110	MAKANG	Yes	K6-57
111	MIXCRS	Yes	K6-52
112	MIXER	Yes	K6-53
113	MIXER	Yes	K6-54
114	MIXIT	Yes	K6-15
115	NORM1D	No	K6-56
116	NSUPG	Yes	K6-79
117	NSUPG	No	K6-81
118	PARAM	No	K6-7 or K6-8
120	RDALB	No	K6-49
121	RDCALC	No	K6-116 or K6-146 or K6-147 or K6-177 or K6-178
122	RDICE	No	K6-50
123	RDRST	No	K6-48
124	RDTAPE	No	K6-69
125	READGM	No	K6-20
126	SAVST6	Yes	K6-140
128	START	Yes	K6-101
129	BOOK5	No	K6-100
130	BOOK1	No	K6-117 or K6-128 or K6-228
131	VOLFIS	No	K6-110
132	WRTICE	No	K6-72
133	WRTRST	No	K6-70
135	DATAIN	Yes	K6-142
136	DATAIN	Yes	K6-143
137	ARAYIN	No	K6-151
138	SORTA	Yes	K6-153
139	SORTA	No	K6-154
140	LODARA	No	K6-155
142	SORTA	No	K6-159
143	HOLE	No	K6-160
147	BOOK5	Yes	K6-182
148	WAITIN	No	K6-185

STOP NO.	SUBROUTINE	TRACEBACK	ASSOCIATED MESSAGE
150	WATES	Yes	K6-22
151	RDBIAS	No	K6-184
152	DATAIN	No	K6-46
153	RCHRS	Yes	K6-195
154	RDCALC	No	K6-196
155	GEOMIN	Yes	K6-25
156	SORTA	Yes	K6-199
157	START	No	K6-202
158	START	No	K6-206
159	START	No	K6-207
160	START	No	K6-210
161	WAITIN	No	K6-211
162	KENO-VI, O00019	Yes	K6-213
164	TRACK	No	K6-241

F17.7.3 MESSAGES ASSOCIATED WITH STOP 20 IN KENO-VI

The error messages that are associated with a STOP 20 in KENO-VI are listed below in numerical order. Look up the appropriate message number to determine corrective measures.

LMP001 DA ERROR - INVALID UNIT NUMBER.
THE LOGICAL UNIT NUMBER IS ____.

This message from the subroutine library direct-access routines indicates that an invalid unit number was specified as a direct-access device. In KENO-VI, this message is indicative of a code error.

LMP002 DA ERROR - FORTRAN USING THIS UNIT.
THE LOGICAL UNIT NUMBER IS ____.

This message from the subroutine library direct-access routines indicates that the specified unit number is open as a sequential dataset rather than a direct-access dataset. In KENO-VI, this error may be caused by entering a direct-access unit number for LIB= or XSC=.

LMP003 DA ERROR - DCB NOT OPEN.
THE LOGICAL UNIT NUMBER IS ____.

This message from the subroutine library direct-access routines indicates that the program attempted to read or write on a direct-access device, but the data control block was not open. In KENO-VI this message is indicative of a code error.

LMP004 DA ERROR - UNABLE TO OPEN DCB.
THE LOGICAL UNIT NUMBER IS _____.

This message from the subroutine library direct-access routines indicates that the program is unable to open the data control block for the direct-access device. This message indicates that the job control language did not include proper specification of the above named unit.

LMP005 DA ERROR - RELATIVE BLOCK NOT IN DATA SET.
RELATIVE BLOCK NUMBER IS _____.

This message from the subroutine library direct-access routines indicates that the number of direct-access blocks is too small for the problem. Increase the number of direct access blocks in the KENO-VI parameter data by entering the parameter NB8=nnn, where nnn is larger. For example, if the relative block number is 201, try increasing the number of direct-access blocks to 300 or more.

LMP006 DA ERROR - INVALID BLOCK LENGTH.
THE BLOCK LENGTH IS _____.

This message from the subroutine library direct-access routines indicates that the length of the direct-access blocks is invalid. A valid block length must be positive. This message is indicative of a code error.

LMP007 DA ERROR - DCB ALREADY OPEN.
THE LOGICAL UNIT NUMBER IS _____.

This message from the subroutine library direct-access routines indicates that the data control block for the above named unit was previously opened and not closed.

LMP008 DA ERROR - PERMANENT I/O ERROR.

This message from the subroutine library direct-access routines indicates that a permanent IO error has occurred.

F17.A ALPHABETICAL INDEX OF SUBROUTINES

This section provides a convenient alphabetical index of the subroutines used in KENO-VI, the subroutines that call it, and the subroutines it calls.

Subroutine Name	Calling Subroutine	Called Subroutine
albedo	track	dazirn fltrn
albrd	difalb	albuse
albuse	albrd	inquir io rite
angles	prang	q
aralba	jomity	prtlba
arasiz	corsiz	lscan
arayin	datain	aread box clear iread rchrs rdbox stop yread
array	kenog	aread dread iread lcompr lread
arscan	sorta	
badmom	prang	
banker	guide	move
birite	param	
bndry	kenog	iread lread
book1	master	fitflx freak jstime kedit

Subroutine Name	Calling Subroutine	Called Subroutine
book2	master	datain iosdun openda
book3	master	corre icemix iosdun mixer nsupg point wrtrst
book4	master	jomity lodwts
book5	master	clear guide iosdun jstime prtplt stop
box	arayin	
boxc	loadit	
chkstr	guide	move trkwrt
choose	start4 start5	fltrn
chord	kenog	aread clear dread lcompr
clrnit	rdplot	
cmprs	mixmix	
cone	kenog	clear dread
corre	book3	clear inquir ratio reed rite

Subroutine Name	Calling Subroutine	Called Subroutine
corsiz	jomity	arasiz locbox
cross	track	
crsara	track	
crsprd	ppiped rhomb	
cuboid	kenog	clear dread
cylndr	kenog	clear dread
datain	book2	arayin aread clear extra fldata geomin idx1d iowrt mixit rdbias rdplot rdref rdrst rdstrt rt rtara savst6 stop wrtplt
deviat	kedit pltkef	
difalb	fldata	albrd
ditto	kenog	
dodeca	kenog	clear dread
dotprd	track	
ecyl	kenog	clear dread

Subroutine Name	Calling Subroutine	Called Subroutine
editor	kedit	
ellips	kenog	clear dread
extra	datain	
fil2d	fillsg	rd
fillgm	geomin	hrtrns
fillsg	nsupg	clear fil2d inquir iset rd reed rite rt sgalb sgwt
find	getmus	q
findbx	locate track	
fisflx	guide	jstime looper matk statis
fitflx	book1	ioleft prtflx rd
fldata	datain	difalb reed rgused rite sorta sortr wates
freak	book1	erf

Subroutine Name	Calling Subroutine	Called Subroutine
fsabed	kedit	
geomin	datain	clear fillgm kenog klanga readgm stop
getmus	prang	find q
gocurs	volume	hunter
gtvols	volume	
guide	book5	banker chkstr clear fisflx indx inquir ioleft jstime nstart pull rd rdcalc reed reset rite start track wrtcal
hole	lodara sorta	clear stop
holein	kenog	iread lread
hopper	kenog	clear dread
hrotrn	locate track	move
hrtrns	fillgm	

Subroutine Name	Calling Subroutine	Called Subroutine
hunter	gocurs	clear
	volume	move
hxprsm	kenog	clear
		dread
icemix	book3	rdtape
idx1d	datain	iread
indx	guide	
inital		ioleft
		ionums
		jstime
		mesage
		opnfil
		param
		partbl
		scanon
		setbin
iosdun	book2	ioleft
	book3	
	book5	
	jomity	
ixalb	limln	
	nsupg	
jll2	mixer	
jomchk	jomity	pntchk
jomity	book4	aralba
		corsiz
		iosdun
		jomchk
		loadit
		prtjom
		volume

Subroutine Name	Calling Subroutine	Called Subroutine
kedit	book1	clear
		deviat
		editor
		fsabed
		jstime
		looper
		matrix
		pltkef
		rndout
kenog	geomin	aread
		array
		bndry
		chord
		clear
		cone
		cuboid
		cylndr
		ditto
		dodeca
		ecyl
		ellips
		holein
		hopper
		hxprsm
		iread
		lcompr
		media
		origin
		plane
		ppiped
		quad
		rchrs
		rhomb
		rotate
		rotrns
		sphere
		wedge
klanga	geomin	clear
		prtsec
ldwrt	track	
legend	prang	
libeql	param	

Subroutine Name	Calling Subroutine	Called Subroutine
limln	nsupg	ixalb
loadit	jomity	boxc lodalb lodara prtara reed
locate	mesh start	findbx hrotrn move posit
locbox	corsiz lodara sorta start tkarry track	
lodalb	loadit	reed
lodara	loadit	clear hole locbox reed
lodrgc	rgused	
lodwts	book4	clear prtwts rd reed
looper	fisflx kedit	reed rite
lscan	arasiz rgused start	
makang	mixer	clear prang reed stop

Subroutine Name	Calling Subroutine	Called Subroutine
maktap	mixer	io reed scoot
master		book1 book2 book3 book4 book5
matk	fisflx matrix	clear
matrix	kedit	jstime labl matk prtsrc
media	kenog	iread lread
mesh	print	clear locate
mix1d	mixmix	
mix2d	mixmix	mgcwrd
mix2m	mixmix	
mixcrs	mixer	clear stop
mixer	book3	closda jll2 makang maktap mixcrs mixmix openda reed rite stop xlnths

Subroutine Name	Calling Subroutine	Called Subroutine
mixit	datain	aread fread inquir iread lread rdmixt rite stop
mixmix	mixer	cmprs io mix1d mix2d mix2m nnitl norm1d norm2d prtmix reed rite sumsct
nnitl	mixmix	clear inquir rite
norm1d	mixmix	
norm2d	mixmix	
normal	track	
nstart	guide	clear fltrn move sortbk
nsupg	book3	fillsg freecr inquir ixalb limln openda point pratbl prtxs reed rite stop

Subroutine Name	Calling Subroutine	Called Subroutine
origin	kenog	aread dread lcompr
param	inital	aread birite clear fread iread libeql opnfil rndin rndout timfac zread
partbl	inital	rndout
period	track	
plane	kenog	aread clear dread lcompr
pltkef	kedit	deviat
pntchk	jomchk	
point	book3 nsupg	
posit	locate	
ppiped	kenog	clear crsprd dread
prang	makang	angles badmom getmus inquir legend rd rite
pratbl	nsupg	rtadj

Subroutine Name	Calling Subroutine	Called Subroutine
print	prtplt	filnam mesh system
prt1ds	prtxs	
prt2ds	prtxs	
prtara	loadit	
prtflx	fitflx	
prtjom	jomity	
prtlba	aralba	
prtmix	mixmix	sortmx
prtplt	book5	clear inquir jstime print reed relate untcrs
prtpos	track	
prtsec	klanga	
prtsrc	matrix	
prtwts	lodwts	
prtxs	nsupg	prt1d prt1ds prt2ds reed
q	angles find getmus	
quad	kenog	aread clear dread lcompr vexcav

Subroutine Name	Calling Subroutine	Called Subroutine
ratio	corre	
rchrs	arayin	aread
	kenog	getptr
	rdplot	rcrdln
		rstptr
		stop
rcolor	rdplot	aread
		iread
rdalb	rdrst	clear
		inquir
		io
		rite
rdara	rdrst	inquir
		io
		rite
rdbias	datain	aread
		fread
		io
		iread
		waitin
rdbox	arayin	iread
rdcalc	guide	io
		rdgrp
		rite
		rndin
		shufl
		stop
rdgrp	rdcalc	
rdice	rdrst	clear
		inquir
		io
		rite
rdmixt	mixit	

Subroutine Name	Calling Subroutine	Called Subroutine
rdplot	datain	aread clear clrnit fread io iread rchrs rcolor
rdref	datain	aread cread
rdrst	datain	inquir io rdalb rdara rdice rdwts rite
rdstrt	datain	aread clear fread io iread opnfil
rdtape	icemix	clear inquir io rite xsec1d xtenda
rdwts	rdrst	inquir io rite
readgm	geomin	clear rothol
relate	prtplt	clear
reset	guide	move
rgused	fldata	clear lodrgc lscan

Subroutine Name	Calling Subroutine	Called Subroutine
rhomb	kenog	clear crsprd dread
rotate	kenog	aread dread lcompr
rothol	readgm	
rotrns	kenog	
rt	datain fillsg	inquir rite
rtadj	pratbl	
rtara	datain	clear inquir rite
savst6	datain	clear inquir io move rite stop
scoot	maktap	
sgalb	fillsg	inquir rd reed
sgwt	fillsg	rd
shufl	rdcalc	
sorta	fldata	arscan clear hole locbox reed stop
sortbk	nstart	dgtiso move

Subroutine Name	Calling Subroutine	Called Subroutine
sortmx	prtmix	
sortr	fldata	
sphere	kenog	clear dread
start	guide	clear dgtiso fltrn jstime locate locbox lscan move start0 start1 start2 start3 start4 start5 start6 stop strtsu volfis
start0	start	fltrn
start1	start start2	fltrn
start2	start	start1 start5
start3	start	
start4	start	choose
start5	start start2	choose fltrn
start6	start	fltrn move reed
statis	fisflx	

Subroutine Name	Calling Subroutine	Called Subroutine
strtsu	start	dazirn dgtiso fltrn
sumsct	mixmix	
tkarry	track	locbox move
track	guide	albedo cross crsara dazirn dgtiso dotprd exprn findbx fltrn hrotrn ldwrt locbox move normal period prtpos sflra tkarry trkwrt
trkwrt	chkstr track	rndout
untcrs	prtplt	clear
vexcav	quad	
volfis	start	
volume	jomity	clear gocurs gtvols hunter move
waitin	rdbias	inquir io rite

Subroutine Name	Calling Subroutine	Called Subroutine
wates	fldata	inquir io move reed rite stop
wedge	kenog	clear dread
wrtalb	wrtrst	inquir io reed
wrtara	wrtrst	io reed
wrtcal	guide	io reed rndout wrtgrp
wrtgrp	wrtcal	
wrtice	wrtrst	inquir io reed
wrtplt	datain	inquir io rite
wrtrst	book3	inquir io reed rndout wrtalb wrtara wrtice wrtwts
wrtwts	wrtrst	io reed
xlnths	mixer	
xsec1d	rdtape	
xxlim		

F17.B ALPHABETICAL INDEX OF COMMONS

This section lists the labeled common blocks used in KENO-VI and an alphabetical listing of the subroutines that reference them.

Common Name	Referencing Subroutine
albdat	albedo
	albrd
	albuse
	book3
	corre
	datain
	difalb
	fillsg
	fldata
	guide
	jomchk
	kedit
	limln
	loadit
	nsupg
	param
	point
	pratbl
	rdalb
	rdref
	rdrst
	sgalb
	track
	wrtalb
	wrtrst
albnam	albedo
	albrd
	albuse
	book3
	corre
	datain
	difalb
	fillsg
	fldata
	guide
	jomchk
	kedit
	limln
	loadit
	nsupg
	param

Common Name	Referencing Subroutine
	point
	pratbl
	rdalb
	rdref
	rdrst
	sgalb
	track
	wrtalb
	wrtrst
angle	angles
	badmom
	find
	getmus
	legend
	makang
	prang
blklnc	book2
	fitflx
	guide
	nsupg
	param
	partbl
	rdtape
dimen	arasiz
	arayin
	book1
	book2
	book3
	book4
	book5
	box
	boxc
	corre
	corsiz
	cross
	crsara
	datain
	editor
	fil2d
	fillgm
	fillsg

Common Name	Referencing Subroutine
	fisflx
	fitflx
	fldata
	geomin
	guide
	hrotrn
	icemix
	inital
	jomchk
	jomity
	kedit
	kenog
	limln
	locate
	locbox
	lodara
	looper
	matrix
	mesh
	mixer
	mixit
	nstart
	nsupg
	param
	partbl
	point
	pratbl
	print
	prtjom
	prtplt
	rdbias
	rdbox
	rdcalc
	rdgrp
	rdice
	rdrst
	rdstrt
	rdtape
	readgm
	reset
	rgused
	rtara
	sorta

Common Name	Referencing Subroutine
	sortr
	start
	statis
	tkarry
	track
	volfis
	volume
	wrtcal
	wrtgrp
	wrtice
	wrtrst
	xsec1d
drtacs	albuse
	book2
	book3
	corre
	datain
	fil2d
	fillsg
	fldata
	guide
	inital
	loadit
	lodara
	lodwts
	mixer
	mixit
	nsupg
	prang
	prtplt
	rdalb
	rdara
	rdice
	rdrst
	rdtape
	rdwts
	rt
	rtara
	savst6
	sgalb
	sgwt
	sorta
	start6

Common Name	Referencing Subroutine
	waitin
	wates
	wrtalb
	wrtice
	wrtplt
	wrtrst
	wrtwts
errflg	arayin
	inital
	prtlba
final	book1
	book2
	book5
	fisflx
	guide
	inital
	kedit
	matrix
	prtplt
	start
	statis
	tkarry
	track
lifetm	book1
	editor
	fisflx
	guide
	kedit
	rdcalc
	statis
	track
	wrtcal

Common Name	Referencing Subroutine
logic	albrd
	arasiz
	arayin
	book1
	book2
	book3
	book4
	book5
	chkstr
	corre
	corsiz
	datain
	editor
	fil2d
	fillgm
	fillsg
	fisflx
	fitflx
	fldata
	geomin
	guide
	icemix
	idx1d
	inital
	jomchk
	jomity
	kedit
	kenog
	locate
	looper
	matrix
	mesh
	mixer
	nsupg
	param
	partbl
	point
	posit
	prtjom
	prtplt
	prtxs
	ratio
	rdbias

Common Name	Referencing Subroutine
	rdbox
	rdcalc
	rdgrp
	rdref
	rdrst
	rdtape
	readgm
	reset
	rtara
	sorta
	sortr
	start
	statis
	tkarry
	track
	trkwrt
	volfis
	volume
	wates
	wrtcal
	wrtgrp
	wrtrst
lowbnd	albedo
	corsiz
	fillsg
	fisflx
	gocurs
	guide
	hrotrn
	hunter
	jomchk
	kedit
	locate
	lodara
	mesh
	point
	print
	prtjom
	prtplt
	reset
	start
	tkarry
	track
	volume

Common Name	Referencing Subroutine
lpnt	book2
	book3
	book4
	book5
	start6
matrx	fldata
	matrix
	pratbl
	reset
	sorta
	track
mstr	book1
	book2
	book3
	book4
	book5
nutron	albedo
	chkstr
	choose
	findbx
	fldata
	guide
	hrotrn
	locate
	mesh
	nstart
	reset
	shufl
	start
	start0
	start1
	start2
	start3
	start4
	start5
	start6
	strtsu
	tkarry
	track
	trkwrt

Common Name	Referencing Subroutine
pcolor	clrnit
	mesh
	print
	prtplt
	rdplot
	rdrst
	relate
	wrtrst
pict	clrnit
	mesh
	print
	prtplt
	rdplot
	rdrst
	relate
	wrtrst
picttl	clrnit
	mesh
	print
	prtplt
	rdplot
	rdrst
	relate
	wrtrst
pointr	book1
	book3
	book4
	book5
	datain
	fillsg
	icemix
	jomity
	loadit
	nsupg
	point
	rdcalc
	wrtcal

Common Name	Referencing Subroutine
rtrn	cone
	cuboid
	cylndr
	ditto
	dodeca
	ecyl
	ellips
	fillgm
	holein
	hopper
	hrtrns
	hxprsm
	kenog
	origin
	plane
	ppiped
	quad
	readgm
	rhomb
	rotate
	rothol
	rotrns
	sphere
	wedge
runtyp	birite
	datain
	inital
	param
	partbl
stdata	chkstr
	datain
	rdrst
	rdstrt
	start
	start0
	start1
	start2
	start3
	start4
	start5
	start6
	volfis
	wrtrst

Common Name	Referencing Subroutine
titl	datain
	editor
	fitflx
	freak
	guide
	inital
	kedit
	kenog
	matrix
	nsupg
	param
	partbl
	point
	pratbl
	prtjom
	prtlba
	prtmix
	prtwts
	prtxs
	rdbox
	rdplot
	rdtape
	volume
	wrtrst
unit	albedo
	albrd
	albuse
	arasiz
	arayin
	array
	bndry
	book1
	book2
	book3
	book4
	book5
	box
	chkstr
	chord
	cone
	corre
	corsiz

Common Name	Referencing Subroutine
	cross
	crsara
	cuboid
	cylndr
	datain
	difalb
	ditto
	dodeca
	ecyl
	editor
	ellips
	fil2d
	fillgm
	fillsg
	fisflx
	fitflx
	fldata
	freak
	geomin
	guide
	holein
	hopper
	hrotrn
	hxprsm
	icemix
	idx1d
	indx
	inital
	jomchk
	jomity
	kedit
	kenog
	ldwrt
	limln
	loadit
	lodara
	lodwts
	looper
	makang
	maktap
	matk
	matrix
	media

Common Name	Referencing Subroutine
	mixer
	mixit
	mixmix
	nnitl
	nstart
	nsupg
	origin
	param
	partbl
	plane
	pntchk
	point
	posit
	ppiped
	prang
	print
	prtara
	prtflx
	prtjom
	prtlba
	prtplt
	prtpos
	prtsrc
	prtxs
	quad
	ratio
	rdalb
	rdbias
	rdbox
	rdcalc
	rdgrp
	rdice
	rdplot
	rdref
	rdrst
	rdstrt
	rdtape
	rdwts
	readgm
	rhomb
	rotate
	rothol
	rt
	rtara

Common Name	Referencing Subroutine
	savst6
	sgalb
	sgwt
	shufl
	sorta
	sortr
	sphere
	start
	start2
	start6
	statis
	strtsu
	tkarry
	track
	trkwrt
	vexcav
	volfis
	volume
	waitin
	wates
	wedge
	wrtalb
	wrtcal
	wrtgrp
	wrtice
	wrtrst
	wrtwts

F17.C KENO-VI INPUT SUMMARY

This appendix consists of a summary of the KENO-VI input data requirements.

Table F17.C.1 Summary of parameter data

TITLE: The title must be entered first (80 characters, including blanks) See Sect. F17.4.3

PARAMETERS: Format: READ PARAM enter parameter data here END PARAM
If parameters are entered, they must follow the title. See Sects. F17.4.3, F17.5.2, and F17.5.3.

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=	79	albedo
TBA=	0.5 min	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	LTB=	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. prod. by array	WRS=	0	write restart
NPG=	300	no. per generation	XS2=	NO	2-D xsecs	HAL=	NO	MKA at highest level			
NSK=	3	generations skipped	PKI=	NO	fission spectrum	PLT=	YES	printer plots			
RES=	0	gens. between restart	P1D=	NO	extra 1-D xsecs	BUG=	NO	debug print			
NBK=	NPG+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=	1000000	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						
NG8=	200	blocks for d.a. unit									
NL8=	512	length of d.a. block									
SIG=	given	standard deviation to terminate problem									

Table F17.C.2 Summary of array data

ARRAY Format: READ ARRAY array parameters data type orientation data END ARRAY See Sects. F17.5.5, F17.5.6, and F17.5.7.

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
TYP=	cuboidal	array type (cuboidal or triangular)
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=	none	global, array number*
COM=	none	delim comment delim optional comment is a maximum of 132 characters

*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. Orientation data for FILL are terminated by entering 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	N	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 the end points, j and k, with i entries linearly interpolated between them (i.e., a total of i+2 points). At least one blank must separate j and k. When used for an integer array, the I option should only be used to generate integer steps (i.e., (k-j)/(i+1) should be a whole number).
	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. Orientation data for LOOP are terminated by entering END LOOP.

ENTER DATA IN THE FORM:

DATA ENTRY	COMMENTS
LTYPE	The unit or box type. LTYPE must be greater than 0.
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 int he 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.
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 F17.C.3 Summary of biasing data

BIAS (weighting)

Format: READ BIAS keyword correlation data auxiliary data END BIAS
See Sects. F17.4.7 and F17.5.8

KEYWORD	DESCRIPTION		MATERIAL	ID	ENERGY GROUPS	THICKNESS/ INCREMENT
ID=	CORRELATION DATA will be read next.					
	id	material ID. Enter from table at right to use weighting data from the library	concrete	301	16,27,123	5 cm
			paraffin	400	16,27,123	3 cm
	ibgn	beginning bias ID	water	500	16,27,123,218	3 cm
	iend	ending bias ID	graphite	6100	16,27,123	20 cm
WT=	AUXILIARY DATA will be read next.					
WTS=	AUXILIARY DATA will be read next.					
	wttitl	material title (12-character maximum)				
	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 x ngpwt values of wtavg				

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=. When AUXILIARY DATA are entered, CORRELATION DATA must also be entered to use the data.

Beginning and ending bias IDs are defined by the user. The geometry specificaiton that has the bias ID equal to the beginning bias ID utilizes the wt av from the first interval of material ID.

Table F17.C.4 Summary of boundary condition data

BNDS
(albedo or boundary conditions)

Format: READ BNDS face code albedo name END BNDS
See Sect. F17.4.7

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	+XY=	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

Any albedo conditin on any face is allowed if the global unit boundary record consists of only one geometry record label that references a cuboid
Any one albedo condition on all surfaces is allowed if the global unit boundary record contains only one geometry record label that references a shape composed of paired planes
Either a void, mirror, or white boundary condition on all surfaces is allowed if the global unit boundary record contains only one geometry record label.
If the global unit boundary record contains more than one entry only the void boundary condition is allowed. Void is the default

ALBEDO NAMES AVAILABLE ON THE KENO V.a ALBEDO LIBRARY, FOR USE WITH THE FACE CODES

ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION	ALBEDO NAME	DESCRIPTION
DP0H2O	12-in. double P0 water	CONC-4	4-in. concrete differential	VACUUM	vacuum condition
DP0H2O	differential albedo with	CON4	albedo with 4 incident angles	VOID	
DP0	4 incident angles	CONC4		VACU	
DP0				VAC	
		CONC-8	8-in. concrete differential		
H2O	12-in. water differential	CON8	albedo with 4 incident angles	SPECULAR	mirror image reflection
WATER	albedo with 4 incident angles	CONC8		MIRROR	
				MIRR	
PARAFFIN	12-in. paraffin differential	CONC-12	12-in. concrete differential	SPEC	
PARA	albedo with 4 incident angles	CON12	albedo with 4 incident angles	SPE	
WAX		CONC12		MIR	
CARBON	200-cm carbon differential	CONC-15	16-in. concrete differential	PERIODIC	periodic boundary condition
GRAPHITE	albedo with 4 incident angles	CON16	albedo with 4 incident angles	PERI	
C		CONC16		PER	
ETHYLENE	12-in. polyethylene	CONC-24	24-in. concrete differential		
POLY	differential albedo with	CON24	albedo with 4 incident angles		
CH2	4 incident angles	CONC24			

Table F17.C.5 Summary of geometry data

GEOMETRY (region)

Format: READ GEOM geometry region data END GEOM
See Sects. F17.4.4, F17.5.1.2, F17.5.6, and F17.5.7.

GEOMETRY REGION DATA consist of SIMPLE GEOMETRY REGION DATA and EXTENDED GEOMETRY REGION DATA.

ENTER GEOMETRY REGION DATA IN THE FOLLOWING FORM:

GLOBAL SPECIFICATION
UNIT n
OPTIONAL GEOMETRY COMMENT
GEOMETRY, MODIFICATION, MEDIA, and BOUNDARY DATA
* *

ENTER DATA IN THE FOLLOWING FORM:
GLOBAL Enter once 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.
Geometry Keyword Label Input Parameters Optional origin data (ORIGIN) Optional chord data (CHORD) Optional rotation data (ROTATE degrees)
Media Keyword Input Parameters
Boundary Keyword Inpput Parameters Each unit must have one boundary record.
Enter as many geometry and media descriptions as necessary to describe the unit and as many units as necessary to describe the system.

GEOMETRY REGION INPUT DATA REQUIREMENTS

TYPE OF DATA	KEYWORD	INPUT PARAMETERS	NOTES
GEOMETRY	CONE	Rt Zt Rb Zb	
	CUBOID	+X -X +Y -Y +Z -Z	
	CYLINDER	R Zt Zb	
	DODECAHEDRON	R	
	ECYLINDER	Rx Ry Zt Zb	
	ELLIPSOID	Rx Ry Rz	
	HEXPRISM	R Zt Zb	
	HOPPER	Lxt Lyt Zt Lxb Lyb Zb	
	PARALLELPIPED	XDIST YDIST ZDIST PSI THETA PHI	
	PLANE*	XPL=a YPL=b ZPL=c CON=d	Data are entered after the = following the respective subordinate keyword. Only nonzero data are required.
	QUADRATIC*	AQU=a BQU=b CQU=c DQU=d EQU=e FQU=f GQU=g HQU=h IQU=i JQU=j	Date are entered after the = following the respective subordiante keyword. Only nonzero data are required.
	RHOMBOID	DX PSI	
	SPHERE	R	
	WEDGE	XBASE XPT YPT ZLNG	
MODIFICATION[a]	CHORD	+X= -X= +Y= -Y= +Z= -Z=	Only nonzero data are required.
	ORIGIN	X= Y= Z=	Only nonzero data are required.
	ROTATE	A1= A2= A3=	The body is rotated about the Origin using the Euler angle x-convention. Only nonzer data are required.
MEDIA[b]	MEDIA	Mid Bid i1 i2...	
	HOLE	Hid i1 i2...	The region boundary containing the hole must exactly match the boundary of the unit being placed in the hole.
	ARRAY	Aid i1 i2... PLACE nx ny nz xx yy zz	Place positions the array by placing the origin of the unit in array position nx ny nz at xx yy zz. The region boundary containing the array must be enclosed in or on the array boundary.
BOUDARY	BOUNDARY	i1, i2, ...	Defines the overal volume of the unit.

[a]Modification Data can be entered in any order. Any combination of modification data can follow any geometry record.
[b]The Region Definition Vector, i1 i2 . . ., should contain references to all geometry records which share a boundary with the region being defined.

Table F17.C.6 Summary of mixing table data

MIXTURES

Format: READ MIXT xsec parameters END MIXT
These data are entered only if an AMPX working-format library is being used (LIB=) in the parameter data, Sect. F17.4.3. Do not enter if an ICE mixed library is used, (XSC=) in the parameter data, Sects. F17.4.10 and F17.5.5.

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. K6-60

MIXING TABLE DATA consist 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 F17.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 Sects. F17.4.1 and F17.5.9

KEYWORD	DEFAULT	DEFINITION
TTL=	prob. title	delim ptitl delim delim is a one-character delimiter that signals the beginning and end of the title. ptitl is the plot title (max. 132 char.)
PIC=	MAT	Type of plot: MIXTURE, UNIT NO. or BIAS ID NO.
		MIXTURE ---------- MAT
		MIX
		MIXT
		MIXTURE
		MEDI
		MEDIA
		UNIT NO. ---------- 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=	8.0 (character plots) 10 (color plots)	Vertical to horizontal scaling factor for plot proportionality.
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
SCR=	YES	Display plot method SCR=YES utilizes color plot display SCR=NO utilizes printer plot display
NCH=	CHRS*	delim CHRS delim a one character delimiter signals the beginning and end of the character string
CLR=	Table F17.4.7	num(i) red(num(i)) green (num(i)) blue (num(i)) num(i) defines mix. no., unit no., or bias ID next 3 entries define red, green, and blue components of the color representing num(i).

PLOT ORIGIN:
(1) GLOBAL UNIT - coincides with origin of global unit geometry description.

*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 F17.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 Sect. F17.4.8.

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	

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=		fissile mixture no.
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

*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.
**Defaulted to the fissile mixture having the smallest mixture number.
Unless otherwise specified, the starting volume is defined by the unrotated, untranslated geometric sampe specified by the label in the first position in the global unit boundary record.

F17.D SAMPLE PROBLEMS

This section contains sample problems to demonstrate some of the options available in KENO-VI. All the options of KENO V.a are available in KENO VI. When KENO VI is run as a part of SCALE, the driver allows KENO VI to be executed each time it encounters an "=KENOVI" These KENO-VI problems are set up to run as stand-alone KENO-VI. A working format cross-section set was generated on unit 4, using BONAMI and NITAWL, for use with this problem set.

F17.D.1 KENO-VI SAMPLE PROBLEM DATA

A brief problem description and the associated record input data are included for each KENO-VI sample problem. Different options may be easily activated by making changes in the data. These problems use a 44-group AMPX working-format library on Unit 4 which was generated using BONAMI and NITAWL via CSASN from the 44GROUPNDF4 master library. The nuclide identifiers for this library are consistent with the SCALE identifiers created by CSAS. Input data to create this library are given in Sect. F17.D.2. The unit number is defined by the parameter LIB= in the parameter data.

SAMPLE PROBLEM 1 2C8 BARE

This simple 2 × 2 × 2 array of uranium metal cylinders is 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 F17.D.1 shows the critical experiment.

INPUT DATA

```
=kenovi
kenovi  sample problem 1  case 2c8 bare
read parameters
  flx=yes fdn=yes far=yes gas=no lib=4
end parameters
read mixt  sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry
unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary 20
global unit 2
cuboid 10  4p13.74 2p13.01
com='2x2x2  2c8 array'
array 1 +10 place 1 1 1 2r-6.87 -6.505
boundary  10
end geometry
read array  gbl=1 nux=2 nuy=2 nuz=2  fill f1  end array
end data
end
```

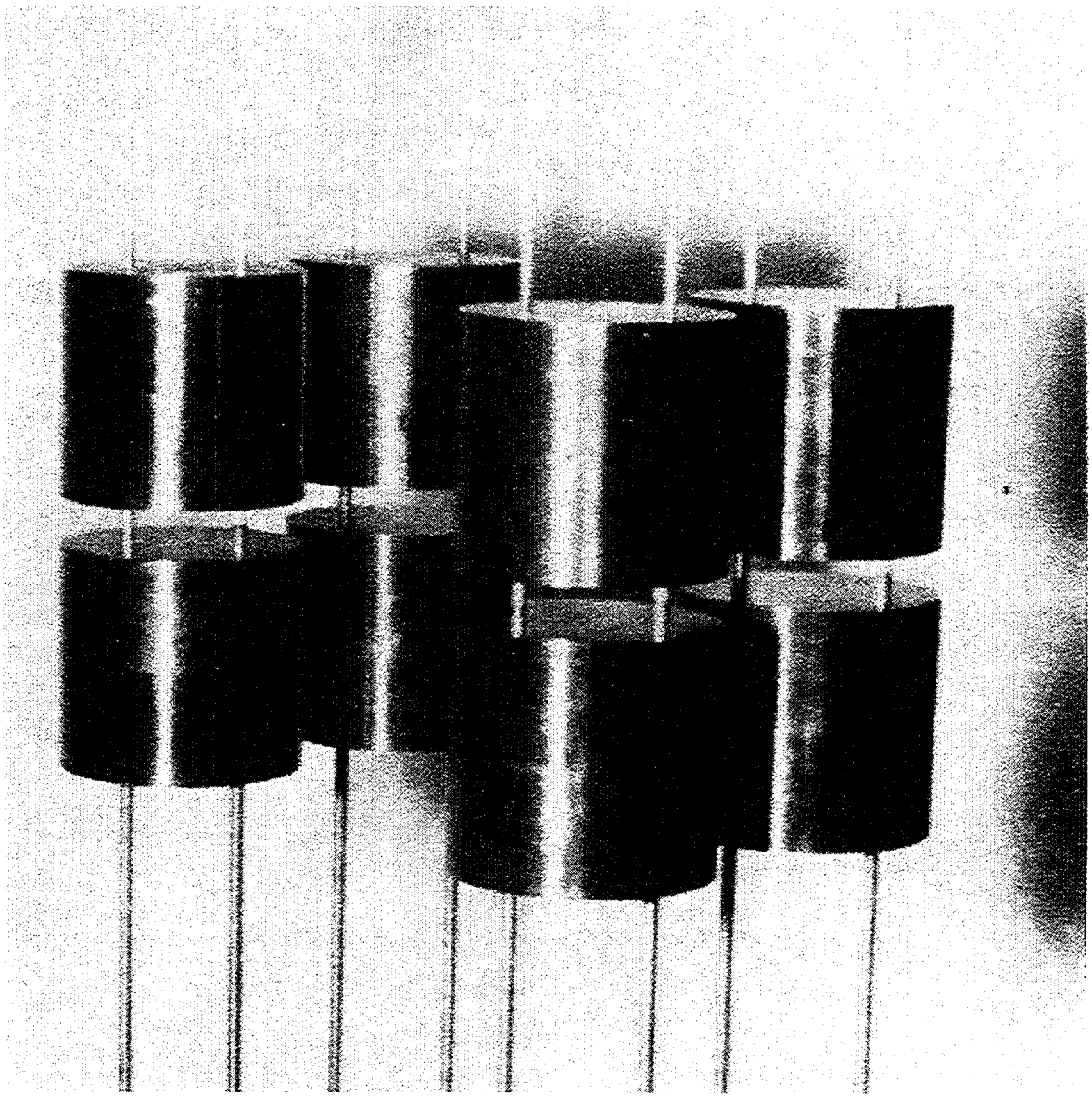


Figure F17.D.1 Critical 2C8 bare assembly

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.

INPUT DATA

```
=kenovi
kenovi sample problem 2  case 2c8 bare with 8 unit types matrix cal
read param
  lib=4 flx=yes fdn=yes mku=yes cku=yes fmu=yes mkp=yes ckp=yes fmp=yes
end parameters
read mixt  sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry   unit 1
cylinder 10 5.748 5.3825 -5.3825
cuboid   20 4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary   20
unit 2
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary   20
unit 3
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary  20
unit 4
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary 20
unit 5
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary  20
unit 6
cylinder 10  5.748 5.3825 -5.3825
cuboid   20 4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary  20
unit 7
cylinder 10  5.748 5.3825 -5.3825
cuboid   20 4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary 20
```

```
unit 8
cylinder 10  5.748 5.3825 -5.3825
cuboid   20 4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary  20
global unit 9
cuboid  10 4p13.74 2p13.01
com='2x2x2  2c8 array'
array 1 +10 place 1 1 1 2r-6.87 -6.505
boundary  10
end geometry
read array gbl=1 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
end
```

SAMPLE PROBLEM 3 2C8 15.24-CM PARAFFIN REFL

A 2 × 2 × 2 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 F17.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. F17.D.2, but it was present when criticality was achieved.

INPUT DATA

```
=kenovi
keno-vi sample problem 3  2c8  15.24 cm paraffin refl
read param lib=4 flx=yes fdn=yes pwt=yes end param
read mixt sct=2
mix=1  1092234 4.82716e-04  1092235 4.47971e-02 1092236 9.57231e-05
       1092238 2.65767e-03
mix=2 10006012 3.97311e-02 10001001 8.26407e-02
end mixt
read geometry
unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p11.74 2p11.375
media  1 1 10
media  0 1 20 -10
boundary  20
global unit 2
com='2x2x2  2c8 array with reflector'
cuboid 10 4p23.48 2p22.75
cuboid 20   26.48 -26.48 26.48 -26.48 25.75 -25.75
cuboid 30   29.48 -29.48 29.48 -29.48 28.75 -28.75
cuboid 40   32.48 -32.48 32.48 -32.48 31.75 -31.75
cuboid 50   35.48 -35.48 35.48 -35.48 34.75 -34.75
cuboid 60   38.72 -38.72 38.72 -38.72 37.99 -37.99
array 1 +10 place 1 1 1 2r-11.74 -11.375
media  2 2 -10 +20
media  2 3 -20 +30
media  2 4 -30 +40
media  2 5 -40 +50
media  2 6 60 -50
boundary  60
```

```
end geometry
read bias  id=400 2 6  end bias
read array  gbl=1 nux=2 nuy=2 nuz=2 fill f1 end array
end data
end
```

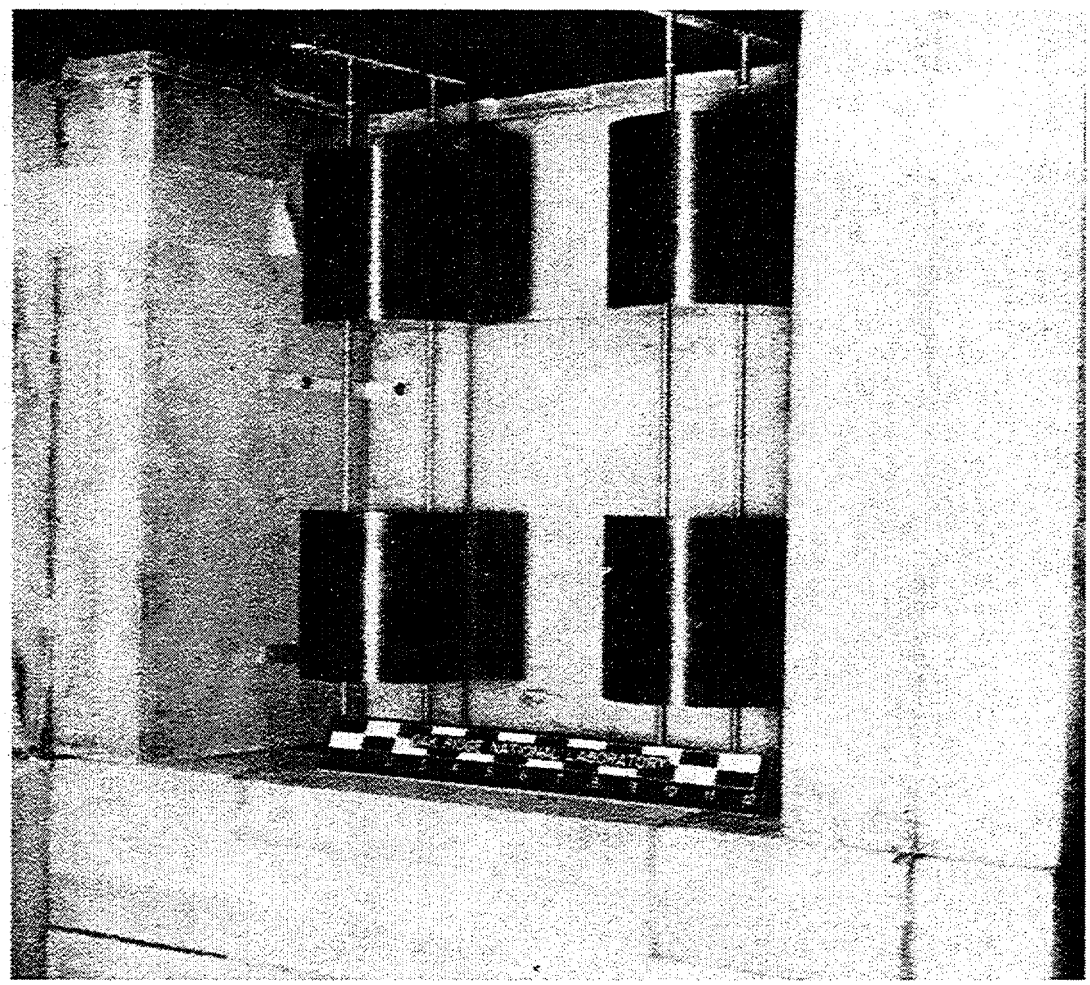


Figure F17.D.2 Half of the paraffin reflected 2C8 assembly before the top reflector was added

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 more reflector regions.

INPUT DATA

```
=kenovi
keno-vi sample problem 4  2c8  15.24 cm paraffin refl
read param lib=4 flx=yes fdn=yes pwt=yes end param
read mixt sct=2
mix=1  1092234 4.82716e-04  1092235 4.47971e-02  1092236 9.57231e-05
       1092238 2.65767e-03
mix=2 10006012 3.97311e-02 10001001 8.26407e-02
end mixt
read geometry
unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p11.74 2p11.375
media  1 1 10
media  0 1 20 -10
boundary  20
global unit 2
com='2x2x2  2c8 array with reflector'
cuboid 10 4p23.48 2p22.75
cuboid 20   26.48 -26.48 26.48 -26.48 25.75 -25.75
cuboid 30   29.48 -29.48 29.48 -29.48 28.75 -28.75
cuboid 40   32.48 -32.48 32.48 -32.48 31.75 -31.75
cuboid 50   35.48 -35.48 35.48 -35.48 34.75 -34.75
cuboid 60   38.48 -38.48 38.48 -38.48 37.75 -37.75
cuboid 70   38.72 -38.72 38.72 -38.72 37.99 -37.99
array 1 +10 place 1 1 1 2r-11.74 -11.375
media  2 2 -10 +20
media  2 3 -20 +30
media  2 4 -30 +40
media  2 5 -40 +50
media  2 6  60 -50
media  2 7  70 -60
boundary 70
end geometry
read bias  id=400 2 7  end bias
read array  gbl=1 nux=2 nuy=2 nuz=2 fill f1 end array
end data
end
```

SAMPLE PROBLEM 5 2C8 12-INCH PARAFFIN ALBEDO REFLECTOR

This problem is the same as samples 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]

INPUT DATA

```
=kenovi
kenovi sample problem 5  2c8 12 inch paraffin albedo reflector
read para flx=yes far=yes gas=no fdn=yes lib=4 end para
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read bounds  all=paraffin  end bounds
read geometry
unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
cuboid   20 4p11.74 2p11.375
media  1 1 10
media  0 1 20 -10
boundary  20
global unit 2
cuboid 10  4p23.48 2p22.75
com='2x2x2  2c8 array'
array 1 +10 place 1 1 1 2r-11.74 -11.375
boundary 10
end geometry
read array  gbl=1 nux=2 nuy=2 nuz=2  fill f1 end array
end data
end
```

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.

INPUT DATA

```
=kenovi
kenovi sample problem 6  one 2c8 unit (single unit)
read para lib=4 flx=yes fdn=yes far=yes gas=no end para
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry
global unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
media  1 1 10
boundary  10
end geometry
end data
end
```

SAMPLE PROBLEM 7 BARE 2C8 USING SPECULAR REFLECTION

One of the 2C8 units[1] is described and the 2 × 2 × 2 array is simulated by using specular reflection on the positive x, y, and z faces of the unit. This problem is a simulation of sample problem 1.

INPUT DATA

```
=kenovi
keno-vi  sample problem 7  bare 2c8 using specular reflection
read para flx=yes fdn=yes far=yes gas=no lib=4 end parameters
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry
global unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary  20
end geometry
read bounds  +fc=specular   end bounds   end data
end
```

SAMPLE PROBLEM 8 INFINITELY LONG CYLINDER FROM 2C8 UNIT

The fuel 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.

INPUT DATA

```
=kenovi
keno-vi  sample problem 8 infinitely long cylinder from 2c8 unit
read parameters lib=4  end parameters
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixtures
read geometry
global unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 2p10.0
cuboid   20  4p6.87 2p10.0
media  1 1 10
media  0 1 20 -10
boundary  20
end geometry
read bounds zfc=mirror end bounds
end data
end
```

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.

INPUT DATA

```
=kenovi
keno-vi  sample problem 9  infinite array of 2c8 units
read parameters lib=4 end parameters
read mixtures    sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
     1092238 2.65767e-03
end mixt
read geometry
global unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary  20
end geometry
read bounds  all=mirror   end bounds
end data
end
```

SAMPLE PROBLEM 10 2C8 BARE WRITE RESTART

This problem is the same as sample problem 1, a 2 × 2 × 2 array of metal cylinders. Restart information is written on unit 94 after the completion of every fifth generation.

INPUT DATA

```
=kenovi
sample problem 10  case 2c8 bare  write restart
read parameters
  flx=yes fdn=yes far=yes gas=no lib=4 res=5 wrs=94
end parameters
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixtures
read geometry
unit 1
com='single 2c8 unit centered'
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  4p6.87 2p6.505
media  1 1 10
media  0 1 20 -10
boundary  20
global unit 2
cuboid 10  4p13.74 2p13.01
com='2x2x2  2c8 array'
array 1 +10 place 1 1 1 2r-6.87 -6.505
boundary  10
end geometry
```

```
read array  gbl=1 nux=2 nuy=2 nuz=2  fill f1  end array
end data
end
```

SAMPLE PROBLEM 11 2C8 BARE READ RESTART DATA

This problem is a restart of sample problem 10. The problem is restarted from the tenth set of restart data that was written by sample problem 10 (i.e., it restarts with the fifty-first generation).

INPUT DATA

```
=kenovi
sample problem 11  2c8 bare  read restart data
read param beg=51 rst=94 res=0 end param
end data
end
```

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. F17.D.3.

INPUT DATA

```
=kenovi
sample problem 12 4 aqueous 4 metal mixed units
read param
  lib=4  flx=yes fdn=yes nub=yes smu=yes mku=yes fmp=yes fmu=yes
end param
read mixt  sct=2
mix=1  1092234 4.82716e-04   1092235 4.47971e-02   1092236 9.57231e-05
       1092238 2.65767e-03
mix=2  2001001 5.77964e-02   2007014 2.13092e-03   2008016 3.74130e-02
       2092234 1.06784e-05   2092235 9.84599e-04   2092236 5.29385e-06
       2092238 6.19413e-05
mix=3 11001001 5.68187e-02  11006012 3.55117e-02  11008016 1.42047e-02
end mixt
read geom
unit 1
cylinder 10  9.525 8.89 -8.89
cylinder 20  10.16 9.525 -9.525
cuboid   30  10.875 -10.875 10.875 -10.875 10.24 -10.24
media 2 1 10
media 3 1 -10 20
media 0 1 30 -20
boundary  30
unit      2
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  6.59 -15.16 6.59 -15.16 6.225 -14.255
media 1 1 10
```

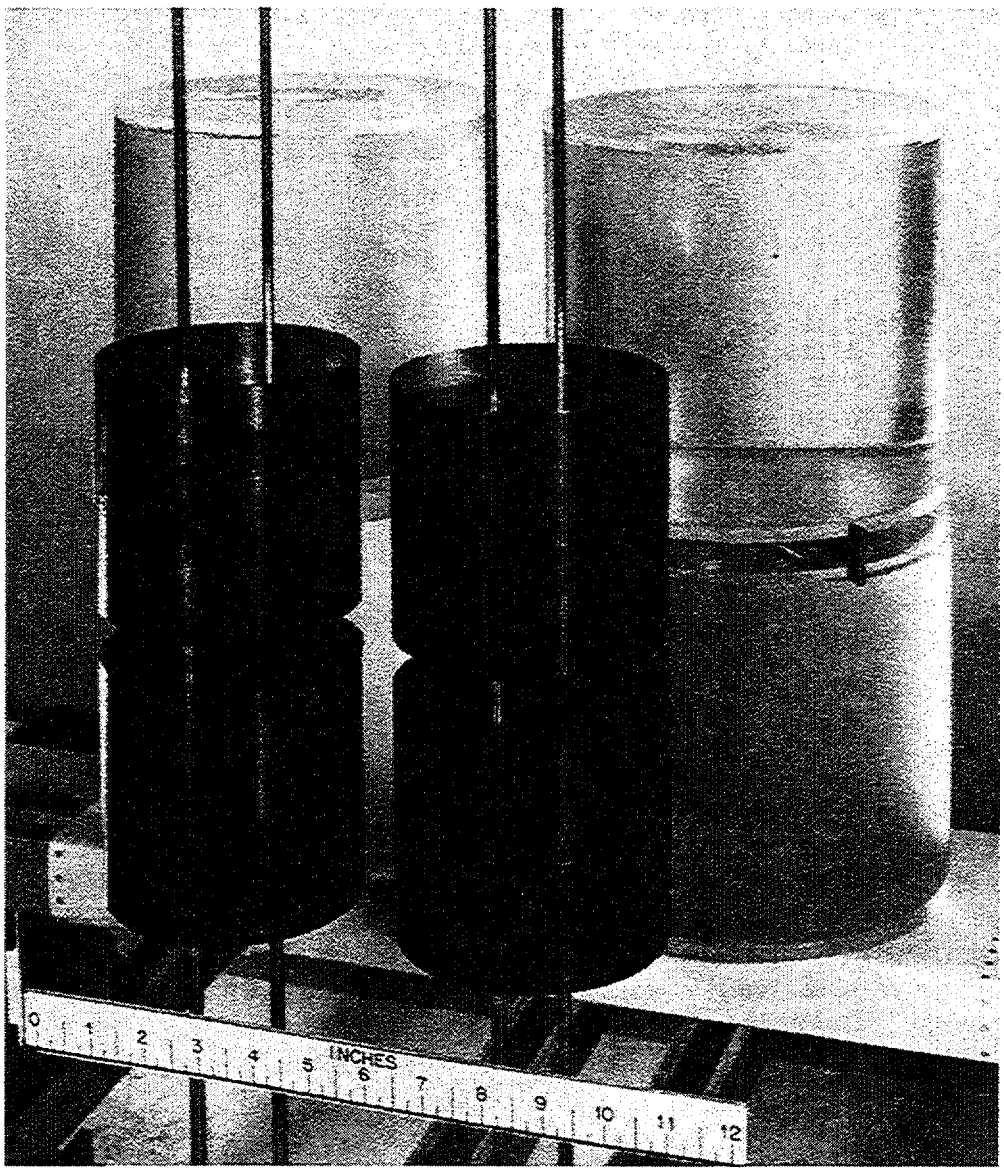


Figure F17.D.3 Critical assembly of 4 solution units and 4 metal units

```
media 0 1 20 -10
boundary  20
unit      3
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  6.59 -15.16 15.16 -6.59 6.225 -14.255
media 1 1 10
media 0 1 20 -10
boundary  20
unit      4
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  6.59 -15.16 6.59 -15.16 14.255 -6.225
media 1 1 10
media 0 1 20 -10
boundary  20
unit      5
cylinder 10  5.748 5.3825 -5.3825
cuboid   20  6.59 -15.16 15.16 -6.59 14.255 -6.225
media 1 1 10
media 0 1 20 -10
boundary  20
global
unit  6
cuboid 10  43.5 0.0 43.5 0.0 40.96 0.0
array 1 +10 place 1 1 1 15.16 15.16 14.255
boundary  10
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
end
```

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 = 18.69 g/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. F17.D.4.

INPUT DATA

```
=kenovi
sample problem 13  two cuboids in a cylindrical annulus
read param lib=4  end param
read mixt sct=2
mix=1  3092234 4.80915e-04  3092235 4.46299e-02  3092236 9.53659e-05
       3092238 2.64775e-03
end mixt
read geom
unit 1
cuboid    10 6.35 -6.35 6.35 -6.35 7.62 0.0
cylinder  20 13.97 7.62 0.0 orig x=-6.0934
cylinder  30 19.05 7.62 0.0 orig x=-6.0934
cuboid    40 12.9566 -25.1434 19.05 -19.05 7.62 0.0
media  1 1 10
media  0 1 20 -10
media  1 1 30 -20
```

```
media  0 1 40 -30
boundary  40
unit 2
cuboid   10 6.35 -6.35 6.35 -6.35 8.56 0.0
cylinder 20  13.97 8.56 0.0 origin x=6.0934
cylinder 30  19.05 8.56 0.0 origin x=6.0934
cuboid   40 25.1434 -12.9566 19.05 -19.05 8.56 0.0
media  1 1 10
media  0 1 20 -10
media  1 1 30 -20
media  0 1 40 -30
boundary  40
unit 3
cuboid 10  6.35 -6.35 6.35 -6.35 2.616 0.0
cuboid 20 25.1434 -12.9566 19.05 -19.05 2.616 0.0
media  1 1 10
media  0 1 20 -10
boundary  20
global unit 4
cuboid  10 12.9566 -25.1434 2p19.05 18.796 0.
array 1 10 place 1 1 1 3r0.
boundary  10
end geom
read array nux=1 nuy=1 nuz=3  fill 1 2 3 t end array end data
end
```

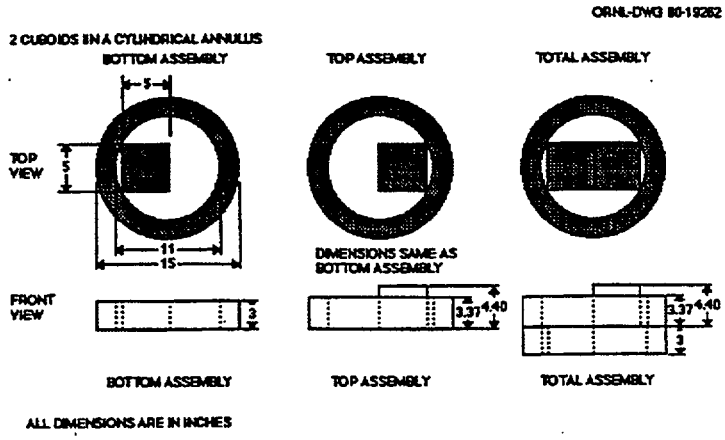


Figure F17.D.4 Drawing of two cuboids in an annulus critical assembly

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. F17.D.5. The uranium metal specification is identical to that used in sample problem 13.

INPUT DATA

```
=kenovi
sample problem 14  u metal cylinder in an annulus
read param lib=4 end param
read mixt sct=2
mix=1  3092234 4.80915e-04  3092235 4.46299e-02  3092236 9.53659e-05
       3092238 2.64775e-03
end mixt
read geom
global unit 1
cylinder 10  8.89 10.109 0.0 orig x=5.08
cylinder 20  13.97 10.109 0.0
cylinder 30  19.05 10.109 0.0
media  1 1 10
media  0 1 20 -10
media  1 1 30 -20 -10
boundary  30
end geom
end data
end
```

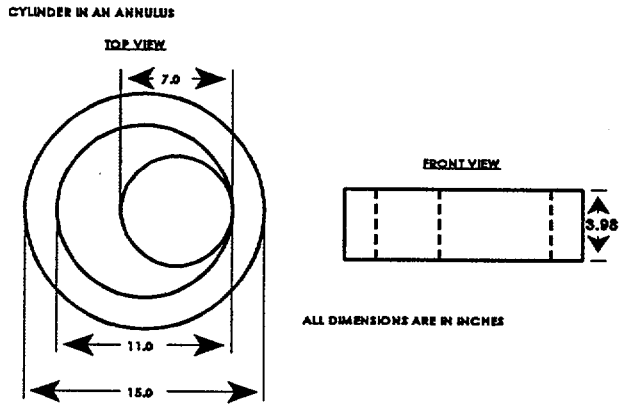


Figure F17.D.5 Drawing of the cylinder in an annulus critical assembly

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 VI geometry package cannot rigorously describe a doughnut; therefore, the KENO VI mockup 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. F17.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.

INPUT DATA

```
=kenovi
sample problem 15  small water reflected sphere on plexiglas collar
read param lib=4 flx=yes fdn=yes plt=yes end param
read mixt  sct=2
mix=1  4092234 5.65801e-04   4092235 4.70211e-02   4092236 9.58966e-05
       4092238 4.65935e-04
mix=2 11001001 5.68187e-02  11006012 3.55117e-02  11008016 1.42047e-02
mix=3 12001001 6.67514e-02
mix=3 12008016 3.33757e-02
end mixt
read start rfl=no end start
read geom
unit 1
sphere   10  6.5537
cylinder 20  4.1275 -5.09066 -7.63065
cylinder 30  12.7   -5.09066 -7.63065
cuboid   40  4p12.7  6.5537  -7.63065
media  1 1 10
media  3 1 20 -10
media  2 1 30 -20 -10
media  3 1 40 -30 -20 -10
boundary  40
global unit 2
cuboid    10 4p12.7 6.5537  -7.63065
cylinder  20 17.97  6.5537  -7.63065
cylinder  30 20.97  9.5537 -10.63065
cylinder  40 23.97 12.5537 -13.63065
cylinder  50 26.97 15.5537 -16.63065
cylinder  60 29.97 18.5537 -19.63065
cylinder  70 32.97 21.5537 -22.63065
array 1 10 place 1 1 1 0.0 0.0 0.0
media  3 1 20 -10
media  3 2 30 -20 -10
media  3 3 40 -30
media  3 4 50 -40
media  3 5 60 -50
media  3 6 70 -60
boundary  70
end geom
read array nux=1 nuy=1 nuz=1 fill 1  end fill   end array
read bias  id=500 2 6 end bias
read plot scr=yes  lpi=10
ttl='x-z slice through 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=400 end plot
end data
end
```

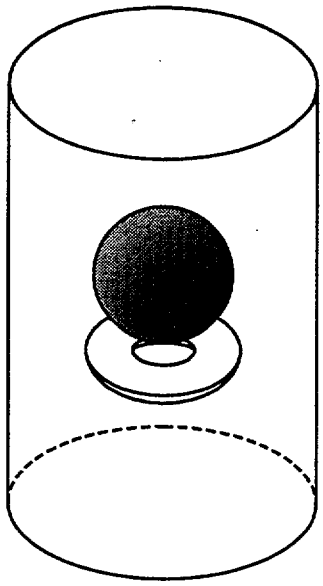


Figure F17.D.6 Drawing of a critical assembly consisting of a uranium sphere on a plexiglas collar with a cylindrical water reflector

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 fluoride solution. The uranyl fluoride slab is 4.958 cm thick, 93.2% enriched, and has a density of 578.7 g U/L. 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 are present in the borated solution.

INPUT DATA

```
=kenovi
sample problem 16 uo2f2 infinite slab k-infinity
read parameters lib=4 amx=yes xap=no   end parameters
read mixt  sct=2
mix=1  5009019 2.96286e-03  5001001 6.04824e-02  5008016 3.32041e-02
       5092235 1.38188e-03  5092238 9.95503e-05
mix=2 13011023 2.39502e-03 13013027 4.97719e-04 13014000 1.80267e-02
      13005010 9.08241e-04 13005011 3.68719e-03 13008016 4.49173e-02
mix=3  6009019 2.96286e-03  6001001 6.04824e-02  6008016 3.32041e-02
       6092235 1.38188e-03  6092238 9.95503e-05
       6005010 2.92803e-04  6005011 1.18870e-03
end mixt
read geometry
global unit 1
cuboid 10  2.479 -2.479 100.0 -100.0 100.0 -100.0
cuboid 20  3.749 -3.749 100.0 -100.0 100.0 -100.0
cuboid 30  17.479 -17.479 100.0 -100.0 100.0 -100.0
media 1 1 10
media 2 1 20 -10
media 3 1 30 -20 -10
boundary  30
end geom
read bounds all=mirror end bounds
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.

INPUT DATA

```
=kenovi
sample problem 17 93% uo2f2 solution sphere  adjoint calculation
read parameters
  lib=4 amx=yes pwt=yes adj=yes npg=10000 lng=500000
end parameters
read mixt  sct=2
mix=1  7001001 6.54785e-02  7008016 3.34202e-02  7009019 6.80923e-04
       7092235 3.16909e-04  7092238 2.35521e-05
end mixt
read geometry
global unit 1
sphere 10 16.0
media 1 1 10
```

```
boundary  10
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 L of 92.6% enriched solution at a concentration of 415 g/L, a specific gravity of 1.555 and 0.39* mg excess nitrate/g soln. The 3 × 3 × 3 array was surrounded by a 6-in. paraffin reflector. Most of the print options available in KENO-VI are exercised in this problem. A perspective of this critical experiment is shown in Fig. F17.D.7. A photograph of one of the experiments utilized 27 of the plexiglas bottles is shown in Fig. F17.D.8. Sample problem 18 has 15.24 cm of paraffin on all six faces rather than the 2.54-cm plexiglas shown on five faces.

INPUT DATA

```
=kenovi
sample problem 18   1f27  critical experiment
read para
 gen=103 npg=500 lng=100000 fdn=yes nub=yes lib=4 plt=yes
 mku=yes cku=yes fmu=yes fmh=yes mka=yes cka=yes fma=yes pwt=yes
 far=yes flx=yes amx=yes pax=yes pgm=yes rnd=f12c09ed2195
end para
read mixt sct=2
 mix=1  2001001 5.77964e-02  2007014 2.13092e-03   2008016 3.74130e-02
        2092234 1.06784e-05  2092235 9.84599e-04   2092236 5.29385e-06
        2092238 6.19413e-05
 mix=2 11001001 5.68187e-02 11006012 3.55117e-02  11008016 1.42047e-02
 mix=3 10006012 3.97311e-02 10001001 8.26407e-02
 mix=4 15008016 3.33757e-11 15001001 6.67514e-11
 end mixt
read bounds  -zb=h2o  end bounds
read geom
unit 1
 cylinder 10  9.52 8.7804 -8.7804
 cylinder 20  9.52 8.9896 -8.7804
 cylinder 30  10.16 9.6296 -9.4204
 cuboid   40  18.45 -18.45 18.45 -18.45 17.8946 -17.6854
 media 1 1 10
 media 0 1 -10 20
 media 2 1 -10 -20 30
 media 0 1 40 -20 -30
 boundary  40
unit 2
 cuboid 10  18.45 -55.35  55.35 -18.45  53.37 -17.79
 cuboid 20  18.45 -55.35  55.35 -18.45 -17.79 -53.37
 cuboid 30  55.35  18.45  55.35 -18.45  53.37 -53.37
 cuboid 40  55.35 -55.35 -18.45 -55.35  53.37 -53.37
 cuboid 50  55.35 -55.35  55.35 -55.35  53.37 -53.37
 array  1 10 place 1 1 1 -36.90 0.0 -0.1046
 array  2 20 -10 place 1 1 1 -36.90 0.0 -35.6846
 array  3 30 -20 -10 place 1 1 1 36.90 0.0 -35.6846
 array  4 40 -30 -20 -10 place 1 1 1 -36.90 -36.90 -35.6846
 media 0 1 50 -40 -30 -20 -10
```

*From experimental facility documents (not reported in ORNL/TM-719).

```
 boundary  50
global unit 3
 cuboid 10  55.35 -55.35 55.35 -55.35 53.37 -53.37
 cuboid 20  58.35 -58.35 58.35 -58.35 56.37 -56.37
 cuboid 30  61.35 -61.35 61.35 -61.35 59.37 -59.37
 cuboid 40  64.35 -64.35 64.35 -64.35 62.37 -62.37
 cuboid 50  67.35 -67.35 67.35 -67.35 65.37 -65.37
 cuboid 60  70.59 -70.59 70.59 -70.59 68.61 -68.61
 array 5 10 place 1 1 1 3*0.0
 media 3 2 -10 20
 media 3 3 -20 30
 media 3 4 -30 40
 media 3 5 -40 50
 media 3 6 60 -50
 boundary  60
 end geom
 read bias  id=400 2 6  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
 gbl=5 ara=5 nux=1 nuy=1 nuz=1 fill f2 end fill
end array
read plot  scr=yes lpi=10
 ttl='  1f27 xy plot at z=0.0 '
 xul=-71.0 yul=71.0 zul=0.0 xlr=71.0 ylr=-71.0 zlr=0.0
 uax=1 vdn=-1 nax=400 end plt0
 ttl='unit map 1f27 xy plot at z=0.0'
 pic=unit end plot
end data
end
```

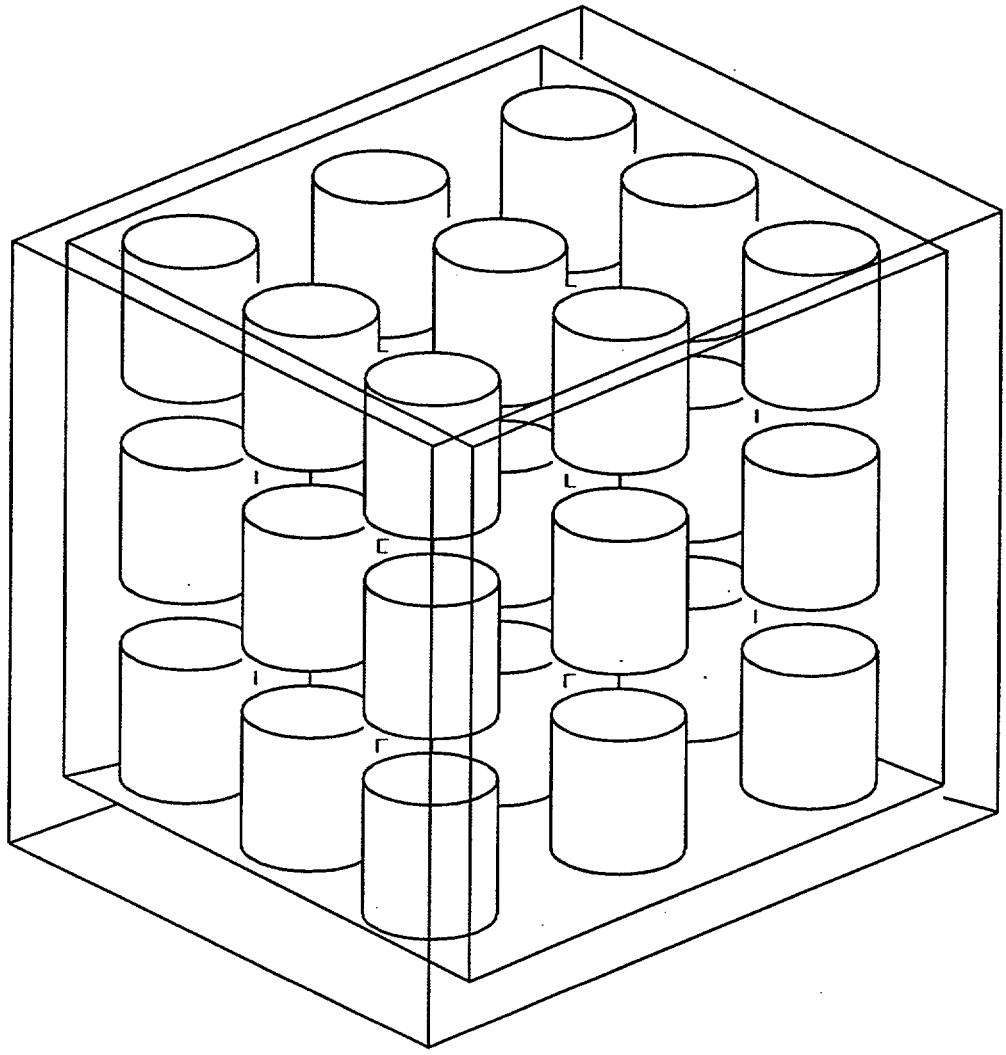


Figure F17.D.7 Perspective of critical 1F27 experiment

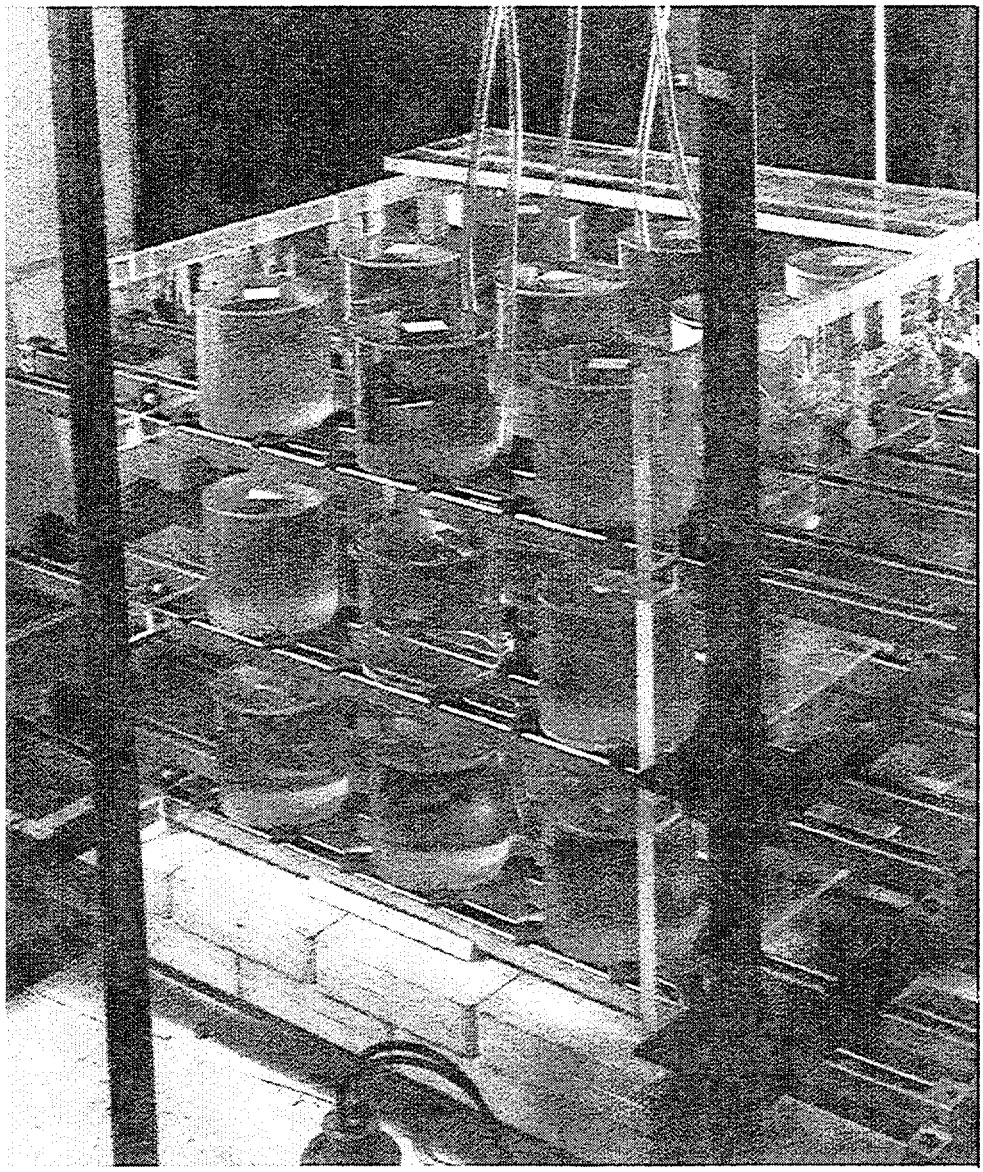


Figure F17.D.8 View of a 27-unit cubic array with 2.54-cm-thick plexiglas reflector on five sides and a 15.24-cm-thick paraffin base

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 utilize the array of arrays option. See Fig. F17.D.3.

INPUT DATA

```
=kenovi
sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)
read param
  lib=4 flx=yes fdn=yes nub=yes smu=yes mkp=yes mku=yes fmp=yes fmu=yes
end param
read mixt sct=2
mix=1  1092234 4.82716e-04   1092235 4.47971e-02   1092236 9.57231e-05
       1092238 2.65767e-03
mix=2  2001001 5.77964e-02   2007014 2.13092e-03   2008016 3.74130e-02
       2092234 1.06784e-05   2092235 9.84599e-04   2092236 5.29385e-06
       2092238 6.19413e-05
mix=3 11001001 5.68187e-02  11006012 3.55117e-02  11008016 1.42047e-02
end mixt
read geometry
unit 1
com='uranyl nitrate solution in a plexiglas container'
cylinder 10   9.525 2p8.89
cylinder 20   10.16 2p9.525
cuboid   30  4p10.875 2p10.24
media 2 1 10
media 3 1 -10 20
media 0 1 30 -20
boundary  30
unit 2
com='uranium metal cylinder'
cylinder 10  5.748 2p5.3825
cuboid   20 4p6.59 2p6.225
media 1 1 10
media 0 1 20 -10
boundary  20
unit 3
com='1x2x2 array of solution units'
cuboid 10  21.75 0.0 43.5 0.0 40.96 0.0
array 1 +10 place 1 1 1 10.875 10.875 10.240
boundary  10
unit 4
com='1x2x2 array of metal units padded to match solution array'
cuboid 10  13.18 0.0 26.36 0.0 24.9 0.0
cuboid 20  13.18 0.0 34.93 -8.57 32.93 -8.03
array 2 +10 place 1 1 1 6.59 6.59 6.225
media 0 1 20 -10
boundary  20
global unit 5
com='global unit of arrays 1 and 2'
cuboid 10 34.93 0.0 43.5 0.0 40.96 0.0
array 3 +10 place 1 1 1 0 8.57 8.03
boundary  10
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
end
```

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 in. 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 fluoride with a $H/^{235}U$ atomic ratio of 44.3 and a density of 576.87 g U/L. The apparatus for conducting this experiment is shown in Fig. F17.D.9.

INPUT DATA

```
=kenovi
sample problem 20  triangular pitched array 7 pins in a circle
read parameters lib=4 end parameters
read mixt sct=2
mix=1  8092235 1.37751e-03  8092238 9.92354e-05  8008016 3.32049e-02
       8009019 2.95349e-03  8001001 6.05028e-02
mix=2 14013027 6.02374e-02
end mixt
read geometry
unit 1
com='single cell fuel can in hexprism'
cylinder 10  10.16  18.288  0.0
cylinder 20  10.312 18.288 -0.152
hexprism 30  10.503 18.288 -0.152
media  1 1 10
media  2 1 20 -10
media 0 1 30 -20
boundary  30
unit 2
com='empty cell'
hexprism 10  10.503 18.288 -0.152
media  0 1 10
boundary  10
global unit 3
cylinder 10  31.500 18.288 -0.152
com='7 cylinders in a circle with cylindrical boundary'
array  1 10 place 3 3 1 3*0.0
boundary  10
end geometry
read array  gbl=1 typ=triangular nux=5 nuy=5 nuz=1
fill 7*2 2*1 2*2 3*1 2*2 2*1 7*2 end fill  end array
end data
end
```

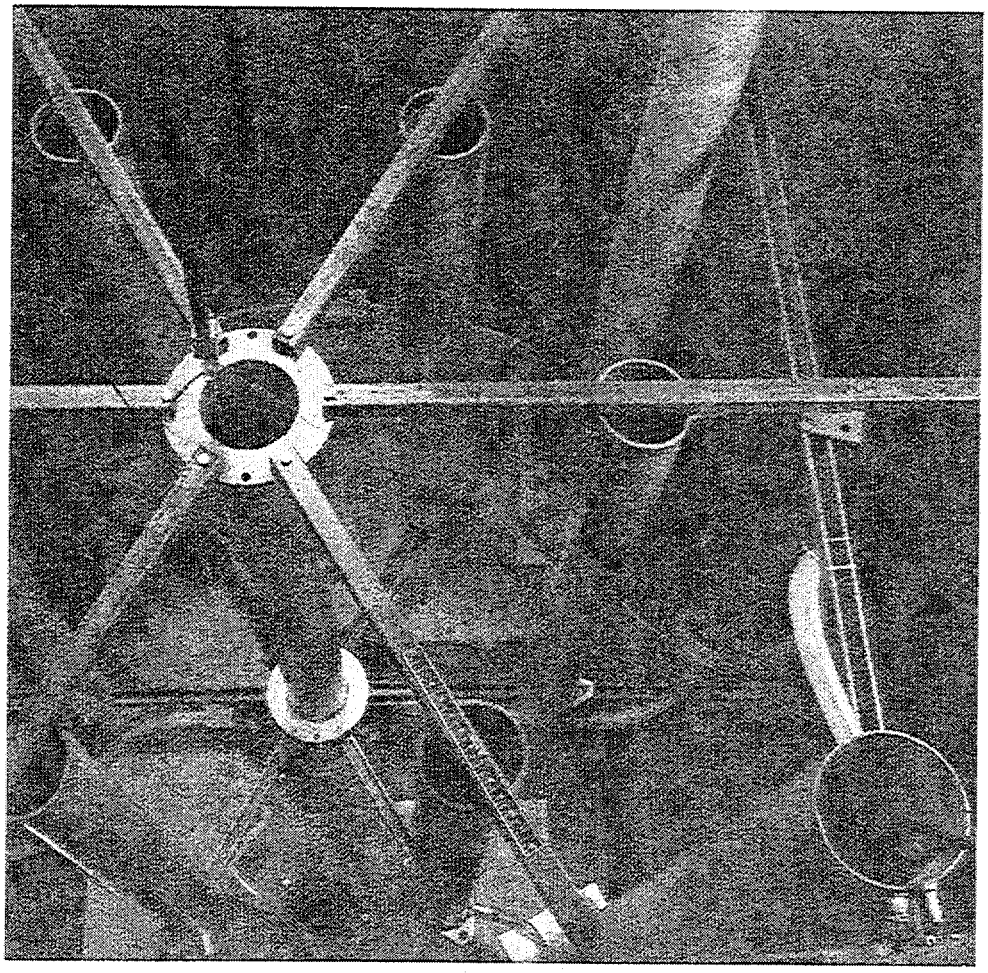


Figure F17.D.9 Typical arrangement for critical experiments with interacting arrays of aluminum cylinders containing enriched ^{235}U solutions

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.-diam 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. F17.D.10. The steel tank was ignored.

INPUT DATA

```
=kenovi
sample problem 21  partially filled sphere
read param lib=4 end param
read mixt sct=2
mix=1  9001001 6.15670e-02  9008016 3.32845e-02  9009019 2.50098e-03
       9092234 2.54223e-07  9092235 6.18922e-05  9092238 1.18834e-03
mix=2 14013027 6.02374e-02
end mixt
read geom
global unit 1
sphere 10 34.6  chord -z=30.0
sphere 20 34.6
sphere 30 34.759
media 1 1 10
media 0 1 20 -10
media 2 1 30 -20 -10
boundary  30
end geom
end data
end
```

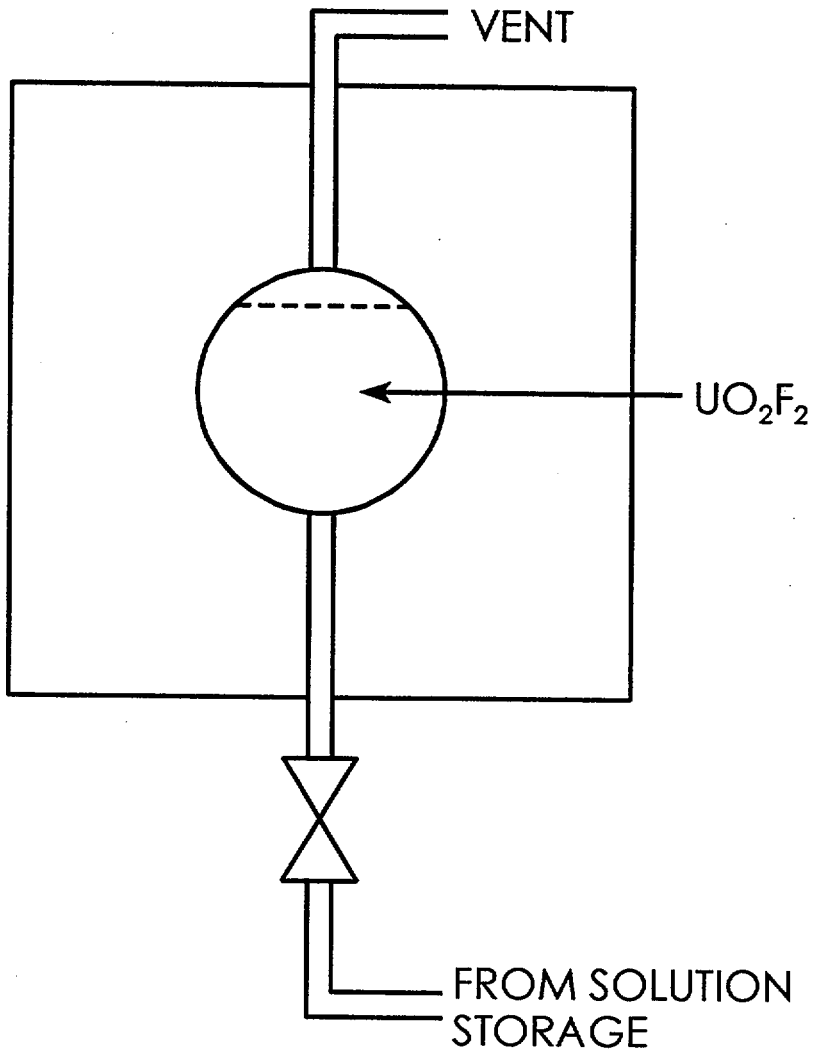


Figure F17.D.10 Schematic of bare partially filled sphere experiment inside a 9.5-ft-diam, 9-ft-high steel tank

SAMPLE PROBLEM 22 CASE 2C8 BARE WITH 3 NESTED HOLES, EACH IS EQUAL VOLUME

The physical representation of this sample problem is the critical experiment described in sample problem 1. It is a simple 2 × 2 × 2 array of 93.2% enriched uranium metal cylinders. This sample problem defines a uranium cylinder in a void spacing cuboid using nested holes. Eight of these units are stacked together in a 2 × 2 × 2 array.

INPUT DATA

```
=kenovi
sample problem 22  case 2c8 bare with 3 nested, equal volume holes
read parameters
  flx=yes fdn=yes far=yes gas=no lib=4 mkh=yes ckh=yes fmh=yes
end parameters
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry
unit 1
cylinder 10 3.621 2p3.3907
media 1 1 10
boundary  10
unit 2
cylinder 20 4.5622 2p4.2721
hole 1
media 1 1 20
boundary 20
unit 3
cylinder 20 5.2224 2p4.8903
hole 2
media 1 1 20
boundary  20
unit 4
cylinder 20  5.748 2p5.3825
cuboid   30 6.87 -6.87 6.87 -6.87 6.505 -6.505
hole 3
media 1 1 20
media 0 1 30 -20
boundary  30
global unit 5
cuboid 10 20.61 -6.87 20.61 -6.87 19.515 -6.505
array 1 10 place 1 1 1 3*0.0
boundary  10
end geometry
read array nux=2 nuy=2 nuz=2 fill f4 end fill end array
end data
end
```

SAMPLE PROBLEM 23 CASE 2C8 BARE AS STACKED CYLINDERS

The physical representation of this sample problem is the critical experiment described in sample problem 1. This sample problem describes each of the eight units in the critical 2 × 2 × 2 array using hemicylinders with different chord sizes and directions.

INPUT DATA

```
=kenovi
sample problem 23  case 2c8 bare as mixed unrotated zcylinders
read parameters fdn=yes lib=4 end parameters
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry
unit 1
com='-x half of unit 3'
cylinder 10 5.748 5.3825 -5.3825 chord -x=0.0
cuboid   20 0.0 -6.87 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 2
com='+x half of unit 3'
cylinder 10 5.748 5.3825 -5.3825 chord +x=0.0
cuboid   20 6.87 0.0 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 3
com='cylinder composed of equal halves (zhemicylinders with x radii)'
cuboid 10 6.87 -6.87 6.87 -6.87 6.505 -6.505
array 1 10 place 1 1 1 0.0 0.0 0.0
boundary  10
unit 4
com='-x portion (more than half) of unit 6'
cylinder 10 5.748 5.3825 -5.3825 chord -x=3.0
cuboid   20 3.0 -6.87 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 5
com='+x portion (less than half) of unit 6'
cylinder 10 5.748 5.3825 -5.3825 chord +x=3.0
cuboid   20 6.87 3.0 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 6
com='cylinder composed of unequal halves (zhemicylinders with x radii)'
cuboid 10 6.87 -6.87 6.87 -6.87 6.505 -6.505
array 2 10 place 1 1 1 3*0.0
boundary  10
unit 7
com='cylinder of a single zhemicylinder in the -x direction'
cylinder 10 5.748 5.3825 -5.3825 chord -x=5.748
cuboid   20 6.87 -6.87 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
```

```
unit 8
com='cylinder of a single zhemicylinder in the +x direction'
cylinder 10 5.748 5.3825 -5.3825 chord +x=-5.748
cuboid   20 6.87 -6.87 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 9
com='-y half of unit 11'
cylinder 10 5.748 5.3825 -5.3825 chord -y=0.0
cuboid   20 6.87 -6.87 0.0 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 10
com='+y half of unit 11'
cylinder 10 5.748 5.3825 -5.3825 chord +y=0.0
cuboid   20 6.87 -6.87 6.87 0.0 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 11
com='cylinder composed of equal halves (zhemicylinders with y radii)'
cuboid 10 6.87 -6.87 6.87 -6.87 6.505 -6.505
array 3 10 place 1 1 1 0.0 0.0 0.0
boundary  10
unit 12
com='-y portion (more than half) of unit 14'
cylinder 10 5.748 5.3825 -5.3825 chord -y=3.0
cuboid   20 6.87 -6.87 3.0 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 13
com='+y portion (less than half) of unit 14'
cylinder 10 5.748 5.3825 -5.3825 chord +y=3.0
cuboid   20 6.87 -6.87 6.87 3.0 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 14
com='cylinder composed of unequal halves (zhemicylinders with y radii)'
cuboid 10 6.87 -6.87 6.87 -6.87 6.505 -6.505
array 4 10 place 1 1 1 3*0.0
boundary  10
unit 15
com='cylinder of a single zhemicylinder in the -y direction'
cylinder 10 5.748 5.3825 -5.3825 chord -y=5.748
cuboid   20 6.87 -6.87 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
unit 16
com='cylinder of a single zhemicylinder in the +y direction'
cylinder 10 5.748 5.3825 -5.3825 chord +y=-5.748
cuboid   20 6.87 -6.87 6.87 -6.87 6.505 -6.505
media 1 1 10
media 0 1 20 -10
boundary  20
global unit 17
cuboid 10 13.74 -13.74 13.74 -13.74 13.010 -13.010
array 5 10 place 1 1 1 -6.87 -6.87 -6.505
boundary  10
end geometry
```

```
read array
com='array 1 defines unit 3 (zhemicylinders with x radii)'
ara=1 nux=2 nuy=1 nuz=1 fill 1 2 end fill
com='array 2 defines unit 6 (zhemicylinders with x radii)'
ara=2 nux=2 nuy=1 nuz=1 fill 4 5 end fill
com='array 3 defines unit 11 (zhemicylinders with y radii)'
ara=3 nux=1 nuy=2 nuz=1 fill 9 10 end fill
com='array 4 defines unit 14 (zhemicylinders with y radii)'
ara=4 nuz=1 nuy=2 nuz=1 fill 12 13 end fill
com='array 5 defines the total 2c8 problem'
gbl=5 ara=5 nux=2 nuy=2 nuz=2 fill 3 7 6 8 11 15 14 16 end fill
end array
end data
end
```

SAMPLE PROBLEM 24 CASE 2C8 BARE AS STACKED X-ROTATED CYLINDERS

The physical representation of this sample problem is the critical experiment described in sample problem 1 except the long axis of the cylinders are oriented along the x-axis This sample problem describes each of the eight units in the critical 2 × 2 × 2 array using hemicylinders with different chord sizes and directions whose long axes are rotated in the x-direction.

INPUT DATA

```
=kenovi
sample problem 24  case 2c8 bare as mixed x-rotated cylinders
read parameters rnd=4c6a61962572 fdn=yes lib=4 end parameters
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry
unit 1
com='-y half of unit 3'
cylinder 10 5.748 5.3825 -5.3825 chord -x=0.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 0.0 -6.87 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 2
com='+y half of unit 3'
cylinder 10 5.748 5.3825 -5.3825 chord +x=0.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 0.0 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 3
com='cylinder composed of equal halves (xhemicylinders with y radii)'
cuboid 10 6.505 -6.505 6.87 -6.87 6.87 -6.87
array 1 10 place 1 1 1 0.0 0.0 0.0
boundary  10
unit 4
com='-y portion (more than half) of unit 6'
cylinder 10 5.748 5.3825 -5.3825 chord -x=3.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 3.0 -6.87 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
```

```
unit 5
com='+y portion (less than half) of unit 6'
cylinder 10 5.748 5.3825 -5.3825 chord +x=3.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 3.0 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 6
com='cylinder composed of unequal halves (xhemicylinders with y radii)'
cuboid 10 6.505 -6.505 6.87 -6.87 6.87 -6.87
array 2 10 place 1 1 1 3*0.0
boundary  10
unit 7
com='cylinder of a single xhemicylinder in the -y direction'
cylinder 10 5.748 5.3825 -5.3825 chord -x=5.748 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 8
com='cylinder of a single xhemicylinder in the +y direction'
cylinder 10 5.748 5.3825 -5.3825 chord +x=-5.748 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 9
com='-z half of unit 11'
cylinder 10 5.748 5.3825 -5.3825 chord -y=0.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 0.0 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 10
com='+z half of unit 11'
cylinder 10 5.748 5.3825 -5.3825 chord +y=0.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 6.87 0.0
media 1 1 10
media 0 1 20 -10
boundary  20
unit 11
com='cylinder composed of equal halves (xhemicylinders with z radii)'
cuboid 10 6.505 -6.505 6.87 -6.87 6.87 -6.87
array 3 10 place 1 1 1 0.0 0.0 0.0
boundary  10
unit 12
com='-z portion (more than half) of unit 14'
cylinder 10 5.748 5.3825 -5.3825 chord -y=3.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 3.0 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 13
com='+z portion (less than half) of unit 14'
cylinder 10 5.748 5.3825 -5.3825 chord +y=3.0 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 6.87 3.0
media 1 1 10
media 0 1 20 -10
boundary  20
unit 14
com='cylinder composed of unequal halves (xhemicylinders with z radii)'
cuboid 10 6.505 -6.505 6.87 -6.87 6.87 -6.87
array 4 10 place 1 1 1 3*0.0
boundary  10
```

```
unit 15
com='cylinder of a single xhemicylinder in the -z direction'
cylinder 10 5.748 5.3825 -5.3825 chord -y=5.748 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 16
com='cylinder of a single xhemicylinder in the +z direction'
cylinder 10 5.748 5.3825 -5.3825 chord +y=-5.748 rotate a1=90 a2=90
cuboid   20 6.505 -6.505 6.87 -6.87 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
global unit 17
cuboid 10 13.01 -13.01 13.74 -13.74 13.74 -13.74
array 5 10 place 1 1 1 -6.505 -6.87 -6.87
boundary  10
end geometry
read array
com='array 1 defines unit 3 (xhemicylinders with y radii)'
ara=1 nux=1 nuy=2 nuz=1 fill 1 2 end fill
com='array 2 defines unit 6 (xhemicylinders with y radii)'
ara=2 nux=1 nuy=2 nuz=1 fill 4 5 end fill
com='array 3 defines unit 11 (xhemicylinders with z radii)'
ara=3 nux=1 nuy=1 nuz=2 fill 9 10 end fill
com='array 4 defines unit 14 (xhemicylinders with z radii)'
ara=4 nux=1 nuy=1 nuz=2 fill 12 13 end fill
com='array 5 defines the total 2c8 problem'
ara=5 nux=2 nuy=2 nuz=2 fill 3 7 6 8 11 15 14 16 end fill
end array
end data
end
```

SAMPLE PROBLEM 25 CASE 2C8 BARE AS MIXED YHEMICYLINDERS

The physical representation of this sample problem is the critical experiment described in sample problem 1 except the long axis of the cylinders are oriented along the y-axis. This sample problem describes each of th eight units in the critical 2 × 2 × 2 array using hemicylinders with different chord sizes and directions whose long axes are rotated in the y-direction.

INPUT DATA

```
=kenovi
sample problem 25  case 2c8 bare as mixed y-rotated cylinders
read parameters fdn=yes lib=4 end parameters
read mixt sct=2
mix=1 1092234 4.82716e-04 1092235 4.47971e-02 1092236 9.57231e-05
      1092238 2.65767e-03
end mixt
read geometry
unit 1
com='-x half of unit 3'
cylinder 10 5.748 5.3825 -5.3825 chord -y=0.0 rotate a1=180 a2=90 a3=90
cuboid   20 0.0 -6.87 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
```

```
unit 2
com='+x half of unit 3'
cylinder 10 5.748 5.3825 -5.3825 chord +y=0.0 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 0.0 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 3
com='cylinder composed of equal halves (yhemicylinders with x radii)'
cuboid 10 6.87 -6.87 6.505 -6.505 6.87 -6.87
array 1 10 place 1 1 1 0.0 0.0 0.0
boundary  10
unit 4
com='-x portion (more than half) of unit 6'
cylinder 10 5.748 5.3825 -5.3825 chord -y=3.0 rotate a1=180 a2=90 a3=90
cuboid   20 3.0 -6.87 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 5
com='+x portion (less than half) of unit 6'
cylinder 10 5.748 5.3825 -5.3825 chord +y=3.0 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 3.0 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 6
com='cylinder composed of unequal halves (yhemicylinders with x radii)'
cuboid 10 6.87 -6.87 6.505 -6.505 6.87 -6.87
array 2 10 place 1 1 1 3*0.0
boundary  10
unit 7
com='cylinder of a single yhemicylinder in the -x direction'
cylinder 10 5.748 5.3825 -5.3825 chord -y=5.748 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 8
com='cylinder of a single yhemicylinder in the +x direction'
cylinder 10 5.748 5.3825 -5.3825 chord +y=-5.748 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 9
com='-z half of unit 11'
cylinder 10 5.748 5.3825 -5.3825 chord -x=0.0 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 0.0 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 10
com='+z half of unit 11'
cylinder 10 5.748 5.3825 -5.3825 chord +x=0.0 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 6.87 0.0
media 1 1 10
media 0 1 20 -10
boundary  20
```

```
unit 11
com='cylinder composed of equal halves (yhemicylinders with z radii)'
cuboid 10 6.87 -6.87 6.505 -6.505 6.87 -6.87
array 3 10 place 1 1 1 0.0 0.0 0.0
boundary  10
unit 12
com='-z portion (more than half) of unit 14'
cylinder 10 5.748 5.3825 -5.3825 chord -x=3.0 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 3.0 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 13
com='+z portion (less than half) of unit 14'
cylinder 10 5.748 5.3825 -5.3825 chord +x=3.0 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 6.87 3.0
media 1 1 10
media 0 1 20 -10
boundary  20
unit 14
com='cylinder composed of unequal halves (yhemicylinders with z radii)'
cuboid 10 6.87 -6.87 6.505 -6.505 6.87 -6.87
array 4 10 place 1 1 1 3*0.0
boundary  10
unit 15
com='cylinder of a single yhemicylinder in the -z direction'
cylinder 10 5.748 5.3825 -5.3825 chord -x=5.748 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
unit 16
com='cylinder of a single yhemicylinder in the +z direction'
cylinder 10 5.748 5.3825 -5.3825 chord +x=-5.748 rotate a1=180 a2=90 a3=90
cuboid   20 6.87 -6.87 6.505 -6.505 6.87 -6.87
media 1 1 10
media 0 1 20 -10
boundary  20
global unit 17
cuboid 10 13.74 -13.74 13.01 -13.01 13.74 -13.74
array 5 10 place 1 1 1 -6.87 -6.505 -6.87
boundary  10
end geometry
read array
com='array 1 defines unit 3 (yhemicylinders with z radii)'
ara=1 nux=2 nuy=1 nuz=1 fill 1 2 end fill
com='array 2 defines unit 6 (yhemicylinders with z radii)'
ara=2 nux=2 nuy=1 nuz=1 fill 4 5 end fill
com='array 3 defines unit 11 (yhemicylinders with x radii)'
ara=3 nux=1 nuy=1 nuz=2 fill 9 10 end fill
com='array 4 defines unit 14 (yhemicylinders with x radii)'
ara=4 nux=1 nuy=1 nuz=2 fill 12 13 end fill
com='array 5 defines the total 2c8 problem'
gbl=5 ara=5 nux=2 nuy=2 nuz=2 fill 3 7 6 8 11 15 14 16 end fill
end array
end data
end
```

SAMPLE PROBLEM 26 BARE CRITICAL SPHERE 3.4420-IN. RADIUS

This problem is a critical experiment[9] consisting of a critical Oralloy sphere. The density of the Oralloy is 18.747 g/cc, and the isotopic enrichment (wt %) is 93.21% ^{235}U, 5.7697% ^{238}U, 0.9844% ^{234}U, and 0.0359% ^{236}U. The critical radius was 8.74268 cm. A photograph of the experiment is given in Fig. F17.D.11. The support structure was ignored in the input data.

INPUT DATA

```
=kenovi
sample problem 26  bare critical sphere   3.4420" radius
read parameters fdn=yes lib=4 end parameters
read mixt
mix=1    16092235    4.47708e-02
         16092238    2.73631e-03
         16092234    4.74857e-04
         16092236    1.71704e-05
end mixt
read geometry
global unit 1
sphere    10   8.74268
media   1 1 10
boundary 10
end geometry
end data
end
```

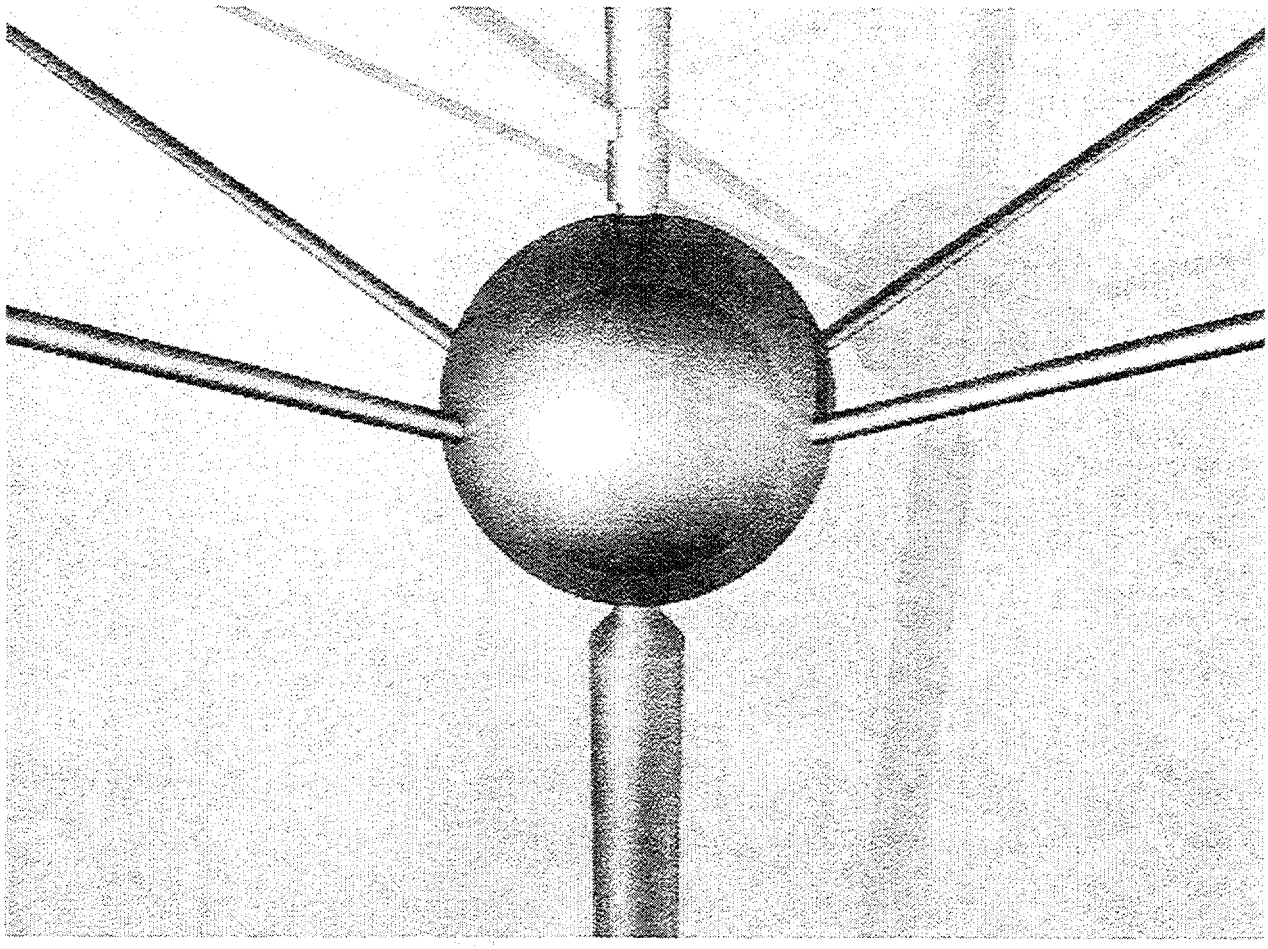


Figure F17.D.11 Critical oralloy sphere

SAMPLE PROBLEM 27 CRITICAL TRIANGULAR PITCHED ARRAY OF ANNULAR RODS

This sample problem represents a critical experiment[10,11] that consists of a partially flooded array of 19 low enriched uranium metal cylindrical annuli billets arranged in a triangular pitched array. The density of the uranium metal was 19.0 g/cc, and the isotopic enrichment in weight percent was 1.95% ^{235}U, 98.02% ^{238}U, 0.006% ^{236}U, and 0.002% ^{234}U. The cylindrical annuli had an inside diameter of 6.604 cm, an outside diameter of 18.288 cm, and were placed with a pitch of 20.828 cm. Each billet was 101.6 cm long. The array was positioned in a very large tank. This configuration was critical when the tank was filled to a height of 47.7 cm on a scale whose zero point was 0.6cm below the bottom of the billets. The bottom of the billets was 21.6 cm above the bottom of the tank. The tank and all support structures have been ignored in this model. The model utilizes only 15.24 cm of water reflector on all sides of the array. Figures F17.D.12 and F17.D.13 provide a representation of the model. A photograph of a single annular billet is shown in Fig. F17.D.14.

INPUT DATA

```
=kenovi
sample problem 27   critical triangular pitched array of annular rods
read parameters  fdn=yes nub=yes lib=4 end parameters
read mixt sct=2
mix=1    17092235   9.49269e-04
         17092238   4.71245e-02
         17092234   9.77784e-07
         17092236   2.90843e-06
mix=2    18008016   3.33757e-02
         18001001   6.67514e-02
mix=3    19001001   6.67514e-02
         19008016   3.33757e-02
mix=4    20092235   9.49269e-04
         20092238   4.71245e-02
         20092234   9.77784e-07
         20092236   2.90843e-06
end mixt
read geom
unit 1
cylinder 10  3.302 102.2 0.6
cylinder 20  9.144 102.2 0.6
plane    30  zpl=1.0 con=-47.7
hexprism 40 10.414 102.2 0.0
media  2 1 10 -30
media  1 1 20 -10 -30
media  3 1 40 -20 -30
media  0 1 10  30
media  4 1 20 -10  30
media  0 1 40 -20  30
boundary 40
unit 2
plane    10  zpl=1.0 con=-47.7
hexprism 20 10.414 102.2 0.0
media  3 1 -10  20
media  0 1  10  20
boundary 20
global unit 3
cylinder 10  52.07 102.2 0.0
plane    20  zpl=1.0 con=-47.7
cuboid   30 2p67.31 2p61.73 -21.0 102.2
array  1 10 place 4 4 1 3*0.0
media  0 1 30  20 -10
media  3 1 30 -20 -10
boundary 30
```

```
end geom
read array
ara=1 nux=7 nuy=7 nuz=1 typ=hex fill
2 2 2 2 2 2 2
 2 2 2 1 1 1 2
  2 2 1 1 1 1 2
   2 1 1 1 1 1 2
    2 1 1 1 1 2 2
     2 1 1 1 2 2 2
      2 2 2 2 2 2 2   end fill
end array
read plot scr=yes lpi=10
ttl='x-z plot of pins at y=0.0'
xul=-68.0 yul=0.0 zul=102.7 xlr=70.0 ylr=0.0 zlr=-25.0
uax=1.0 wdn=-1.0 nax=400  end plt0
ttl='x-y plot of pins and water at z=45.0'
xul=-68.0 yul=68.0 zul=45.0 xlr=68.0 ylr=-68.0 zlr=45.0
uax=1.0  vdn=-1.0 nax=400 end plt1
end plot
end data
end
```

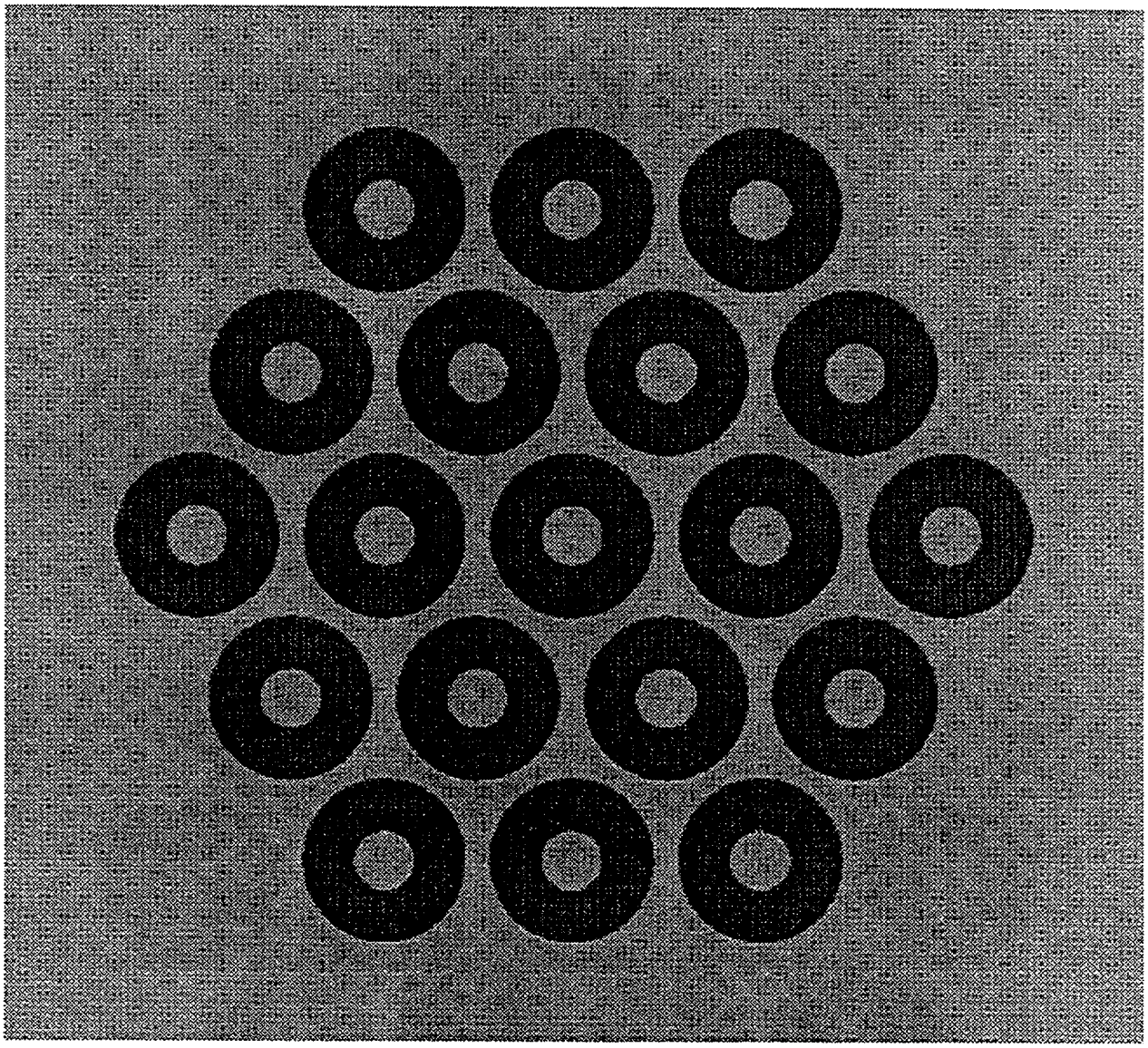


Figure F17.D.12 Horizontal slice through a critical triangular-pitched array of partially flooded 1.95% enriched uranium metal annular billets

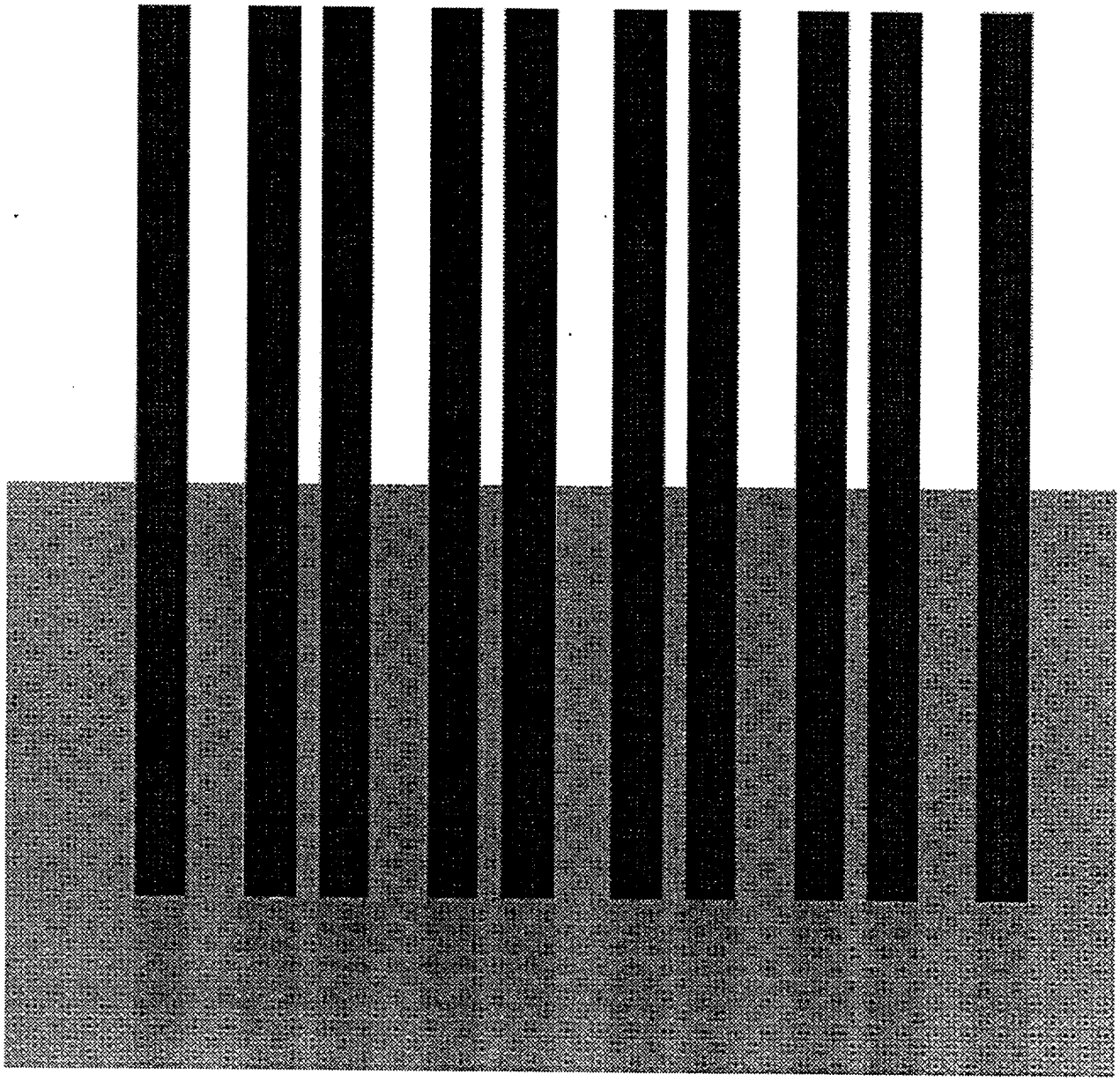


Figure F17.D.13 Vertical slice through the center of a critical triangular-pitched array of partially flooded 1.95% enriched uranium metal annular billets

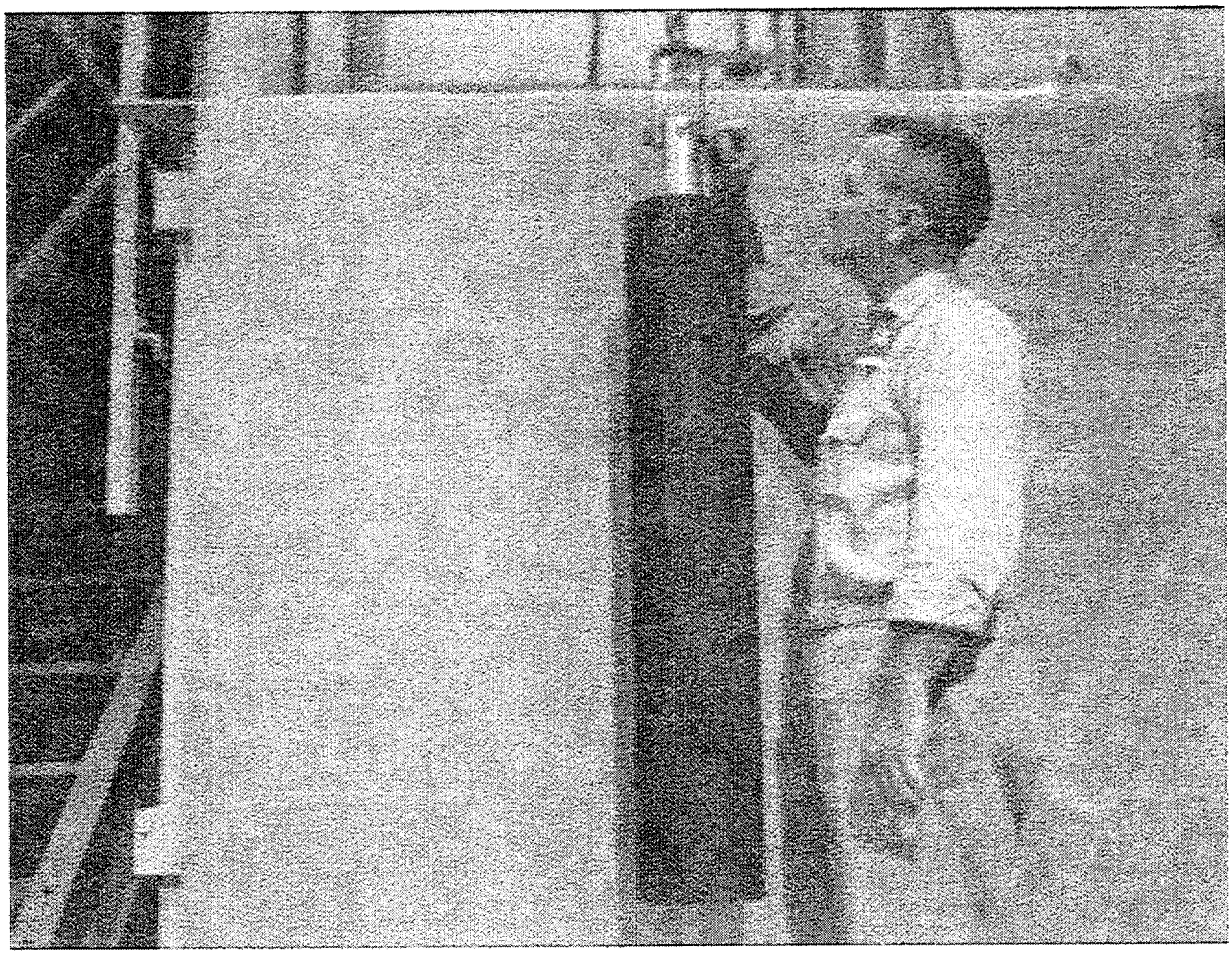


Figure F17.D.14 1.95% enriched uranium metal annular billet used in critical experiments

F17.D.2 SAMPLE PROBLEM WORKING FORMAT LIBRARY

The KENO VI sample problems use nuclide IDs that are consistent with the SCALE CSAS nuclide ID naming convention. The first time a nuclide is encountered, its ZA number is used. Nuclides are identified by the ZA number plus 1000000 times the mixture number. Either CSASN or the BONAMI/NITAWL explicit sequence can be used to create a problem-dependent working-format cross-section library suitable for use with the sample problems. CSASN or the BONAMI/NITAWL sequence can (1) be run alone with logical unit 4 of the job control language saved for later use with the KENO-VI sample problems, or (2) be placed in the job stream in front of the KENO VI sample problems

The CSASN SCALE control module is easy to use because it calculates the necessary resonance data required to create the problem-dependent AMPX working-format library. It is recommended that CSASN be used if it is operational at the installation where the problems are to be run. CSASN is one of the control modules associated with CSAS4 or CSAS6 and is automatically operational if CSAS4 or CSAS6 is operational. See Sect. C4 or C6 of the SCALE manual for assistance in using CSASN. If the SCALE DRIVER and the control modules are not operational at your installation, BONAMI and NITAWL should be used to create the problem-dependent AMPX working format cross-section library. Obtaining accurate resonance data for BONAMI and NITAWL may be difficult. See Sect. F1 and F2 of the SCALE manual for assistance in using BONAMI and NITAWL.

The KENO-VI sample problem input data are independent of energy group structure. However, if a cross-section library containing Bondarenko shielding factors is used, BONAMI must be run before NITAWL to provide the Bondarenko corrections necessary for the this library. If CSASN is used to create the AMPX working-format cross-section library, the Bondarenko corrections are automatically performed when necessary To use a different master library, simply supply the desired master cross-section library name in the CSASN data. See Sects. M7.5.3, M7.5.4, and C4.5.2 for additional information and examples. See Sect. M4 for information about the master format cross-section libraries that are available in SCALE.

Data for both CSASN and the BONAMI/NITAWL sequence are provided to create a problem-dependent AMPX working-format cross-section library suitable for use with the KENO-VI sample problems. These data, which include all of the mixtures used in the KENO-VI sample problems, will create an AMPX working-format cross-section library. The nuclide IDs from the BONAMI/NITAWL sequence match those in the KENO-VI sample problem mixing tables. *This cross-section library is problem-specific and is not appropriate for use with other problems.* None of the KENO-VI sample problems involve arrays of pins. Therefore, Dancoff corrections have not been incorporated into any of the nuclides in this library.

F17.D.2.1 CSASN Data

The CSASN input data to produce an AMPX working-format cross-section library for the KENO-VI sample problems are given on the following page.

INPUT DATA

```
=csasn
csasn to prepare  44 group working format xsec lib for keno-vi smp prbs
44groupndf5  latticecell
' 1 - uranium metal for smp prbs 1,2,3,4,5,6,7,8,9,10,11,12,19,22,23,24,25
uranium  1 den=18.76 1 293 92235 93.2 92238 5.6 92234 1.0 92236 0.2 end
' 2 - uranyl nitrate solution for smp prbs 12,18,19
solnuo2(no3)2  2 415 9.783-3 spg=1.555 1.0 293 92235 92.6 92238 5.9
                                              92234 1.0  92236 0.5 end
```

```
' 3 - uranium metal for smp prbs 13,14
uranium  3 den=18.69 1 293  92235 93.2 92238 5.6 92234 1.0 92236 0.2 end
' 4 - uranium metal for smp prb 15
uranium  4 den=18.794 1 293 92235 97.67 92238 0.98
                            92234 1.17  92236 0.20 end
' 5 - uranyl fluoride solution for smp prb 16
solnuo2f2      5 578.7 0 1.0 293 92235 93.2 92238 6.8 end
' 6 - borated uranyl fluoride solution for smp prb 16
solnuo2f2      6 578.7 0 1.0 293 92235 93.2 92238 6.8 end
boron          6 den=.0266 end
' 7 - uranyl fluoride solution for smp prb 17
solnuo2f2  7 133 0 1.0 293 92235 93.0 92238 7 end
' 8 - uranyl fluoride solution for smp prb 20
solnuo2f2  8 576.87 0 1.0 293 92235 93.2 92238 6.8 end
' 9 - uranyl fluoride solution for smp prb 21
solnuo2f2  9 494 0 spg= 1.56 1 293 92235 4.89 92238 95.09 92234 0.02 end
' 10 - paraffin for smp prbs 3,4,18
para(h2o) 10 end
' 11 - plexiglas for smp prbs 12,15,18,19
plexiglas 11 end
' 12 - water for smp prbs 15
h2o 12 end
' 13 - pyrex glass for smp prb 16
pyrex 13 end
' 14 - aluminum for smp prb 20, 21
al 14 end
' 15 - low density water for smp prb 18
h2o 15 1-9 end
' 16 - uranium metal for smp prbs 26
uranium  16 den=18.747 1 293  92235 93.21 92238 5.7697 92234 0.9844
                              92236 0.0359 end
' 17 - uranium metal for water moderated portion of smp prb 27
uranium  17 den=19.0 1 293  92235 1.95 92238 98.042 92234 0.002 92236 0.006  end
' 18 - internal (2nd) moderator water for smp prb 27
h2o      18 end
' 19 - external moderator water and reflector for smp prb 27
h2o      19 end
' 20 - uranium metal for bare portion of smp prb 27
uranium  20 den=19.0 1 293  92235 1.95 92238 98.042 92234 0.002 92236 0.006  end
end comp
atriangpitch  20.828 18.288 17 19 18 6.604 end
more data  res=20 slab 5.842 dan(20)=1.0   end more
end
```

F17.D.2.2 BONAMI Data

The BONAMI input data to produce an AMPX working-format cross-section library for the KENO-VI sample problems are given below. This input is only required if Bondarenko shielding factors are used for the referenced nuclides in the specified master cross-section library. If the SCALE DRIVER is not used, the first (=BONAMI) and last (END) records should be omitted from the BONAMI data.

INPUT DATA

```
=bonami
0$$ 84 a4 1
1$$ 2 20 79 0 0 0
2** .0001 1.35 1t
3$$ 1 2 3 4 5 6 7 8 9 16 17 20
    1 2 3 4 5 6 7 8 9 16 17 20
```

```
    1 2 3 4 9 16 17 20
    1 2 3 4 16 17 20
    2 2 5 6 7 8 9 11 12 13 15 18 19 2 5 6 7 8 9 10 11 12 15
    18 19 5 6 7 8 9 6 13 6 13 10 11 13 14 13 13
4$$ 92235 92235 92235 92235 92235 92235 92235 92235 92235 92235 92235
    92235
    92238 92238 92238 92238 92238 92238 92238 92238 92238 92238 92238
    92238
    92234 92234 92234 92234 92234 92234 92234 92234
    92236 92236 92236 92236 92236 92236 92236
     7014  8016  8016  8016  8016  8016  8016  8016  8016
     8016  8016  8016  8016  1001  1001  1001  1001  1001  1001  1001
     1001  1001  1001  1001  1001  9019  9019  9019  9019  9019  5010
     5010  5011  5011  6012  6012 13027 13027 14000 11023
5**   4.47971e-02 9.84601e-04 4.46299e-02 4.70211e-02 1.38188e-03
      1.38188e-03 3.16910e-04 1.37751e-03 6.18924e-05 4.47708e-02
      9.49269e-04 9.49269e-04
      2.65767e-03 6.19414e-05 2.64776e-03 4.65936e-04 9.95504e-05
      9.95504e-05 2.35521e-05 9.92356e-05 1.18834e-03 2.73631e-03
      4.71245e-02 4.71245e-02
      4.82717e-04 1.06784e-05 4.80915e-04 5.65802e-04 2.54224e-07
      4.74857e-04 9.77784e-07 9.77784e-07
      9.57232e-05 5.29386e-06 9.53660e-05 9.58967e-05 1.71704e-05
      2.90843e-06 2.90843e-06
      2.13092e-03 3.74130e-02 3.32040e-02 3.32040e-02 3.34202e-02
      3.32049e-02 3.32845e-02 1.42047e-02 3.33757e-02 4.49174e-02
      3.33757e-11 3.33757e-02 3.33757e-02 5.77964e-02 6.04824e-02
      6.04824e-02 6.54785e-02 6.05029e-02 6.15670e-02 8.26407e-02
      5.68187e-02 6.67514e-02 6.67514e-11 6.67514e-02 6.67514e-02
      2.96286e-03 2.96286e-03 6.80924e-04 2.95349e-03 2.50098e-03
      2.92804e-04 9.08242e-04 1.18870e-03 3.68719e-03 3.97311e-02
      3.55117e-02 4.97719e-04 6.02374e-02 1.80268e-02 2.39502e-03
10$$ 1092235  2092235  3092235  4092235  5092235  6092235  7092235
      8092235  9092235 16092235 17092235 20092235
      1092238  2092238  3092238  4092238  5092238  6092238  7092238
      8092238  9092238 16092238 17092238 20092238
      1092234  2092234  3092234  4092234  9092234 16092234 17092234
     20092234
      1092236  2092236  3092236  4092236 16092236 17092236 20092236
      2007014
      2008016  5008016  6008016  7008016  8008016  9008016 11008016
      12008016 13008016 15008016 18008016 19008016
      2001001  5001001  6001001  7001001  8001001  9001001 10001001
     11001001 12001001 15001001 18001001 19001001
      5009019  6009019  7009019  8009019  9009019
      6005010 13005010  6005011 13005011 10006012 11006012 13013027
     14013027
     13014000 13011023
6$$ 18 17 19 14i1 16 20
7** 3.302 9.144 10.9355 15i15.9355 95.9355
8** f293 9** f0 a2 .0952248 e
11$$ f0 2t
end
```

F17.D.2.3 NITAWL Data

The NITAWL input data to produce an AMPX working-format cross-section library for the KENO-VI sample problems are given below. If the SCALE DRIVER is not used, the first (=NITAWL) and last (END) records should be omitted from the NITAWL data.

```
=nitawl
0$$ 1 e  1$$ 2001 79 a8 40 2 e  1t
2$$
 1092235  2092235  3092235  4092235  5092235  6092235  7092235  8092235
 9092235 16092235 17092235 20092235
 1092238  2092238  3092238  4092238  5092238  6092238  7092238  8092238
 9092238 16092238 17092238 20092238
 1092234  2092234  3092234  4092234  9092234 16092234 17092234 20092234
 1092236  2092236  3092236  4092236 16092236 17092236 20092236
 2007014
 2008016  5008016  6008016  7008016  8008016  9008016 11008016 12008016
13008016 15008016 18008016 19008016
 2001001  5001001  6001001  7001001  8001001  9001001 10001001 11001001
12001001 15001001 18001001 19001001
 5009019  6009019  7009019  8009019  9009019
 6005010 13005010
 6005011 13005011
10006012 11006012
13013027 14013027
13014000
13011023
3**
 1092235 293.00    0 0.0 0.0 0.0 4.47971e-02  1 238.051 4.92413e-01
 1 234.370 1.35581e-01  1    1.0000
 2092235 293.00    0 0.0 0.0 0.0 9.84601e-04  1   1.008 1.19749e+03
 1  15.767 1.64828e+02  1    1.0000
 3092235 293.00    0 0.0 0.0 0.0 4.46299e-02  1 238.051 4.92413e-01
 1 234.370 1.35581e-01  1    1.0000
 4092235 293.00    0 0.0 0.0 0.0 4.70211e-02  1 234.041 1.26346e-01
 1 237.634 1.03660e-01  1    1.0000
 5092235 293.00    0 0.0 0.0 0.0 1.38188e-03  1   1.008 8.92870e+02
 1  16.294 9.84222e+01  1    1.0000
 6092235 293.00    0 0.0 0.0 0.0 1.38188e-03  1   1.008 8.92870e+02
 1  15.951 1.02996e+02  1    1.0000
 7092235 293.00    0 0.0 0.0 0.0 3.16910e-04  1   1.008 4.21496e+03
 1  16.068 4.03814e+02  1    1.0000
 8092235 293.00    0 0.0 0.0 0.0 1.37751e-03  1   1.008 8.96007e+02
 1  16.294 9.87105e+01  1    1.0000
 9092235 293.00    0 0.0 0.0 0.0 6.18924e-05  1   1.008 2.02927e+04
 1  17.317 2.32155e+03  1    1.0000
16092235 293.00    0 0.0 0.0 0.0 4.47708e-02  1 238.051 5.07281e-01
 1 234.110 1.15394e-01  1    1.0000
17092235 293.00    2 9.14400e+00 4.10601e-03 3.30200e+00 9.49269e-04
 1 238.051 4.12036e+02  1 235.538 4.29860e-02  1    1.0000
20092235 293.00    1 5.84200e+00 1.00000e+00 0.00000e+00 9.49269e-04
 1 238.051 4.12036e+02  1 235.538 4.29860e-02  1    1.0000
 1092238 293.00    0 0.0 0.0 0.0 2.65767e-03  1 235.044 1.76985e+02
 1 234.370 2.28531e+00  1    1.0000
 2092238 293.00    0 0.0 0.0 0.0 6.19414e-05  1   1.008 1.90349e+04
 1  16.689 2.77866e+03  1    1.0000
 3092238 293.00    0 0.0 0.0 0.0 2.64776e-03  1 235.044 1.76985e+02
 1 234.370 2.28531e+00  1    1.0000
 4092238 293.00    0 0.0 0.0 0.0 4.65936e-04  1 235.044 1.05963e+03
 1 234.329 1.49116e+01  1    1.0000
 5092238 293.00    0 0.0 0.0 0.0 9.95504e-05  1   1.008 1.23941e+04
 1  17.868 1.50367e+03  1    1.0000
```

```
 6092238 293.00    0 0.0 0.0 0.0 9.95504e-05  1    1.008 1.23941e+04
 1  17.427 1.56716e+03  1    1.0000
 7092238 293.00    0 0.0 0.0 0.0 2.35521e-05  1    1.008 5.67151e+04
 1  16.448 5.56657e+03  1    1.0000
 8092238 293.00    0 0.0 0.0 0.0 9.92356e-05  1    1.008 1.24377e+04
 1  17.862 1.50767e+03  1    1.0000
 9092238 293.00    0 0.0 0.0 0.0 1.18834e-03  1    1.008 1.05690e+03
 1  16.245 1.13160e+02  1    1.0000
16092238 293.00    0 0.0 0.0 0.0 2.73631e-03  1 235.044 1.71798e+02
 1 234.110 1.88805e+00  1    1.0000
17092238 293.00    2 9.14400e+00 4.10601e-03 3.30200e+00 4.71245e-02
 1 235.044 2.11511e-01  1 235.538 8.65905e-04  1    1.0000
20092238 293.00    1 5.84200e+00 1.00000e+00 0.00000e+00 4.71245e-02
 1 235.044 2.11511e-01  1 235.538 8.65905e-04  1    1.0000
 1092234 293.00    0 0.0 0.0 0.0 4.82717e-04  1 235.044 9.74421e+02
 1 237.963 4.77791e+01  1    1.0000
 2092234 293.00    0 0.0 0.0 0.0 1.06784e-05  1    1.008 1.10414e+05
 1  16.727 1.61556e+04  1    1.0000
 3092234 293.00    0 0.0 0.0 0.0 4.80915e-04  1 235.044 9.74422e+02
 1 237.963 4.77791e+01  1    1.0000
 4092234 293.00    0 0.0 0.0 0.0 5.65802e-04  1 235.044 8.72604e+02
 1 237.634 8.61464e+00  1    1.0000
 9092234 293.00    0 0.0 0.0 0.0 2.54224e-07  1    1.008 4.94039e+06
 1  17.392 5.67741e+05  1    1.0000
16092234 293.00    0 0.0 0.0 0.0 4.74857e-04  1 235.044 9.89970e+02
 1 238.035 4.82075e+01  1    1.0000
17092234 293.00    2 9.14400e+00 4.10601e-03 3.30200e+00 9.77784e-07
 1 238.051 4.00020e+05  1 235.047 1.02250e+04  1    1.0000
20092234 293.00    1 5.84200e+00 1.00000e+00 0.00000e+00 9.77784e-07
 1 238.051 4.00020e+05  1 235.047 1.02250e+04  1    1.0000
 1092236 293.00    0 0.0 0.0 0.0 9.57232e-05  1 235.044 4.91385e+03
 1 237.291 2.83392e+02  1    1.0000
 2092236 293.00    0 0.0 0.0 0.0 5.29386e-06  1    1.008 2.22720e+05
 1  16.732 3.25986e+04  1    1.0000
 3092236 293.00    0 0.0 0.0 0.0 9.53660e-05  1 235.044 4.91385e+03
 1 237.291 2.83392e+02  1    1.0000
 4092236 293.00    0 0.0 0.0 0.0 9.58967e-05  1 235.044 5.14847e+03
 1 235.606 1.02279e+02  1    1.0000
16092236 293.00    0 0.0 0.0 0.0 1.71704e-05  1 235.044 2.73781e+04
 1 237.319 1.61309e+03  1    1.0000
17092236 293.00    2 9.14400e+00 4.10601e-03 3.30200e+00 2.90843e-06
 1 238.051 1.34482e+05  1 235.043 3.43057e+03  1    1.0000
20092236 293.00    1 5.84200e+00 1.00000e+00 0.00000e+00 2.90843e-06
 1 238.051 1.34482e+05  1 235.043 3.43057e+03  1    1.0000
13011023 293.00    0 0.0 0.0 0.0 2.39502e-03  1  15.995 7.03292e+01
 1  18.466 2.41785e+01  1    1.0000
 3t
3$$
 1092235 14s 2092235  14s 3092235  14s 4092235 14s 5092235  14s
 6092235 14s 7092235  14s 8092235  14s 9092235 14s 16092235 14s
17092235 14s 20092235 14s
 1092238 14s 2092238  14s 3092238  14s 4092238 14s 5092238  14s
 6092238 14s 7092238  14s 8092238  14s 9092238 14s 16092238 14s
17092238 14s 20092238 14s
 1092234 14s 2092234  14s 3092234  14s 4092234 14s 9092234  14s
16092234 14s 17092234 14s 20092234 14s
 1092236 14s 2092236  14s 3092236  14s 4092236 14s 16092236 14s
17092236 14s 20092236 14s
13011023 14s e
end
```

F17.D.3 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, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab. (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, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 1963.

7. J. K. Fox, L. W. Gilley, D. Callihan, *Critical Mass Studies, Part IX Aqueous U-235 Solutions*, ORNL-2367, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., February 1958.

8. D. F. Cronin, *Critical Mass Studies, Part X Uranium of Intermediate Enrichment*, ORNL-2968, Union Carbide Corp., Nucl. Div., September 1960.

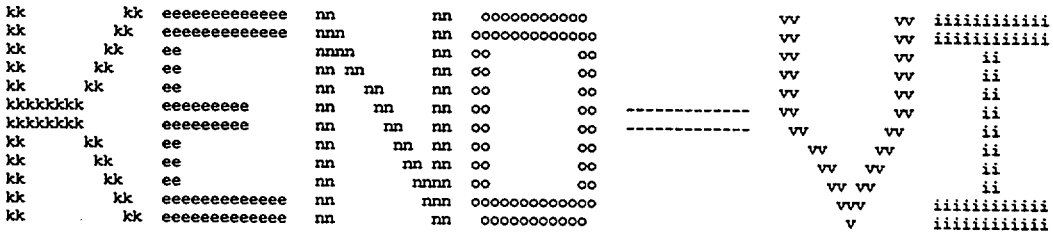
9. J. T. Mihalczo, J. J. Lynn, J. R.Taylor, and G. E. Hansen, "Measurements with an Unreflected Uranium (93.2%) Metal Sphere," *1993 Topical Meeting on Physics and Methods in Criticality Safety,* pp. 26–33, September 19–23, 1993.

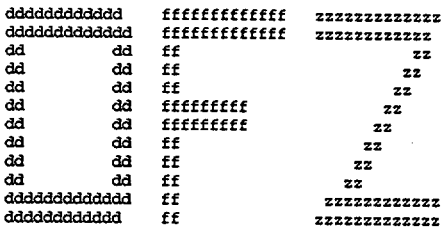
10. E. B. Johnson, "Critical Parameters of U(1.95) Metal Cylindrical Annuli," *Neutron Physics Division Annual Progress Report, September 1996,* ORNL-3973, Union Carbide Corp., Nucl. Div., Oak Ridge Natl. Lab., 1966.

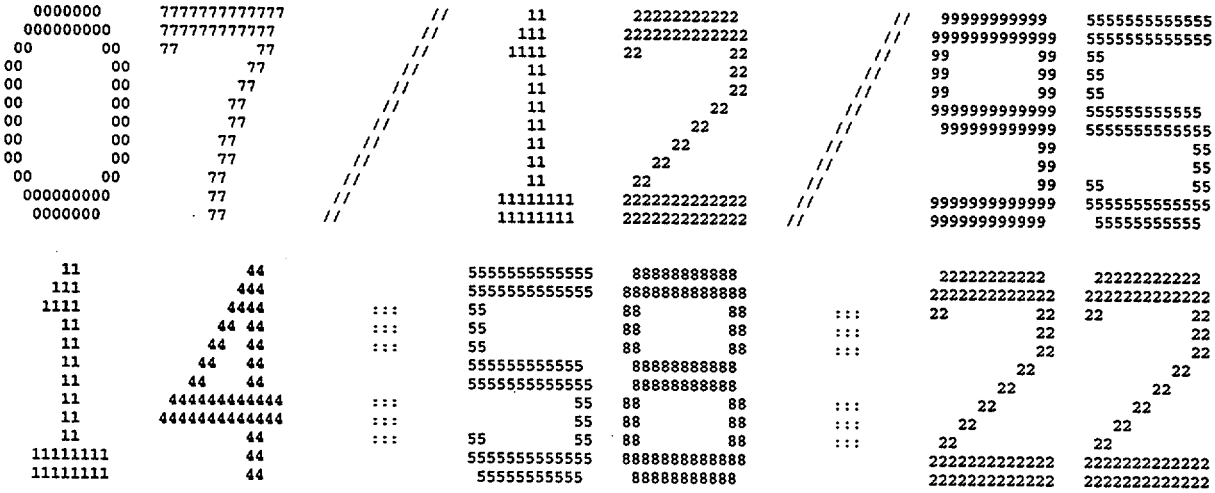
11. E. B. Johnson, "Critical Parameters of U(1.95) Metal Cylindrical Annuli," *Trans. Am. Nucl. Soc.* **9**, 185–186 (1966).

F17.E SELECTED OUTPUT FROM KENO-VI SAMPLE PROBLEMS

The computer output on the following pages is an example of the output for sample problems 4 and 19 described in Sect. F17.D.







```
 sssssssssss     ccccccccccc      aaaaaaaaa     ll              eeeeeeeeeeeee
ssssssssssssss   ccccccccccccc    aaaaaaaaaaa    ll              eeeeeeeeeeeee
ss          ss   cc          cc   aa          aa   ll              ee
ss               cc               aa          aa   ll              ee
ss               cc               aa          aa   ll              ee
ssssssssssss     cc               aaaaaaaaaaaaa   ll              eeeeeeeee
 ssssssssssss    cc               aaaaaaaaaaaaa   ll              eeeeeeeee
          ss     cc               aa          aa   ll              ee
          ss     cc               aa          aa   ll              ee
ss        ss     cc          cc   aa          aa   ll              ee
ssssssssssssss   ccccccccccccc   aa          aa   lllllllllllll   eeeeeeeeeeeee
 sssssssssss      ccccccccccc     aa          aa   lllllllllllll   eeeeeeeeeeeee
```

```
*******************************************************************************
*******************************************************************************
*******************************************************************************
*****                                                                     *****
*****                  program verification information                *****
*****                                                                     *****
*****                 code system:    scale  version:    4.2           *****
*****                                                                     *****
*******************************************************************************
*******************************************************************************
*****                                                                     *****
*****                                                                     *****
*****              program:  kenovi                                       *****
*****                                                                     *****
*****         creation date:  06/23/95                                    *****
*****                                                                     *****
*****              library:  /scale/exe                                   *****
*****                                                                     *****
*****                                                                     *****
*****       production code:  kenovi                                      *****
*****                                                                     *****
*****              version:  2.0                                          *****
*****                                                                     *****
*****              jobname:  dfz                                          *****
*****                                                                     *****
*****     date of execution:  07/12/95                                    *****
*****                                                                     *****
*****     time of execution:  14:58:22                                    *****
*****                                                                     *****
*****                                                                     *****
*******************************************************************************
*******************************************************************************
*******************************************************************************
```

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

****** numeric parameters ******

tme	maximum problem time (min)	30.00
tba	time per generation (min)	.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
nfb	fission bank size	300
xfb	extra positions in fission bank	0
sig	cut off standard deviation	.0000
wta	default value of weight average	.5000
wth	weight high for splitting	3.0000
wtl	weight low for russian roulette	.3333
rnd	starting random number	00000BB827100001
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

```
keno-vi sample problem 4  2c8  15.24 cm paraffin refl

******      logical parameters      ******

run  execute problem after checking data    yes     plt  plot map(s)                                yes
flx  compute flux                           yes     fdn  compute fission densities                  yes
smu  compute avg unit self-multiplication    no     nub  compute nu-bar & avg fission group         yes
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     gas  print far by group                        no
xs2  print 2-d mixture x-sections            no     pax  print xsec-albedo correlation tables      no
xap  print mixture angles & probabilities    no     pwt  print weight average array               yes
pki  print fission spectrum                  no     pgm  print input geometry                     no
p1d  print extra 1-d cross sections          no     bug  print debug information                  no
                                                    trk  print tracking information               no

                          parameter input completed

.....    0 io's were used reading the parameter data      .....

*************** data reading completed ***************
```

```
*****************************************************************************************************************
***                                                                                                           ***
***                          keno-vi sample problem 4  2c8  15.24 cm paraffin refl                            ***
***                                                                                                           ***
*****************************************************************************************************************
*****************************************************************************************************************
***                                                                                                           ***
***          unit                                                volume                                       ***
***         number              data set name                    name        unit function                    ***
***         ------              -------------                    ----        -------------                    ***
***                                                                                                           ***
***       xsc  14     ft14f001                                               mixed cross sections             ***
***                                                                                                           ***
***       alb  79     /scale/datalib/albedos                                 input albedos                    ***
***                                                                                                           ***
***       wts  80     /scale/datalib/weights                                 input weights                    ***
***                                                                                                           ***
***       skt  16      unknown                                               write scratch data               ***
***                                                                                                           ***
***       lib   4     ft04f001                                               input ampx working library       ***
***                                                                                                           ***
***             8     ft08f001                                               input data direct access         ***
***                                                                                                           ***
***             9      unknown                                               super grouped direct access      ***
***                                                                                                           ***
***            10      unknown                                               xsec mixing direct access        ***
***                                                                                                           ***
*****************************************************************************************************************

        ........   0 io's were used preparing input data    ........


        cross sections read from the ampx working library on unit    4
```

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

mixing table

number of scattering angles = 2
cross section message threshold =3.0E-05

mixture = 1 density(g/cc) = 18.760

nuclide	atom-dens.	wgt. frac.	za	awt	nuclide title		
92234	4.82716E-04	9.99999E-03	92234	234.0405	uranium-234	endf/b-iv mat 1043	updated 08/12/94
92235	4.47971E-02	9.32000E-01	92235	235.0441	uranium-235	endf/b-iv mat 1261	updated 08/12/94
92236	9.57231E-05	2.00000E-03	92236	236.0458	u-236 1163 sigo=5+4 newxlacs p-3 293k f-1/e-m(1.+5)		updated 08/12/94
92238	2.65767E-03	5.60000E-02	92238	238.0510	uranium-238	endf/b-iv mat 1262	updated 08/12/94

mixture = 2 density(g/cc) = .92999

nuclide	atom-dens.	wgt. frac.	za	awt	nuclide title		
1001001	8.26407E-02	1.48689E-01	1001	1.0077	hydrogen	endf/b-iv mat 1269/thrm1002	updated 08/12/94
6012	3.97311E-02	8.51311E-01	6000	12.0001	carbon-12	endf/b-iv mat 1274/thrm1065	updated 08/12/94

1001001	hydrogen	endf/b-iv mat 1269/thrm1002	updated 08/12/94
6012	carbon-12	endf/b-iv mat 1274/thrm1065	updated 08/12/94
92234	uranium-234	endf/b-iv mat 1043	updated 08/12/94
92235	uranium-235	endf/b-iv mat 1261	updated 08/12/94
92236	u-236 1163 sigo=5+4 newxlacs p-3 293k f-1/e-m(1.+5)		updated 08/12/94
92238	uranium-238	endf/b-iv mat 1262	updated 08/12/94

0keno message number k6-222 2 transfers for mixture 2 were corrected for bad moments.

..... 0 io's were used mixing cross-sections .....

1-d cross section array id numbers
1 2002 1452 27 18 1018

..... 0 io's were used preparing the cross sections .....

```
keno-vi sample problem 4  2c8  15.24 cm paraffin refl

                          ****** additional information ******

number of energy groups                27      use lattice geometry                    yes
no. of fission spectrum source group    1      global array number                       1
no. of scattering angles in xsecs       2      number of units in the global x dir.      2
entries/neutron in the neutron bank    24      number of units in the global y dir.      2
entries/neutron in the fission bank    14      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 differential albedos used     0      number of holes                           0
total input geometry regions            9      maximum hole nesting level                0
number of geometry regions used         9      use nested arrays                        no
largest geometry unit number            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
```

```
keno-vi sample problem 4  2c8  15.24 cm paraffin refl

                  ****** space and supergroup information ******

       20000 words is the total space available.

       14083 words were used for non-supergroup storage.

        5917 words of storage are available for supergrouped data.

       19880 words of storage are available for constructing the supergroups.

        5857 words of storage are available to each supergroup.

         340 words are needed for the largest group.

       14639 words of storage is sufficient to run this problem.

       18799 words of storage will allow the problem to run with one supergroup.

       18976 words of storage will be used to run this problem.
```

supergroup	starting group	ending group	xsec length	albedo length	total length
1	1	27	678	0	4656

```
.....   0 io's were used in supergrouping    .....

.....   0 io's were used loading the data    .....
```

****** geometry parameters ******

niar	number of independent array references	1
ngblu	global unit number	2
nboxt	number of units in the problem	2
nquad	number of quadratics in the problem	26
ngwrds	number of geometry words read	9
maxgwd	maximum geometry words in a unit	7
maxsfu	largest number of surfaces in a unit	21
maxreg	largest number of media in a unit	7
regtot	number of spatial volumes defined	9
sectot	number of entries in the sector array	53
nucom	number of comments in the geometry data	2
numhol	number of holes in the problem	0

```
                    keno-vi sample problem 4  2c8  15.24 cm paraffin refl
                 geometry description for those units utilized in this problem

                                                  -----  unit    1  -----
single 2c8 unit centered
  1           cylinder          10

                                                  quadratic surfaces
   -1.000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z    33.040
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z    28.971
  2           cuboid            20

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   137.828
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   137.828
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   129.391
                    sector
             imp      definitions
media    1    1        10
media    0    1        20  -10
boundary               20

                                  ********************  global  ********************
                                                  -----  unit    2  -----

2x2x2  2c8 array with reflector
  1           cuboid            10

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   551.310
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   551.310
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   517.562
  2           cuboid            20

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   701.190
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   701.190
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   663.062
  3           cuboid            30

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   869.070
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   869.070
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z   826.562
  4           cuboid            40

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1054.950
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1054.950
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1008.062
  5           cuboid            50

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1258.830
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1258.830
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1207.562
  6           cuboid            60

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1480.710
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1480.710
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1425.062
  7           cuboid            70

                                                  quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1499.238
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1499.238
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z  1443.240
                    sector
             imp      definitions
array    1            10
media    2    2       -10   20
media    2    3       -20   30
media    2    4       -30   40
media    2    5       -40   50
media    2    6        60  -50
media    2    7        70  -60
boundary               70
                     keno-vi sample problem 4  2c8  15.24 cm paraffin refl
                -------  unit orientation description for array   1        -------
z layer   1, x column   1 to   2 left to right   y row   1 to   2  bottom to top
 1 1
 1 1
-z layer   2, x column   1 to   2 left to right   y row   1 to   2  bottom to top
 1 1
 1 1
```

keno-vi sample problem 4 2c8 15.24 cm paraffin refl
volumes for those units utilized in this problem

unit	region	geometry region	volume	cumulative volume
1	1	1	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	2	1.00000E+00 cm**3	1.00000E+00 cm**3
2	1	3	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	4	1.00000E+00 cm**3	1.00000E+00 cm**3
	3	5	1.00000E+00 cm**3	1.00000E+00 cm**3
	4	6	1.00000E+00 cm**3	1.00000E+00 cm**3
	5	7	1.00000E+00 cm**3	1.00000E+00 cm**3
	6	8	1.00000E+00 cm**3	1.00000E+00 cm**3
	7	9	1.00000E+00 cm**3	1.00000E+00 cm**3

unit	uses	region	mixture	total volume
1	8	1	1	8.00000E+00 cm**3
		2	0	8.00000E+00 cm**3
2	1	1		1.00000E+00 cm**3
		2	2	1.00000E+00 cm**3
		3	2	1.00000E+00 cm**3
		4	2	1.00000E+00 cm**3
		5	2	1.00000E+00 cm**3
		6	2	1.00000E+00 cm**3
		7	2	1.00000E+00 cm**3

total mixture volumes

mixture	total volume
0	8.00000E+00 cm**3
1	8.00000E+00 cm**3
2	6.00000E+00 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 .500 will be used for all other bias id's.

keno-vi sample problem 4 2c8 15.24 cm paraffin 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	6.18390E-01	8.85090E-01	1.25430E+00	1.80190E+00	2.61220E+00	3.80230E+00
2	5.00000E-01	5.91790E-01	8.52020E-01	1.33350E+00	2.23270E+00	3.89200E+00	6.93920E+00
3	5.00000E-01	5.90830E-01	8.85190E-01	1.52270E+00	2.90060E+00	5.87640E+00	1.23420E+01
4	5.00000E-01	5.86770E-01	9.05420E-01	1.68610E+00	3.57900E+00	8.22490E+00	1.98220E+01
5	5.00000E-01	5.81890E-01	9.30070E-01	1.88130E+00	4.43390E+00	1.14320E+01	3.10120E+01
6	5.00000E-01	5.74820E-01	9.87290E-01	2.28780E+00	6.30330E+00	1.90190E+01	5.96620E+01
7	5.00000E-01	5.87290E-01	1.15630E+00	3.20150E+00	1.04420E+01	3.60420E+01	1.23200E+02
8	5.00000E-01	6.20770E-01	1.41080E+00	4.44510E+00	1.57830E+01	5.64360E+01	1.91970E+02
9	5.00000E-01	6.57610E-01	1.64320E+00	5.50540E+00	2.00300E+01	7.15900E+01	2.40440E+02
10	5.00000E-01	6.87950E-01	1.83600E+00	6.38830E+00	2.34660E+01	8.35140E+01	2.77740E+02
11	5.00000E-01	6.92740E-01	1.94220E+00	6.96040E+00	2.56820E+01	9.09270E+01	2.99680E+02
12	5.00000E-01	6.98060E-01	2.06680E+00	7.59760E+00	2.80730E+01	9.87930E+01	3.22680E+02
13	5.00000E-01	7.01860E-01	2.17810E+00	8.12680E+00	3.00000E+01	1.05060E+02	3.40830E+02
14	5.00000E-01	7.12050E-01	2.33100E+00	8.78950E+00	3.23810E+01	1.12840E+02	3.63710E+02
15	5.00000E-01	7.34300E-01	2.53080E+00	9.60390E+00	3.53030E+01	1.22520E+02	3.92830E+02
16	5.00000E-01	7.84110E-01	2.80270E+00	1.06540E+01	3.91200E+01	1.35510E+02	4.33480E+02
17	5.00000E-01	7.83000E-01	2.82230E+00	1.07360E+01	3.93890E+01	1.36290E+02	4.35350E+02
18	5.00000E-01	8.11260E-01	2.96800E+00	1.12930E+01	4.14180E+01	1.43220E+02	4.57120E+02
19	5.00000E-01	8.32890E-01	3.08460E+00	1.17400E+01	4.30330E+01	1.48700E+02	4.74160E+02
20	5.00000E-01	8.41560E-01	3.14930E+00	1.19880E+01	4.38940E+01	1.51420E+02	4.81790E+02
21	5.00000E-01	8.37120E-01	3.15740E+00	1.20190E+01	4.39320E+01	1.51180E+02	4.79580E+02
22	5.00000E-01	8.36030E-01	3.17070E+00	1.20670E+01	4.40680E+01	1.51460E+02	4.79740E+02
23	5.00000E-01	8.61360E-01	3.32330E+00	1.26400E+01	4.60960E+01	1.58130E+02	4.99640E+02
24	5.00000E-01	8.86690E-01	3.46960E+00	1.31820E+01	4.79880E+01	1.64220E+02	5.17330E+02
25	5.00000E-01	9.02200E-01	3.55250E+00	1.34830E+01	4.90100E+01	1.67360E+02	5.25870E+02
26	5.00000E-01	9.20350E-01	3.64340E+00	1.38170E+01	5.01620E+01	1.71010E+02	5.36250E+02
27	5.00000E-01	9.43470E-01	3.74870E+00	1.42020E+01	5.14930E+01	1.75220E+02	5.48200E+02

..... 0 io's were used in keno-v before tracking .....

..... .00000minutes were used processing data. .....

volume fraction of fissile material in the system= .36364E+00

start type 0 was used.

the neutrons were started with a flat distribution in a cuboid defined by:
+x= 3.87200E+01 -x=-3.87200E+01 +y= 3.87200E+01 -y=-3.87200E+01 +z= 3.79900E+01 -z=-3.79900E+01

.02133 minutes were required for starting. total elapsed time is .02133 minutes.

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

generation	generation k-effective	elapsed time minutes	average k-effective	avg k-eff deviation	matrix k-effective	matrix k-eff deviation
1	1.05232E+00	6.40000E-02	1.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	1.03003E+00	1.28000E-01	1.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	1.02401E+00	1.92000E-01	1.02401E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	9.74890E-01	2.13333E-01	9.99448E-01	2.45582E-02	0.00000E+00	0.00000E+00
5	1.02300E+00	2.77333E-01	1.00730E+00	1.62072E-02	0.00000E+00	0.00000E+00
6	1.02899E+00	3.41333E-01	1.01272E+00	1.26783E-02	0.00000E+00	0.00000E+00
7	9.79124E-01	3.84000E-01	1.00600E+00	1.18994E-02	0.00000E+00	0.00000E+00
8	1.03230E+00	4.48000E-01	1.01038E+00	1.06586E-02	0.00000E+00	0.00000E+00
9	1.06259E+00	5.12000E-01	1.01784E+00	1.16950E-02	0.00000E+00	0.00000E+00
10	1.01395E+00	5.76000E-01	1.01736E+00	1.01399E-02	0.00000E+00	0.00000E+00
11	9.02205E-01	6.18667E-01	1.00456E+00	1.56100E-02	0.00000E+00	0.00000E+00
12	1.00224E+00	6.61333E-01	1.00433E+00	1.39639E-02	0.00000E+00	0.00000E+00
13	9.62315E-01	7.25333E-01	1.00051E+00	1.31957E-02	0.00000E+00	0.00000E+00
14	9.77535E-01	7.89333E-01	9.98596E-01	1.21972E-02	0.00000E+00	0.00000E+00
15	9.46200E-01	8.32000E-01	9.94565E-01	1.19217E-02	0.00000E+00	0.00000E+00
16	1.09511E+00	8.96000E-01	1.00175E+00	1.31684E-02	0.00000E+00	0.00000E+00
17	1.00411E+00	9.60000E-01	1.00190E+00	1.22601E-02	0.00000E+00	0.00000E+00
18	1.01688E+00	1.00267E+00	1.00284E+00	1.15064E-02	0.00000E+00	0.00000E+00
19	9.38776E-01	1.04533E+00	9.99072E-01	1.14465E-02	0.00000E+00	0.00000E+00
20	1.05753E+00	1.10933E+00	1.00232E+00	1.12699E-02	0.00000E+00	0.00000E+00
21	1.00952E+00	1.15200E+00	1.00270E+00	1.06670E-02	0.00000E+00	0.00000E+00
22	9.86390E-01	1.19467E+00	1.00188E+00	1.01524E-02	0.00000E+00	0.00000E+00
23	9.47884E-01	1.25867E+00	9.99312E-01	9.99337E-03	0.00000E+00	0.00000E+00
24	9.20649E-01	1.32267E+00	9.95736E-01	1.01771E-02	0.00000E+00	0.00000E+00
25	9.98392E-01	1.36533E+00	9.95852E-01	9.72524E-03	0.00000E+00	0.00000E+00
26	9.61672E-01	1.40800E+00	9.94427E-01	9.41950E-03	0.00000E+00	0.00000E+00
27	9.72450E-01	1.47200E+00	9.93548E-01	9.07753E-03	0.00000E+00	0.00000E+00
28	1.04718E+00	1.51467E+00	9.95611E-01	8.96207E-03	0.00000E+00	0.00000E+00
29	9.61118E-01	1.55733E+00	9.94334E-01	8.71787E-03	0.00000E+00	0.00000E+00
30	1.01641E+00	1.62133E+00	9.95122E-01	8.43767E-03	0.00000E+00	0.00000E+00
31	9.53034E-01	1.66400E+00	9.93671E-01	8.26986E-03	0.00000E+00	0.00000E+00
32	1.17266E+00	1.70667E+00	9.99637E-01	9.97139E-03	0.00000E+00	0.00000E+00
33	1.00927E+00	1.77067E+00	9.99948E-01	9.64937E-03	0.00000E+00	0.00000E+00
34	8.86708E-01	1.81333E+00	9.96409E-01	9.99069E-03	0.00000E+00	0.00000E+00
35	1.07450E+00	1.87733E+00	9.98776E-01	9.96814E-03	0.00000E+00	0.00000E+00
36	9.36499E-01	1.92000E+00	9.96944E-01	9.84245E-03	0.00000E+00	0.00000E+00
37	9.47681E-01	1.96267E+00	9.95536E-01	9.66019E-03	0.00000E+00	0.00000E+00
38	9.69118E-01	2.02667E+00	9.94803E-01	9.41666E-03	0.00000E+00	0.00000E+00
39	1.04915E+00	2.09067E+00	9.96271E-01	9.27565E-03	0.00000E+00	0.00000E+00
40	1.06399E+00	2.13333E+00	9.98053E-01	9.20244E-03	0.00000E+00	0.00000E+00
41	1.04602E+00	2.19733E+00	9.99283E-01	9.04735E-03	0.00000E+00	0.00000E+00
42	9.65354E-01	2.24000E+00	9.98435E-01	8.85896E-03	0.00000E+00	0.00000E+00
43	9.73261E-01	2.30400E+00	9.97821E-01	8.66198E-03	0.00000E+00	0.00000E+00
44	1.06292E+00	2.34667E+00	9.99371E-01	8.59413E-03	0.00000E+00	0.00000E+00
45	9.55318E-01	2.38933E+00	9.98346E-01	8.45420E-03	0.00000E+00	0.00000E+00
46	1.06056E+00	2.47467E+00	9.99760E-01	8.37995E-03	0.00000E+00	0.00000E+00
47	1.03734E+00	2.53867E+00	1.00060E+00	8.23408E-03	0.00000E+00	0.00000E+00
48	1.06522E+00	2.58133E+00	1.00200E+00	8.17470E-03	0.00000E+00	0.00000E+00
49	9.89311E-01	2.62400E+00	1.00173E+00	8.00344E-03	0.00000E+00	0.00000E+00
50	1.07420E+00	2.68800E+00	1.00324E+00	7.97905E-03	0.00000E+00	0.00000E+00
51	1.01924E+00	2.75200E+00	1.00357E+00	7.82134E-03	0.00000E+00	0.00000E+00
52	1.06604E+00	2.79467E+00	1.00482E+00	7.76451E-03	0.00000E+00	0.00000E+00
53	1.04927E+00	2.85867E+00	1.00569E+00	7.66050E-03	0.00000E+00	0.00000E+00
54	9.90373E-01	2.94400E+00	1.00539E+00	7.51751E-03	0.00000E+00	0.00000E+00
55	9.96628E-01	2.98667E+00	1.00523E+00	7.37616E-03	0.00000E+00	0.00000E+00
56	9.35001E-01	3.05067E+00	1.00393E+00	7.35418E-03	0.00000E+00	0.00000E+00
57	9.95944E-01	3.09333E+00	1.00378E+00	7.22069E-03	0.00000E+00	0.00000E+00
58	1.10795E+00	3.15733E+00	1.00564E+00	7.33053E-03	0.00000E+00	0.00000E+00
59	1.05006E+00	3.20000E+00	1.00642E+00	7.24282E-03	0.00000E+00	0.00000E+00
60	9.19054E-01	3.24267E+00	1.00492E+00	7.27452E-03	0.00000E+00	0.00000E+00
61	9.65863E-01	3.30667E+00	1.00425E+00	7.18073E-03	0.00000E+00	0.00000E+00
62	9.52853E-01	3.37067E+00	1.00340E+00	7.11182E-03	0.00000E+00	0.00000E+00
63	9.70607E-01	3.41333E+00	1.00286E+00	7.01489E-03	0.00000E+00	0.00000E+00
64	1.08541E+00	3.47733E+00	1.00419E+00	7.02809E-03	0.00000E+00	0.00000E+00
65	1.01429E+00	3.54133E+00	1.00435E+00	6.91749E-03	0.00000E+00	0.00000E+00
66	9.28092E-01	3.60533E+00	1.00316E+00	6.91202E-03	0.00000E+00	0.00000E+00
67	1.01887E+00	3.66933E+00	1.00340E+00	6.80914E-03	0.00000E+00	0.00000E+00
68	1.02995E+00	3.71200E+00	1.00380E+00	6.71723E-03	0.00000E+00	0.00000E+00
69	1.01593E+00	3.75467E+00	1.00398E+00	6.61869E-03	0.00000E+00	0.00000E+00
70	1.08576E+00	3.81867E+00	1.00519E+00	6.63061E-03	0.00000E+00	0.00000E+00
71	9.36529E-01	3.86133E+00	1.00419E+00	6.60914E-03	0.00000E+00	0.00000E+00
72	1.03625E+00	3.92533E+00	1.00465E+00	6.53012E-03	0.00000E+00	0.00000E+00
73	9.84095E-01	3.98933E+00	1.00436E+00	6.44400E-03	0.00000E+00	0.00000E+00
74	9.38039E-01	4.01067E+00	1.00344E+00	6.42029E-03	0.00000E+00	0.00000E+00
75	1.00237E+00	4.07467E+00	1.00342E+00	6.33175E-03	0.00000E+00	0.00000E+00
76	1.04787E+00	4.13867E+00	1.00403E+00	6.27441E-03	0.00000E+00	0.00000E+00
77	9.60502E-01	4.18133E+00	1.00344E+00	6.21733E-03	0.00000E+00	0.00000E+00
78	9.76534E-01	4.22400E+00	1.00309E+00	6.14518E-03	0.00000E+00	0.00000E+00
79	9.30093E-01	4.30933E+00	1.00214E+00	6.13850E-03	0.00000E+00	0.00000E+00
80	9.69117E-01	4.35200E+00	1.00172E+00	6.07406E-03	0.00000E+00	0.00000E+00
81	1.04409E+00	4.39467E+00	1.00226E+00	6.02062E-03	0.00000E+00	0.00000E+00
82	8.84657E-01	4.45867E+00	1.00079E+00	6.12393E-03	0.00000E+00	0.00000E+00
83	1.10813E+00	4.50133E+00	1.00211E+00	6.19136E-03	0.00000E+00	0.00000E+00
84	9.57290E-01	4.54400E+00	1.00156E+00	6.13977E-03	0.00000E+00	0.00000E+00
85	9.60983E-01	4.60800E+00	1.00108E+00	6.08502E-03	0.00000E+00	0.00000E+00

86	1.16050E+00	4.67200E+00	1.00297E+00	6.30458E-03	0.00000E+00	0.00000E+00
87	9.75571E-01	4.71467E+00	1.00265E+00	6.23831E-03	0.00000E+00	0.00000E+00
88	1.02697E+00	4.75733E+00	1.00293E+00	6.17182E-03	0.00000E+00	0.00000E+00
89	9.38932E-01	4.82133E+00	1.00220E+00	6.14467E-03	0.00000E+00	0.00000E+00
90	1.03860E+00	4.86400E+00	1.00261E+00	6.08851E-03	0.00000E+00	0.00000E+00
91	1.04515E+00	4.90667E+00	1.00309E+00	6.03865E-03	0.00000E+00	0.00000E+00
92	9.97060E-01	4.97067E+00	1.00302E+00	5.97156E-03	0.00000E+00	0.00000E+00
93	1.04236E+00	5.01333E+00	1.00345E+00	5.92137E-03	0.00000E+00	0.00000E+00
94	1.01219E+00	5.07733E+00	1.00355E+00	5.85743E-03	0.00000E+00	0.00000E+00
95	1.03387E+00	5.14133E+00	1.00388E+00	5.80327E-03	0.00000E+00	0.00000E+00
96	9.71395E-01	5.16267E+00	1.00353E+00	5.75159E-03	0.00000E+00	0.00000E+00
97	9.62189E-01	5.22667E+00	1.00310E+00	5.70734E-03	0.00000E+00	0.00000E+00
98	1.07200E+00	5.29067E+00	1.00381E+00	5.69300E-03	0.00000E+00	0.00000E+00
99	1.02765E+00	5.33333E+00	1.00406E+00	5.63936E-03	0.00000E+00	0.00000E+00
100	1.07155E+00	5.37600E+00	1.00475E+00	5.62384E-03	0.00000E+00	0.00000E+00
101	1.05471E+00	5.44000E+00	1.00525E+00	5.58958E-03	0.00000E+00	0.00000E+00
102	1.01308E+00	5.52533E+00	1.00533E+00	5.53395E-03	0.00000E+00	0.00000E+00
103	9.21325E-01	5.56800E+00	1.00450E+00	5.54166E-03	0.00000E+00	0.00000E+00

keno message number k6-123 execution terminated due to completion of the specified number of generations.

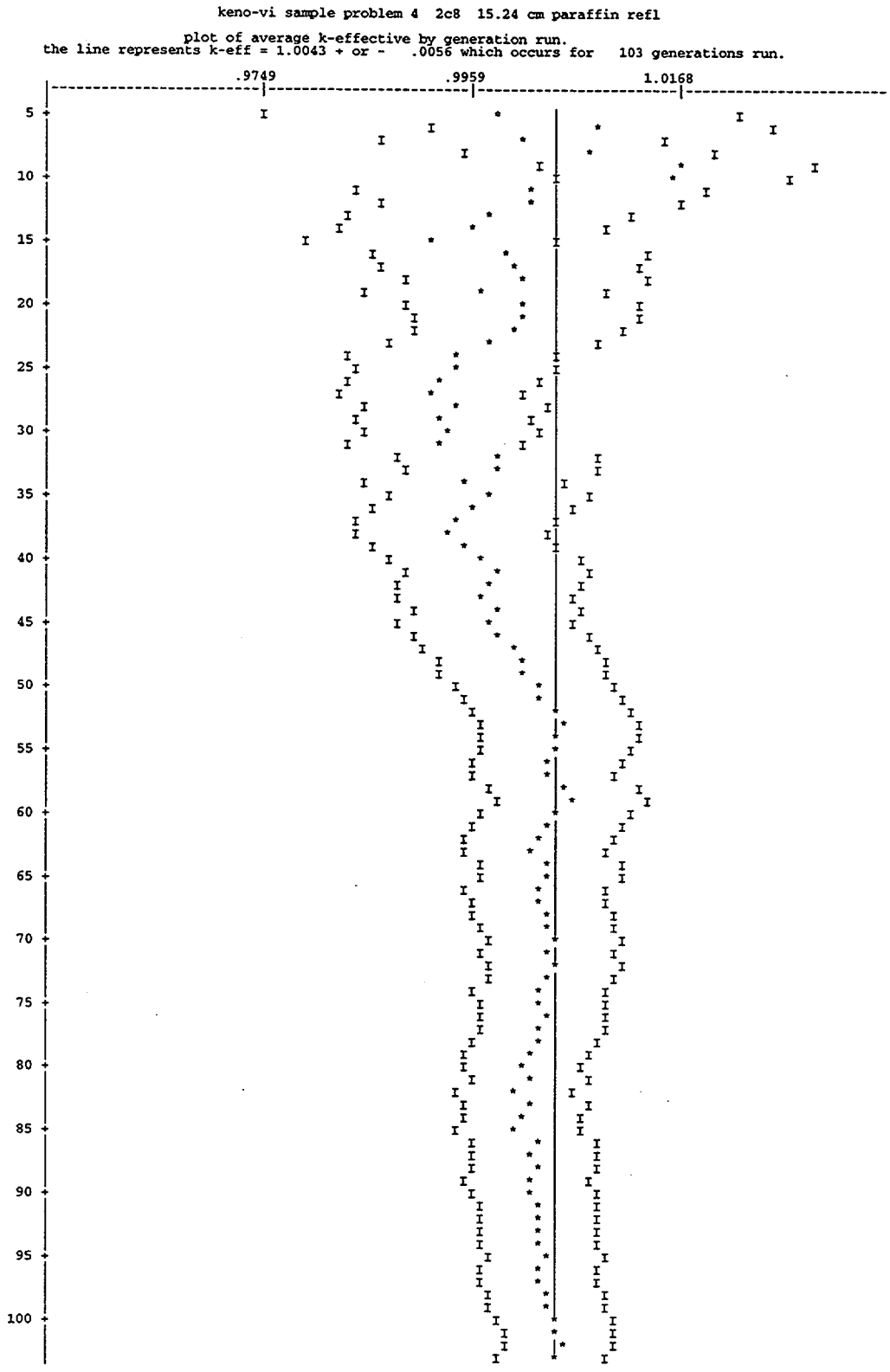
keno-vi sample problem 4 2c8 15.24 cm paraffin refl

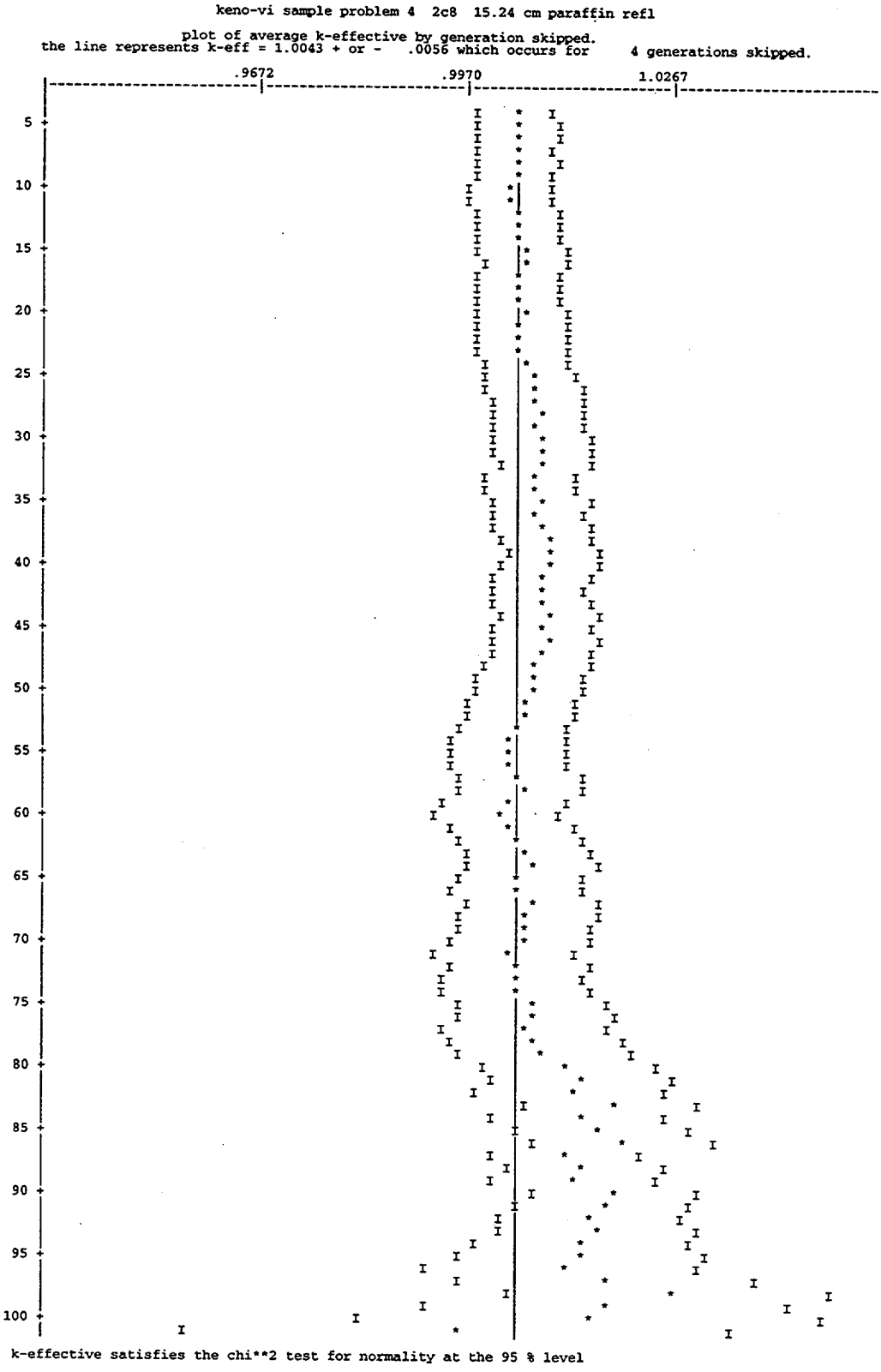
lifetime = 2.68453E-04 + or - 2.04323E-05 generation time = 7.47191E-05 + or - 1.83147E-06
nu bar = 2.56302E+00 + or - 1.26827E-03 average fission group = 9.64538E+00 + or - 6.51398E-02
energy(ev) of the average lethargy causing fission = 1.19057E+04 + or - 6.65147E+02

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	deviation of variance (per cent)
3	1.00430	+ or - .00559	.99871 to 1.00990	.99312 to 1.01549	.98752 to 1.02109	30000	14.5545
4	1.00460	+ or - .00564	.99896 to 1.01024	.99332 to 1.01589	.98767 to 1.02153	29700	14.5554
5	1.00441	+ or - .00570	.99872 to 1.01011	.99302 to 1.01581	.98732 to 1.02151	29400	14.5600
6	1.00416	+ or - .00575	.99841 to 1.00991	.99266 to 1.01566	.98691 to 1.02141	29100	14.5881
7	1.00442	+ or - .00581	.99862 to 1.01023	.99281 to 1.01603	.98701 to 1.02184	28800	14.5742
8	1.00413	+ or - .00586	.99827 to 1.00999	.99241 to 1.01584	.98655 to 1.02170	28500	14.6166
9	1.00350	+ or - .00589	.99762 to 1.00939	.99173 to 1.01528	.98584 to 1.02117	28200	14.8415
10	1.00339	+ or - .00595	.99744 to 1.00934	.99149 to 1.01529	.98554 to 1.02124	27900	14.8228
11	1.00449	+ or - .00591	.99858 to 1.01040	.99267 to 1.01632	.98676 to 1.02223	27600	15.0389
12	1.00452	+ or - .00598	.99854 to 1.01049	.99256 to 1.01647	.98659 to 1.02245	27300	14.9985
17	1.00495	+ or - .00617	.99878 to 1.01112	.99261 to 1.01730	.98643 to 1.02347	25800	15.5482
22	1.00514	+ or - .00646	.99868 to 1.01161	.99221 to 1.01807	.98575 to 1.02454	24300	15.8499
27	1.00810	+ or - .00671	1.00139 to 1.01481	.99468 to 1.02152	.98797 to 1.02823	22800	16.3120
32	1.00655	+ or - .00669	.99986 to 1.01325	.99317 to 1.01994	.98647 to 1.02663	21300	15.2633
37	1.00925	+ or - .00674	1.00251 to 1.01599	.99577 to 1.02273	.98903 to 1.02947	19800	16.1417
42	1.00847	+ or - .00712	1.00136 to 1.01559	.99424 to 1.02271	.98713 to 1.02982	18300	16.9725
47	1.00764	+ or - .00753	1.00010 to 1.01517	.99257 to 1.02270	.98504 to 1.03023	16800	18.0166
52	1.00419	+ or - .00798	.99621 to 1.01217	.98822 to 1.02015	.98024 to 1.02813	15300	19.5567
57	1.00536	+ or - .00866	.99669 to 1.01402	.98803 to 1.02268	.97937 to 1.03134	13800	20.0210
62	1.00611	+ or - .00894	.99718 to 1.01505	.98824 to 1.02398	.97930 to 1.03292	12300	22.7483
67	1.00648	+ or - .00964	.99684 to 1.01612	.98720 to 1.02576	.97756 to 1.03540	10800	24.8960
72	1.00416	+ or - .01059	.99357 to 1.01474	.98299 to 1.02533	.97240 to 1.03591	9300	27.8887
77	1.00754	+ or - .01211	.99543 to 1.01965	.98332 to 1.03176	.97121 to 1.04387	7800	29.6757
82	1.01864	+ or - .01271	1.00593 to 1.03135	.99322 to 1.04406	.98051 to 1.05677	6300	33.4440
87	1.01431	+ or - .01127	1.00305 to 1.02558	.99178 to 1.03685	.98051 to 1.04812	4800	33.7550
92	1.01657	+ or - .01438	1.00220 to 1.03095	.98782 to 1.04533	.97344 to 1.05971	3300	42.7123

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

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	deviation of variance (per cent)
97	1.02672	+ or - .02319	1.00352 to 1.04991	.98033 to 1.07311	.95714 to 1.09630	1800	79.9852


keno-vi sample problem 4 2c8 15.24 cm paraffin refl
plot of average k-effective by generation run.
the line represents k-eff = 1.0043 + or - .0056 which occurs for 103 generations run.
.9749
.9959
1.0168
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100


keno-vi sample problem 4 2c8 15.24 cm paraffin refl
plot of average k-effective by generation skipped.
the line represents k-eff = 1.0043 + or - .0056 which occurs for 4 generations skipped.
.9672
.9970
1.0267
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
k-effective satisfies the chi**2 test for normality at the 95 % level

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

skipping 3 generations

group	fission fraction	unit	region	fissions	percent deviation	absorptions	percent deviation	leakage	percent deviation
1	.0184			1.84701E-02	5.8334	5.54209E-03	5.5607	1.96663E-03	13.8332
2	.1020			1.02450E-01	1.7364	3.54026E-02	1.7363	5.33815E-03	9.6385
3	.1261			1.26693E-01	1.6160	4.85950E-02	1.6159	4.93048E-03	14.3449
4	.0759			7.62752E-02	1.9710	3.08251E-02	1.9709	1.47340E-03	24.8223
5	.1112			1.11671E-01	1.2601	4.75181E-02	1.2601	2.94632E-03	27.2082
6	.1489			1.49504E-01	1.3214	6.81557E-02	1.3213	3.02096E-03	38.6906
7	.1106			1.11064E-01	1.7378	5.51231E-02	1.7376	2.03781E-03	62.7555
8	.0311			3.11845E-02	3.5181	1.74295E-02	3.5150	6.34005E-04	100.0000
9	.0113			1.13691E-02	5.9914	6.55085E-03	5.9603	0.00000E+00	.0000
10	.0124			1.24042E-02	5.6597	7.29867E-03	5.5985	0.00000E+00	.0000
11	.0127			1.27907E-02	6.3373	8.03674E-03	6.1943	0.00000E+00	.0000
12	.0075			7.50082E-03	6.7521	4.88644E-03	6.4147	0.00000E+00	.0000
13	.0062			6.25951E-03	7.8526	4.32592E-03	7.0035	0.00000E+00	.0000
14	.0063			6.37577E-03	7.7963	4.80257E-03	6.5197	0.00000E+00	.0000
15	.0031			3.09272E-03	10.6119	2.63966E-03	8.0315	0.00000E+00	.0000
16	.0019			1.91460E-03	15.8373	1.45083E-03	10.9290	0.00000E+00	.0000
17	.0014			1.42425E-03	20.4165	1.01748E-03	15.9454	0.00000E+00	.0000
18	.0009			9.27781E-04	21.7149	6.64004E-04	15.1611	0.00000E+00	.0000
19	.0013			1.26621E-03	22.7931	1.01547E-03	13.3114	0.00000E+00	.0000
20	.0068			6.86497E-03	8.4038	4.98914E-03	5.3925	0.00000E+00	.0000
21	.0017			1.67639E-03	20.5226	1.57909E-03	11.2471	0.00000E+00	.0000
22	.0042			4.18869E-03	11.4219	4.37557E-03	6.0389	2.55520E-03	100.0000
23	.0285			2.85962E-02	4.5659	4.36570E-02	3.1665	0.00000E+00	.0000\
24	.0642			6.44390E-02	3.3302	1.40858E-01	3.8733	1.72443E-02	100.0000
25	.0498			4.99716E-02	3.6577	1.35130E-01	3.9578	0.00000E+00	.0000
26	.0469			4.70630E-02	3.8095	1.82018E-01	4.1159	0.00000E+00	.0000
27	.0088			8.86754E-03	8.0502	7.66408E-02	4.8079	0.00000E+00	.0000
system total =				1.00430E+00	.5570	9.40527E-01	2.4250	4.21473E-02	41.6057

elapsed time 5.56800 minutes

random number= 000037F7146C0979

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

**** fission densities ****

unit	region	fission density	percent deviation	total fissions
1	1	1.255E-01	.56	1.004E+00
	2	0.000E+00	.00	0.000E+00
global unit				
2	1	0.000E+00	.00	0.000E+00
	2	0.000E+00	.00	0.000E+00
	3	0.000E+00	.00	0.000E+00
	4	0.000E+00	.00	0.000E+00
	5	0.000E+00	.00	0.000E+00
	6	0.000E+00	.00	0.000E+00
	7	0.000E+00	.00	0.000E+00

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

fluxes for unit 1

group	region 1 flux	region 1 percent deviation	region 2 flux	region 2 percent deviation
1	9.598E-03	5.20	2.838E-02	5.33
2	9.093E-02	1.98	2.547E-01	1.91
3	1.137E-01	1.54	3.279E-01	1.59
4	7.125E-02	2.15	2.000E-01	2.20
5	1.110E-01	1.46	3.132E-01	1.50
6	1.664E-01	1.35	4.523E-01	1.26
7	1.170E-01	1.73	3.276E-01	1.83
8	3.190E-02	3.24	1.384E-01	3.07
9	1.389E-02	6.06	8.608E-02	3.35
10	1.230E-02	5.67	7.854E-02	3.83
11	1.111E-02	6.23	7.152E-02	4.57
12	7.136E-03	6.57	4.820E-02	4.79
13	6.563E-03	7.85	4.301E-02	5.40
14	6.823E-03	7.16	4.481E-02	5.02
15	3.015E-03	10.56	2.019E-02	8.05
16	1.618E-03	15.07	1.223E-02	10.51
17	1.001E-03	21.18	4.978E-03	14.14
18	5.960E-04	21.54	4.400E-03	14.38
19	6.735E-04	23.66	8.777E-03	12.93
20	3.900E-03	8.37	2.725E-02	5.85
21	1.152E-03	21.40	7.493E-03	11.52
22	2.579E-03	11.28	1.858E-02	8.83
23	1.713E-02	4.94	1.293E-01	3.72
24	3.781E-02	3.47	2.750E-01	2.63
25	2.923E-02	3.93	1.943E-01	3.14
26	2.738E-02	3.77	1.864E-01	3.20
27	5.231E-03	8.25	3.825E-02	5.56

keno-vi sample problem 4 2c8 15.24 cm paraffin refl

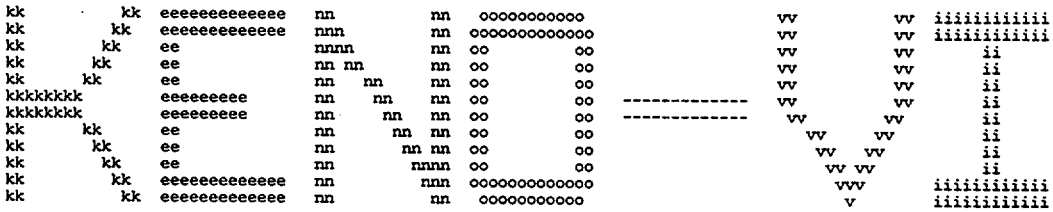
fluxes for global unit

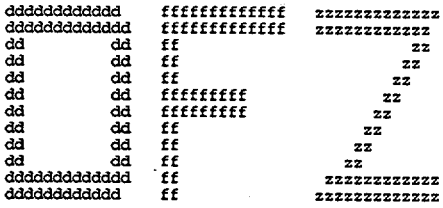
group	region 1 flux	region 1 percent deviation	region 2 flux	region 2 percent deviation	region 3 flux	region 3 percent deviation	region 4 flux	region 4 percent deviation	region -5 flux	region -5 percent deviation	region 6 flux	region 6 percent deviation
1	0.000E+00	.00	4.946E-02	5.82	4.863E-02	6.81	4.292E-02	7.91	3.245E-02	11.04	2.298E-02	12.68
2	0.000E+00	.00	3.826E-01	1.98	3.200E-01	2.84	2.122E-01	3.80	1.234E-01	4.72	6.949E-02	7.02
3	0.000E+00	.00	4.832E-01	1.55	3.605E-01	2.16	2.185E-01	3.14	1.295E-01	4.60	7.154E-02	7.37
4	0.000E+00	.00	3.078E-01	2.13	2.232E-01	2.94	1.244E-01	4.31	6.372E-02	7.22	3.538E-02	10.04
5	0.000E+00	.00	4.738E-01	1.56	3.222E-01	2.24	1.658E-01	3.51	8.231E-02	5.88	4.517E-02	13.59
6	0.000E+00	.00	7.167E-01	1.07	4.415E-01	1.79	2.113E-01	2.92	9.564E-02	6.50	4.359E-02	13.76
7	0.000E+00	.00	7.086E-01	1.14	3.890E-01	1.54	1.755E-01	3.80	8.106E-02	9.38	3.718E-02	19.32
8	0.000E+00	.00	4.816E-01	1.21	2.799E-01	1.75	1.347E-01	4.93	5.375E-02	9.85	1.404E-02	34.07
9	0.000E+00	.00	3.625E-01	1.29	2.301E-01	2.25	9.696E-02	4.65	4.055E-02	16.04	2.309E-02	41.29
10	0.000E+00	.00	3.070E-01	1.10	2.106E-01	1.95	9.619E-02	5.30	3.972E-02	17.72	2.659E-02	49.87
11	0.000E+00	.00	3.094E-01	1.23	2.181E-01	2.20	9.780E-02	5.61	2.974E-02	15.34	1.478E-02	47.03
12	0.000E+00	.00	2.043E-01	1.30	1.497E-01	2.04	7.553E-02	7.30	3.066E-02	23.09	2.542E-02	69.69
13	0.000E+00	.00	1.862E-01	1.56	1.450E-01	2.86	6.045E-02	7.16	4.038E-02	33.65	1.393E-02	60.73
14	0.000E+00	.00	1.937E-01	1.49	1.463E-01	3.06	6.938E-02	7.91	4.053E-02	22.42	2.274E-02	50.22
15	0.000E+00	.00	9.120E-02	2.12	7.647E-02	3.95	3.479E-02	9.54	1.960E-02	26.88	8.887E-03	75.83
16	0.000E+00	.00	4.907E-02	2.89	4.181E-02	4.04	1.940E-02	13.00	1.376E-02	36.28	7.458E-03	69.25
17	0.000E+00	.00	2.393E-02	4.50	1.925E-02	6.22	1.091E-02	17.46	6.639E-03	55.60	3.852E-03	100.00
18	0.000E+00	.00	2.180E-02	4.29	1.629E-02	6.66	8.576E-03	19.15	5.343E-03	40.71	0.000E+00	.00
19	0.000E+00	.00	4.040E-02	3.29	3.075E-02	4.65	2.177E-02	33.54	1.263E-02	34.07	3.264E-03	59.54
20	0.000E+00	.00	1.251E-01	2.17	1.021E-01	3.88	5.595E-02	10.61	3.647E-02	22.66	1.139E-02	48.07
21	0.000E+00	.00	4.095E-02	3.30	3.260E-02	6.78	1.742E-02	14.26	9.322E-03	34.90	5.610E-03	59.47
22	0.000E+00	.00	9.799E-02	2.68	8.628E-02	3.97	5.807E-02	13.34	4.736E-02	26.54	7.647E-03	50.54
23	0.000E+00	.00	8.213E-01	1.75	8.837E-01	3.52	5.801E-01	6.99	3.591E-01	15.67	9.286E-02	32.52
24	0.000E+00	.00	1.934E+00	1.66	2.215E+00	3.27	1.459E+00	6.98	9.681E-01	16.72	2.871E-01	29.99
25	0.000E+00	.00	1.484E+00	1.60	1.703E+00	3.34	1.178E+00	6.69	6.794E-01	17.62	2.022E-01	34.61
26	0.000E+00	.00	1.518E+00	1.65	1.749E+00	3.56	1.129E+00	6.98	6.716E-01	18.32	1.905E-01	35.44
27	0.000E+00	.00	3.551E-01	1.80	4.016E-01	3.86	2.596E-01	7.50	1.794E-01	20.93	5.211E-02	41.35

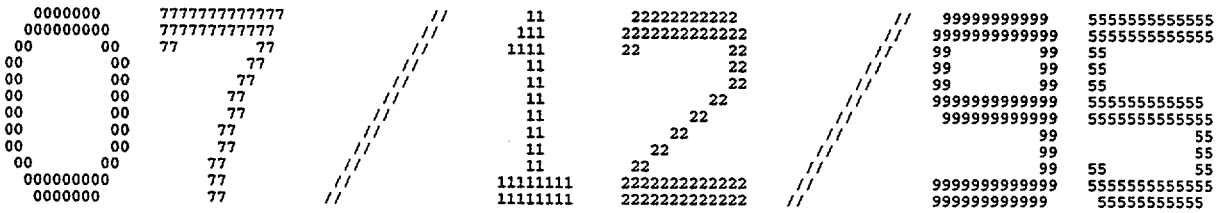
fluxes for global unit
region 7

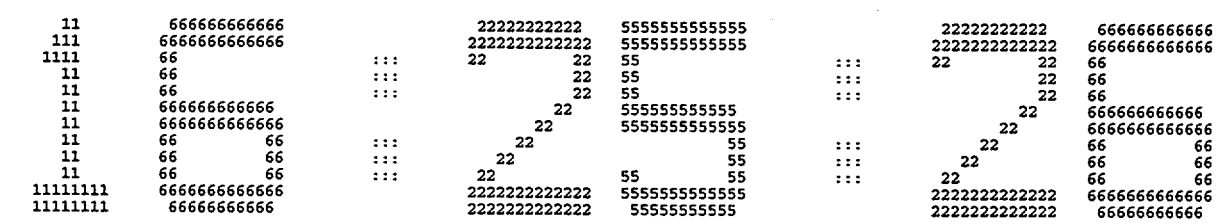
group	flux	percent deviation
1	3.501E-03	35.92
2	5.483E-03	23.16
3	4.652E-03	21.53
4	1.483E-03	38.68
5	3.753E-03	39.54
6	2.250E-03	37.83
7	6.510E-04	60.09
8	1.531E-04	100.00
9	0.000E+00	.00
10	5.947E-04	100.00
11	2.362E-03	100.00
12	8.608E-04	100.00
13	0.000E+00	.00
14	0.000E+00	.00
15	0.000E+00	.00
16	0.000E+00	.00
17	0.000E+00	.00
18	0.000E+00	.00
19	0.000E+00	.00
20	0.000E+00	.00
21	0.000E+00	.00
22	1.001E-03	100.00
23	0.000E+00	.00
24	4.139E-03	100.00
25	2.138E-03	78.23
26	0.000E+00	.00
27	0.000E+00	.00

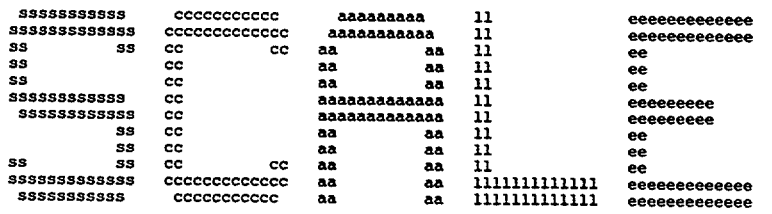
```
keno-vi sample problem 4  2c8  15.24 cm paraffin refl

                                frequency for generations    4 to  103
 .8694 to  .8924      **
 .8924 to  .9155      *
 .9155 to  .9386      *********
 .9386 to  .9617      *************
 .9617 to  .9848      ****************
 .9848 to 1.0079      **********
1.0079 to 1.0310      ****************
1.0310 to 1.0541      **************
1.0541 to 1.0772      ************
1.0772 to 1.1003      ***
1.1003 to 1.1234      **
1.1234 to 1.1465
1.1465 to 1.1696      *
1.1696 to 1.1927      *

                                frequency for generations   29 to  103
 .8694 to  .8924      **
 .8924 to  .9155
 .9155 to  .9386      ********
 .9386 to  .9617      *********
 .9617 to  .9848      ***********
 .9848 to 1.0079      ******
1.0079 to 1.0310      ***********
1.0310 to 1.0541      ************
1.0541 to 1.0772      **********
1.0772 to 1.1003      **
1.1003 to 1.1234      **
1.1234 to 1.1465
1.1465 to 1.1696      *
1.1696 to 1.1927      *

                                frequency for generations   54 to  103
 .8694 to  .8924      *
 .8924 to  .9155
 .9155 to  .9386      *******
 .9386 to  .9617      *****
 .9617 to  .9848      ********
 .9848 to 1.0079      *****
1.0079 to 1.0310      ********
1.0310 to 1.0541      ********
1.0541 to 1.0772      ***
1.0772 to 1.1003      **
1.1003 to 1.1234      **
1.1234 to 1.1465
1.1465 to 1.1696      *
1.1696 to 1.1927

                                frequency for generations   79 to  103
 .8694 to  .8924      *
 .8924 to  .9155
 .9155 to  .9386      **
 .9386 to  .9617      ***
 .9617 to  .9848      ****
 .9848 to 1.0079      *
1.0079 to 1.0310      ****
1.0310 to 1.0541      *****
1.0541 to 1.0772      ***
1.0772 to 1.1003
1.1003 to 1.1234      *
1.1234 to 1.1465
1.1465 to 1.1696      *
1.1696 to 1.1927

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

        congratulations!  you have  successfully traversed the perilous path through keno vi in   5.56800 minutes
*************************************************************************************************************************************************
```

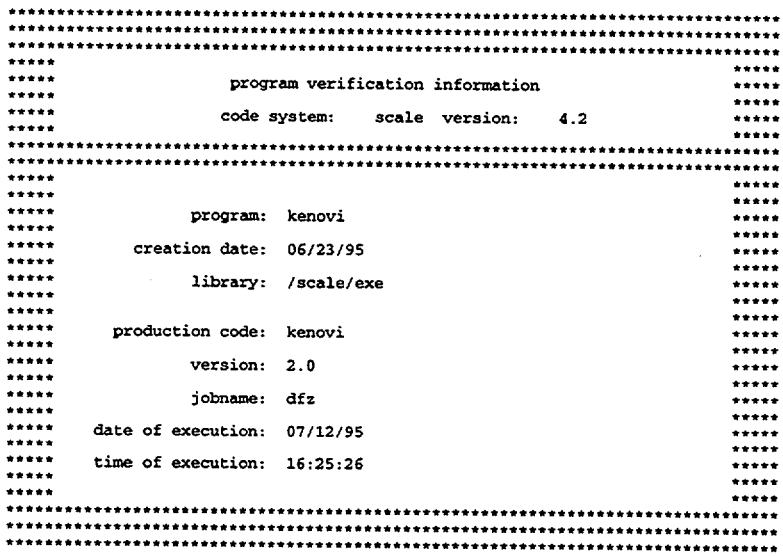









sssssssssss ccccccccccc aaaaaaaaa ll eeeeeeeeeeeee
ssssssssssss ccccccccccccc aaaaaaaaaaa ll eeeeeeeeeeeee
ss ss cc cc aa aa ll ee
ss cc aa aa ll ee
ss cc aa aa ll ee
sssssssssss cc aaaaaaaaaaaaa ll eeeeeeeee
ssssssssssss cc aaaaaaaaaaaaa ll eeeeeeeee
ss cc aa aa ll ee
ss cc aa aa ll ee
ss ss cc cc aa aa ll ee
ssssssssssss ccccccccccccc aa aa llllllllllll eeeeeeeeeeee
sssssssssss ccccccccccc aa aa llllllllllll eeeeeeeeeeee


program verification information
code system: scale version: 4.2
program: kenovi
creation date: 06/23/95
library: /scale/exe
production code: kenovi
version: 2.0
jobname: dfz
date of execution: 07/12/95
time of execution: 16:25:26

sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)

****** numeric parameters ******

tme	maximum problem time (min)	30.00
tba	time per generation (min)	.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
xld	number of extra 1-d cross sections	1
nbk	neutron bank size	325
xnb	extra positions in neutron bank	0
nfb	fission bank size	300
xfb	extra positions in fission bank	0
sig	cut off standard deviation	.0000
wta	default value of weight average	.5000
wth	weight high for splitting	3.0000
wtl	weight low for russian roulette	.3333
rnd	starting random number	00000BB827100001
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 19 4 aqueous 4 metal array of arrays (samp prob 12)                           ***
***                                                                                                                       ***
*****************************************************************************************************************************
***                                                                                                                       ***
***                                  ******      logical parameters      ******                                       ***
***                                                                                                                       ***
***  run  execute problem after checking data    yes          plt  plot 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   no          ckp  compute cofactor k-eff by unit location       no  ***
***                                                                                                                       ***
***  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     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          gas  print far by group                       no  ***
***                                                                                                                       ***
***  xs2  print 2-d mixture x-sections            no          pax  print xsec-albedo correlation tables       no  ***
***                                                                                                                       ***
***  xap  print mixture angles & probabilities    no          pwt  print weight average array                no  ***
***                                                                                                                       ***
***  pki  print fission spectrum                  no          pgm  print input geometry                      no  ***
***                                                                                                                       ***
***  p1d  print extra 1-d cross sections          no          bug  print debug information                   no  ***
***                                                                                                                       ***
***                                                           trk  print tracking information                no  ***
***                                                                                                                       ***
*****************************************************************************************************************************
*****************************************************************************************************************************
*****************************************************************************************************************************
                                        parameter input completed

                .....     0 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	ft14f001		mixed cross sections
alb	79	/scale/datalib/albedos		input albedos
wts	80	/scale/datalib/weights		input weights
skt	16	unknown		write scratch data
lib	4	ft04f001		input ampx working library
	8	ft08f001		input data direct access
	9	unknown		super grouped direct access
	10	unknown		xsec mixing direct access

..... 0 io's were used preparing input data .....

cross sections read from the ampx working library on unit 4

sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)

mixing table

number of scattering angles = 2
cross section message threshold =3.0E-05

mixture = 1		density(g/cc) = 18.760					
nuclide	atom-dens.	wgt. frac.	za	awt	nuclide title		
92234	4.82716E-04	9.99999E-03	92234	234.0405	uranium-234	endf/b-iv mat 1043	updated 08/12/94
92235	4.47971E-02	9.32000E-01	92235	235.0441	uranium-235	endf/b-iv mat 1261	updated 08/12/94
92236	9.57231E-05	2.00000E-03	92236	236.0458	u-236 1163 sigo=5+4 newxlacs p-3 293k f-1/e-m(1.+5)		updated 08/12/94
92238	2.65767E-03	5.60000E-02	92238	238.0510	uranium-238	endf/b-iv mat 1262	updated 08/12/94

mixture = 2		density(g/cc) = 1.5547					
nuclide	atom-dens.	wgt. frac.	za	awt	nuclide title		
1001	5.77964E-02	6.22048E-02	1001	1.0077	hydrogen	endf/b-iv mat 1269/thrm1002	updated 08/12/94
7014	2.13092E-03	3.18719E-02	7014	14.0033	nitrogen-14	endf/b-iv mat 1275	updated 08/12/94
8016	3.74130E-02	6.38986E-01	8016	15.9904	oxygen-16	endf/b-iv mat 1276	updated 08/12/94
292234	1.06784E-05	2.66936E-03	92234	234.0405	uranium-234	endf/b-iv mat 1043	updated 08/12/94
292235	9.84599E-04	2.47184E-01	92235	235.0441	uranium-235	endf/b-iv mat 1261	updated 08/12/94
292236	5.29385E-06	1.33468E-03	92236	236.0458	u-236 1163 sigo=5+4 newxlacs p-3 293k f-1/e-m(1.+5)		updated 08/12/94
292238	6.19413E-05	1.57493E-02	92238	238.0510	uranium-238	endf/b-iv mat 1262	updated 08/12/94

mixture = 3		density(g/cc) = 1.1799					
nuclide	atom-dens.	wgt. frac.	za	awt	nuclide title		
1101001	5.68187E-02	8.05783E-02	1001	1.0077	hydrogen	endf/b-iv mat 1269/thrm1002	updated 08/12/94
1106012	3.55117E-02	5.99750E-01	6000	12.0001	carbon-12	endf/b-iv mat 1274/thrm1065	updated 08/12/94
1108016	1.42047E-02	3.19672E-01	8016	15.9904	oxygen-16	endf/b-iv mat 1276	updated 08/12/94

1001	hydrogen	endf/b-iv mat 1269/thrm1002	updated 08/12/94
1101001	hydrogen	endf/b-iv mat 1269/thrm1002	updated 08/12/94
1106012	carbon-12	endf/b-iv mat 1274/thrm1065	updated 08/12/94
7014	nitrogen-14	endf/b-iv mat 1275	updated 08/12/94
8016	oxygen-16	endf/b-iv mat 1276	updated 08/12/94
1108016	oxygen-16	endf/b-iv mat 1276	updated 08/12/94
92234	uranium-234	endf/b-iv mat 1043	updated 08/12/94
292234	uranium-234	endf/b-iv mat 1043	updated 08/12/94
92235	uranium-235	endf/b-iv mat 1261	updated 08/12/94
292235	uranium-235	endf/b-iv mat 1261	updated 08/12/94
92236	u-236 1163 sigo=5+4 newxlacs p-3 293k f-1/e-m(1.+5)		updated 08/12/94
292236	u-236 1163 sigo=5+4 newxlacs p-3 293k f-1/e-m(1.+5)		updated 08/12/94
92238	uranium-238	endf/b-iv mat 1262	updated 08/12/94
292238	uranium-238	endf/b-iv mat 1262	updated 08/12/94

keno message number k6-222 2 transfers for mixture 2 were corrected for bad moments.

keno message number k6-222 3 transfers for mixture 3 were corrected for bad moments.

..... 0 io's were used mixing cross-sections .....

1-d cross section array id numbers
1 2002 1452 27 18 1018

..... 0 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                27       use lattice geometry                    yes     ***
***                                                                                                   ***
***  no. of fission spectrum source group    1       global array number                       3     ***
***                                                                                                   ***
***  no. of scattering angles in xsecs       2       number of units in the global x dir.      2     ***
***                                                                                                   ***
***  entries/neutron in the neutron bank    32       number of units in the global y dir.      1     ***
***                                                                                                   ***
***  entries/neutron in the fission bank    19       number of units in the global z dir.      1     ***
***                                                                                                   ***
***  number of mixtures used                 3       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            9       maximum hole nesting level                0     ***
***                                                                                                   ***
***  number of geometry regions used         9       use nested arrays                       yes     ***
***                                                                                                   ***
***  largest geometry unit number            5       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 ******                     ***
***                                                                                       ***
***        20000 words is the total space available.                                      ***
***                                                                                       ***
***        18415 words were used for non-supergroup storage.                              ***
***                                                                                       ***
***         1585 words of storage are available for supergrouped data.                    ***
***                                                                                       ***
***        19843 words of storage are available for constructing the supergroups.         ***
***                                                                                       ***
***         1501 words of storage are available to each supergroup.                       ***
***                                                                                       ***
***          477 words are needed for the largest group.                                  ***
***                                                                                       ***
***        19108 words of storage is sufficient to run this problem.                      ***
***                                                                                       ***
***        24920 words of storage will allow the problem to run with one supergroup.      ***
***                                                                                       ***
***        20000 words of storage will be used to run this problem.                       ***
***                                                                                       ***
*********************************************************************************************
*********************************************************************************************
***                                                                                       ***
***                    starting      ending       xsec       albedo       total         ***
***   supergroup        group         group       length      length       length        ***
***                                                                                       ***
***                                                                                       ***
***          1           1             3          234          0          1233           ***
***          2           4             7          264          0          1425           ***
***          3           8            12          225          0          1341           ***
***          4          13            19          206          0          1409           ***
***          5          20            27          170          0          1337           ***
***                                                                                       ***
*********************************************************************************************

        ........    0 io's were used in supergrouping    ........

        ***********************************************************
        **                                                       **
        **  array      units in    units in    units in   nesting **
        **  number      x dir.      y dir.      z dir.     level  **
        **                                                       **
        **    1            1           2           2          2   **
        **                                                       **
        **    2            1           2           2          2   **
        **                                                       **
        **    3 global     2           1           1          1   **
        **                                                       **
        ***********************************************************

        ........    0 io's were used loading the data    ........
```

****** geometry parameters ******

niar	number of independent array references	3
ngblu	global unit number	5
nboxt	number of units in the problem	5
nquad	number of quadratics in the problem	24
ngwrds	number of geometry words read	9
maxgwd	maximum geometry words in a unit	3
maxsfu	largest number of surfaces in a unit	7
maxreg	largest number of media in a unit	3
regtot	number of spatial volumes defined	9
sectot	number of entries in the sector array	19
nucom	number of comments in the geometry data	5
numhol	number of holes in the problem	0

sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)
geometry description for those units utilized in this problem

----- unit 1 -----

uranyl nitrate solution in a plexiglas container

```
  1          cylinder          10
                                                    quadratic surfaces
   -1.000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     90.726
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     79.032
  2          cylinder          20
                                                    quadratic surfaces
   -1.000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z    103.226
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     90.726
  3          cuboid            30
                                                    quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z    118.266
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z    118.266
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z    104.858
                  sector
            imp      definitions
media    2    1        10
media    3    1       -10   20
media    0    1        30  -20
boundary               30
```

----- unit 2 -----

uranium metal cylinder

```
  1          cylinder          10
                                                    quadratic surfaces
   -1.000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     33.040
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     28.971
  2          cuboid            20
                                                    quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     43.428
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     43.428
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y      .000z     38.751
                  sector
            imp      definitions
media    1    1        10
media    0    1        20  -10
boundary               20
```

----- unit 3 -----

1x2x2 array of solution units

```
  1          cuboid            10
                                                    quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz     21.750x      .000y      .000z      .000
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x     43.500y      .000z      .000
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y     40.960z      .000
                  sector
            imp      definitions
array    1              10
boundary                10
```

----- unit 4 -----

1x2x2 array of metal units padded to match solution array

```
  1          cuboid            10
                                                    quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz     13.180x      .000y      .000z      .000
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x     26.360y      .000z      .000
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y     24.900z      .000
  2          cuboid            20
                                                    quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz     13.180x      .000y      .000z      .000
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x     26.360y      .000z    299.350
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y     24.900z    264.428
                  sector
            imp      definitions
array    2              10
media    0    1         20  -10
boundary                20
```

********** global **********

----- unit 5 -----

global unit of arrays 1 and 2

```
  1          cuboid            10
                                                    quadratic surfaces
   -1.000x**2      .000y**2      .000z**2      .000xy      .000xz      .000yz     34.930x      .000y      .000z      .000
     .000x**2    -1.000y**2      .000z**2      .000xy      .000xz      .000yz      .000x     43.500y      .000z      .000
     .000x**2      .000y**2    -1.000z**2      .000xy      .000xz      .000yz      .000x      .000y     40.960z      .000
                  sector
            imp      definitions
array    3              10
boundary                10
```

```
                    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

                -------  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

                -------  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	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	2	1.00000E+00 cm**3	1.00000E+00 cm**3
	3	3	1.00000E+00 cm**3	1.00000E+00 cm**3
2	1	4	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	5	1.00000E+00 cm**3	1.00000E+00 cm**3
3	1	6	1.00000E+00 cm**3	1.00000E+00 cm**3
4	1	7	1.00000E+00 cm**3	1.00000E+00 cm**3
	2	8	1.00000E+00 cm**3	1.00000E+00 cm**3
5	1	9	1.00000E+00 cm**3	1.00000E+00 cm**3

unit	uses	region	mixture	total volume
1	4	1	2	4.00000E+00 cm**3
		2	3	4.00000E+00 cm**3
		3	0	4.00000E+00 cm**3
2	4	1	1	4.00000E+00 cm**3
		2	0	4.00000E+00 cm**3
3	1	1		1.00000E+00 cm**3
4	1	1		1.00000E+00 cm**3
		2	0	1.00000E+00 cm**3
5	1	1		1.00000E+00 cm**3

total mixture volumes

mixture	total volume
0	9.00000E+00 cm**3
1	4.00000E+00 cm**3
2	4.00000E+00 cm**3
3	4.00000E+00 cm**3

biasing information

a default weight of .500 will be used for all bias id's.

..... 0 io's were used in keno-v before tracking .....

..... .02133minutes were used processing data. .....

volume fraction of fissile material in the system= .38095E+00

start type 0 was used.

the neutrons were started with a flat distribution in a cuboid defined by:
+x= 3.49300E+01 -x= 0.00000E+00 +y= 4.35000E+01 -y= 0.00000E+00 +z= 4.09600E+01 -z= 0.00000E+00

.00000 minutes were required for starting. total elapsed time is .02133 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 k6-132		warning....only	267 independent fission points were generated			
1	7.66024E-01	2.13333E-02	1.00000E+00	0.00000E+00	7.83855E-01	0.00000E+00
2	9.38291E-01	4.26667E-02	1.00000E+00	0.00000E+00	8.97564E-01	1.01336E-01
3	9.66243E-01	4.26667E-02	9.66243E-01	0.00000E+00	9.14966E-01	5.96036E-02
4	1.00734E+00	4.26667E-02	9.86789E-01	2.05463E-02	9.53405E-01	6.22936E-02
5	9.07602E-01	6.40000E-02	9.60393E-01	2.89387E-02	9.56865E-01	6.28099E-02
6	9.57428E-01	8.53333E-02	9.59652E-01	2.04762E-02	9.61773E-01	6.00777E-02
7	1.05669E+00	8.53333E-02	9.79059E-01	2.50642E-02	9.82765E-01	6.09066E-02
8	1.10116E+00	1.06667E-01	9.99409E-01	2.88602E-02	9.90786E-01	5.45231E-02
9	1.10255E+00	1.06667E-01	1.01414E+00	2.84961E-02	1.00220E+00	4.86219E-02
10	1.02324E+00	1.06667E-01	1.01528E+00	2.47045E-02	1.00405E+00	4.34626E-02
11	1.14941E+00	1.28000E-01	1.03018E+00	2.63967E-02	1.01487E+00	4.04249E-02
12	9.35704E-01	1.49333E-01	1.02073E+00	2.54301E-02	1.00840E+00	3.75113E-02
13	9.59340E-01	1.49333E-01	1.01515E+00	2.36698E-02	1.00556E+00	3.53034E-02
14	1.01802E+00	1.70667E-01	1.01539E+00	2.16088E-02	1.00522E+00	3.26037E-02
15	9.99063E-01	1.92000E-01	1.01414E+00	1.99169E-02	1.00481E+00	3.05648E-02
16	1.07506E+00	1.92000E-01	1.01849E+00	1.89459E-02	1.00946E+00	2.89452E-02
17	1.01466E+00	1.92000E-01	1.01823E+00	1.76395E-02	1.01003E+00	2.75420E-02
keno message number k6-132		warning....only	296 independent fission points were generated			
18	9.47762E-01	1.92000E-01	1.01383E+00	1.70780E-02	1.00565E+00	2.59399E-02
19	1.01894E+00	2.13333E-01	1.01413E+00	1.60448E-02	1.00557E+00	2.44866E-02
20	9.87500E-01	2.34667E-01	1.01265E+00	1.51993E-02	1.00525E+00	2.33060E-02
21	9.33306E-01	2.34667E-01	1.00847E+00	1.49713E-02	1.00268E+00	2.23544E-02
22	1.04988E+00	2.56000E-01	1.01054E+00	1.43531E-02	1.00570E+00	2.14367E-02
23	9.02897E-01	2.56000E-01	1.00542E+00	1.45831E-02	1.00247E+00	2.08415E-02
24	1.04664E+00	2.56000E-01	1.00729E+00	1.40301E-02	1.00402E+00	2.00991E-02
25	9.99120E-01	2.77333E-01	1.00694E+00	1.34110E-02	1.00450E+00	1.94233E-02
26	1.09381E+00	2.98667E-01	1.01055E+00	1.33405E-02	1.00902E+00	1.88625E-02
27	9.64773E-01	2.98667E-01	1.00872E+00	1.29261E-02	1.00740E+00	1.83023E-02
28	9.04862E-01	3.20000E-01	1.00473E+00	1.30457E-02	1.00510E+00	1.83017E-02
29	9.72398E-01	3.41333E-01	1.00353E+00	1.26102E-02	1.00389E+00	1.82303E-02
30	9.88587E-01	3.41333E-01	1.00300E+00	1.21632E-02	1.00323E+00	1.80317E-02
31	1.03125E+00	3.41333E-01	1.00397E+00	1.17766E-02	1.00336E+00	1.75519E-02
32	9.66574E-01	3.62667E-01	1.00273E+00	1.14454E-02	1.00280E+00	1.70535E-02
33	1.12385E+00	3.62667E-01	1.00663E+00	1.17393E-02	1.00670E+00	1.68599E-02
34	9.68236E-01	3.84000E-01	1.00543E+00	1.14297E-02	1.00642E+00	1.65707E-02
35	1.03058E+00	3.84000E-01	1.00619E+00	1.11041E-02	1.00738E+00	1.64420E-02
36	1.05318E+00	4.05333E-01	1.00758E+00	1.08608E-02	1.00853E+00	1.62703E-02
37	1.01871E+00	4.26667E-01	1.00789E+00	1.05508E-02	1.00818E+00	1.59709E-02
38	1.05509E+00	4.26667E-01	1.00921E+00	1.03370E-02	1.00946E+00	1.55677E-02
39	9.83623E-01	4.26667E-01	1.00851E+00	1.00775E-02	1.00934E+00	1.52476E-02
40	1.10613E+00	4.48000E-01	1.01108E+00	1.01395E-02	1.01192E+00	1.50791E-02
41	1.03181E+00	4.48000E-01	1.01161E+00	9.89034E-03	1.01193E+00	1.47503E-02
42	1.07090E+00	4.69333E-01	1.01310E+00	9.75320E-03	1.01305E+00	1.44444E-02
43	1.03679E+00	4.90667E-01	1.01367E+00	9.52987E-03	1.01373E+00	1.41244E-02
44	1.05677E+00	4.90667E-01	1.01470E+00	9.35664E-03	1.01422E+00	1.38713E-02
45	1.06848E+00	4.90667E-01	1.01595E+00	9.22165E-03	1.01504E+00	1.38070E-02
46	1.06380E+00	5.12000E-01	1.01704E+00	9.07502E-03	1.01647E+00	1.36907E-02
47	1.06224E+00	5.12000E-01	1.01804E+00	8.92774E-03	1.01725E+00	1.34867E-02
48	1.13320E+00	5.33333E-01	1.02055E+00	9.08326E-03	1.01938E+00	1.34012E-02
49	9.98548E-01	5.54667E-01	1.02008E+00	8.90022E-03	1.01877E+00	1.31955E-02
50	9.61480E-01	5.76000E-01	1.01886E+00	8.79793E-03	1.01839E+00	1.29485E-02
51	1.03020E+00	5.76000E-01	1.01909E+00	8.61962E-03	1.01894E+00	1.27361E-02
52	9.27334E-01	5.76000E-01	1.01725E+00	8.64255E-03	1.01699E+00	1.25846E-02
53	1.03050E+00	5.97333E-01	1.01751E+00	8.47537E-03	1.01688E+00	1.23602E-02
54	1.02054E+00	5.97333E-01	1.01757E+00	8.31099E-03	1.01701E+00	1.21453E-02
55	1.01085E+00	6.18667E-01	1.01745E+00	8.15366E-03	1.01679E+00	1.20240E-02
56	1.04757E+00	6.18667E-01	1.01800E+00	8.02067E-03	1.01712E+00	1.19359E-02
57	1.02074E+00	6.40000E-01	1.01805E+00	7.87364E-03	1.01764E+00	1.19392E-02
58	9.34674E-01	6.40000E-01	1.01656E+00	7.87382E-03	1.01651E+00	1.21162E-02
59	1.05164E+00	6.40000E-01	1.01718E+00	7.75888E-03	1.01706E+00	1.21847E-02
60	1.01307E+00	6.61333E-01	1.01711E+00	7.62427E-03	1.01701E+00	1.24888E-02
61	1.10038E+00	6.61333E-01	1.01852E+00	7.62568E-03	1.01816E+00	1.29250E-02
62	1.06235E+00	6.82667E-01	1.01925E+00	7.53301E-03	1.01851E+00	1.31459E-02
63	1.06503E+00	7.04000E-01	1.02000E+00	7.44640E-03	1.01911E+00	1.29891E-02
64	1.01986E+00	7.04000E-01	1.02000E+00	7.32532E-03	1.01928E+00	1.31282E-02
65	1.08213E+00	7.25333E-01	1.02098E+00	7.27527E-03	1.02047E+00	1.34135E-02
66	9.62545E-01	7.25333E-01	1.02007E+00	7.21867E-03	1.01950E+00	1.35088E-02
67	9.75206E-01	7.25333E-01	1.01938E+00	7.14019E-03	1.01853E+00	1.36395E-02
68	1.02603E+00	7.46667E-01	1.01948E+00	7.03189E-03	1.01852E+00	1.37736E-02
69	9.75495E-01	7.46667E-01	1.01883E+00	6.95719E-03	1.01783E+00	1.39108E-02
70	1.04304E+00	7.68000E-01	1.01918E+00	6.86336E-03	1.01809E+00	1.38588E-02
71	9.97267E-01	7.68000E-01	1.01886E+00	6.77061E-03	1.01724E+00	1.37483E-02
72	1.09516E+00	7.89333E-01	1.01995E+00	6.76161E-03	1.01830E+00	1.36257E-02
73	1.01785E+00	8.10667E-01	1.01992E+00	6.66576E-03	1.01801E+00	1.34444E-02
74	1.04521E+00	8.10667E-01	1.02028E+00	6.58190E-03	1.01855E+00	1.32779E-02
75	9.94144E-01	8.10667E-01	1.01992E+00	6.50097E-03	1.01777E+00	1.31191E-02
76	1.00068E+00	8.32000E-01	1.01966E+00	6.41779E-03	1.01762E+00	1.29719E-02
77	1.11551E+00	8.32000E-01	1.02094E+00	6.45935E-03	1.01901E+00	1.29111E-02
78	1.08147E+00	8.53333E-01	1.02173E+00	6.42337E-03	1.01977E+00	1.27638E-02
79	9.62230E-01	8.74667E-01	1.02096E+00	6.38632E-03	1.01877E+00	1.26085E-02
80	1.01192E+00	8.74667E-01	1.02084E+00	6.30498E-03	1.01870E+00	1.24520E-02
81	1.03051E+00	8.74667E-01	1.02097E+00	6.22586E-03	1.01840E+00	1.23166E-02
82	1.03003E+00	8.96000E-01	1.02108E+00	6.14859E-03	1.01845E+00	1.21759E-02
83	9.92268E-01	8.96000E-01	1.02072E+00	6.08262E-03	1.01806E+00	1.20397E-02
84	1.01649E+00	9.17333E-01	1.02067E+00	6.00820E-03	1.01818E+00	1.19353E-02
85	1.02330E+00	9.38667E-01	1.02070E+00	5.93546E-03	1.01833E+00	1.18126E-02
86	9.92399E-01	9.38667E-01	1.02037E+00	5.87404E-03	1.01785E+00	1.16888E-02
87	8.91626E-01	9.60000E-01	1.01885E+00	5.99888E-03	1.01653E+00	1.15902E-02
88	9.19713E-01	9.60000E-01	1.01770E+00	6.03975E-03	1.01546E+00	1.14812E-02
89	9.77252E-01	9.60000E-01	1.01723E+00	5.98800E-03	1.01485E+00	1.13723E-02
90	1.05494E+00	9.81333E-01	1.01766E+00	5.93504E-03	1.01531E+00	1.13617E-02
91	9.85871E-01	9.81333E-01	1.01731E+00	5.87884E-03	1.01515E+00	1.13296E-02
92	9.30061E-01	1.00267E+00	1.01634E+00	5.89343E-03	1.01437E+00	1.12229E-02
93	1.00741E+00	1.02400E+00	1.01624E+00	5.82913E-03	1.01440E+00	1.11048E-02

94	9.64425E-01	1.02400E+00	1.01567E+00	5.79286E-03	1.01380E+00	1.10428E-02
95	9.92586E-01	1.02400E+00	1.01543E+00	5.73561E-03	1.01345E+00	1.09262E-02
96	9.19276E-01	1.02400E+00	1.01440E+00	5.76572E-03	1.01256E+00	1.08677E-02
97	1.01622E+00	1.04533E+00	1.01442E+00	5.70474E-03	1.01242E+00	1.07951E-02
98	1.01975E+00	1.06667E+00	1.01448E+00	5.64528E-03	1.01247E+00	1.07242E-02
99	1.06121E+00	1.06667E+00	1.01496E+00	5.60751E-03	1.01293E+00	1.06484E-02
100	1.09792E+00	1.08800E+00	1.01581E+00	5.61419E-03	1.01358E+00	1.05702E-02
101	1.00695E+00	1.08800E+00	1.01572E+00	5.55791E-03	1.01369E+00	1.05067E-02
102	9.72192E-01	1.10933E+00	1.01528E+00	5.51924E-03	1.01340E+00	1.04936E-02
103	1.02963E+00	1.10933E+00	1.01542E+00	5.46617E-03	1.01324E+00	1.04766E-02

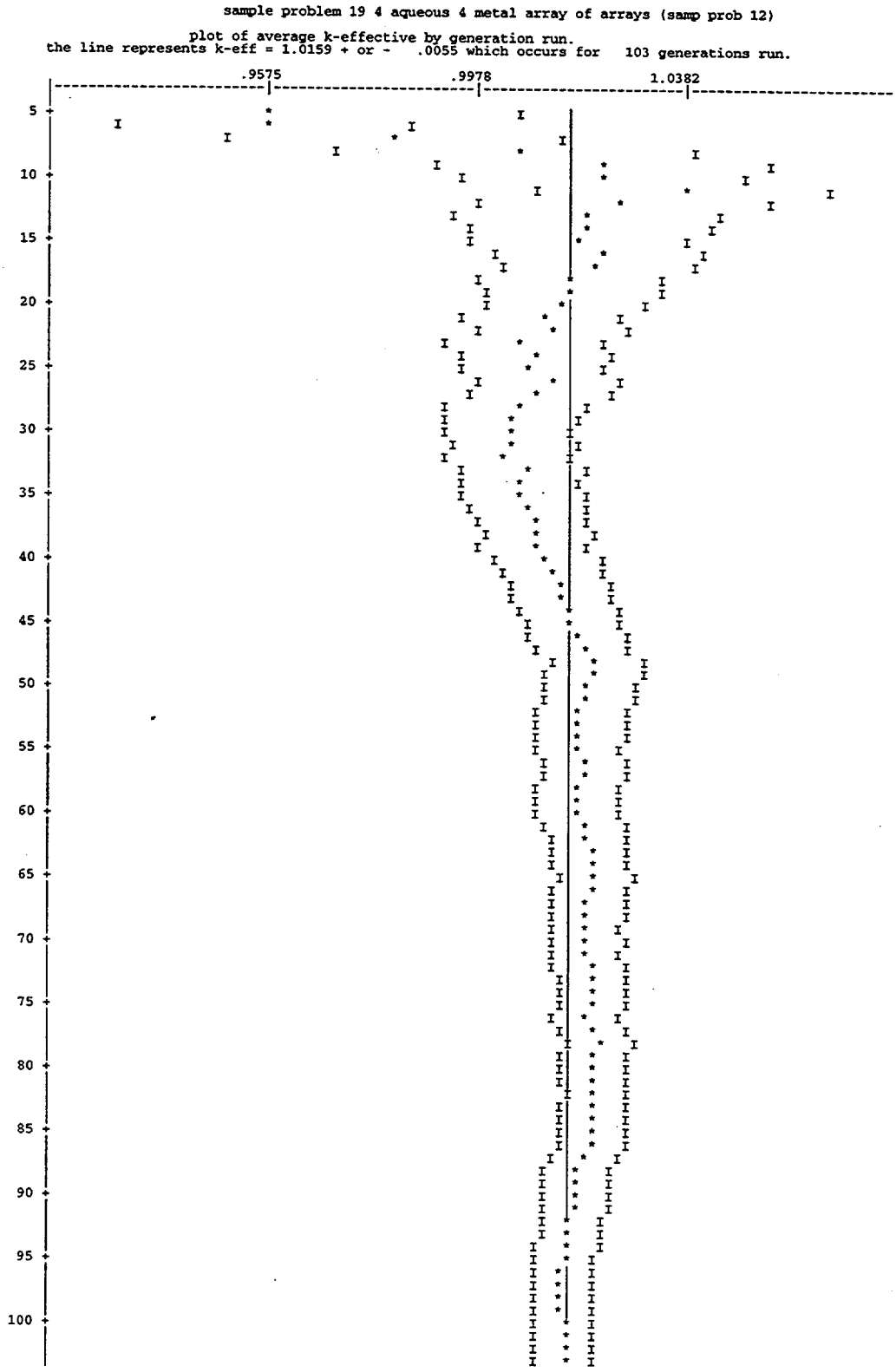
keno message number k6-123 execution terminated due to completion of the specified number of generations.

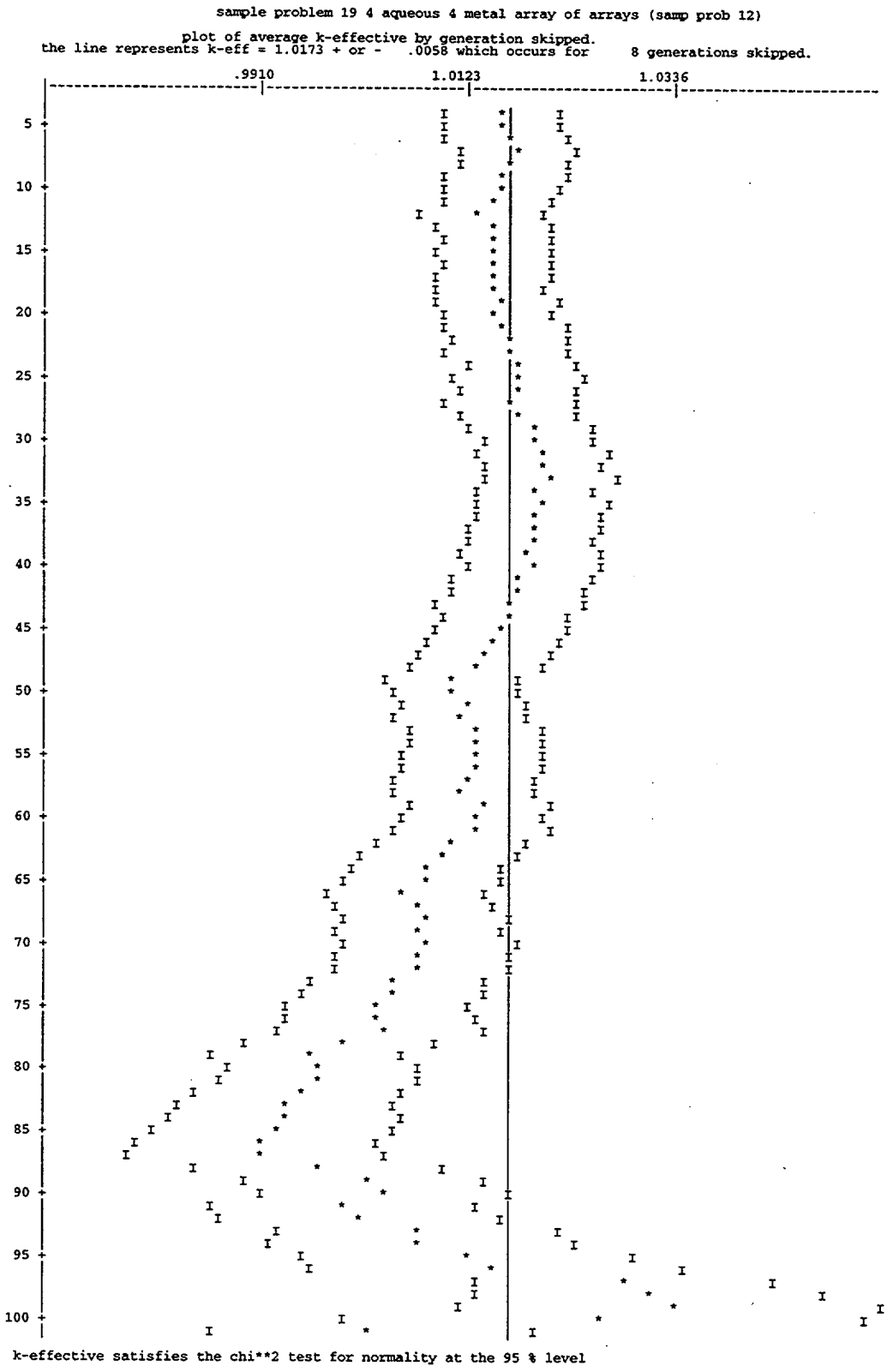
the matrix k-effective is the largest eigenvalue of the fission production by position index matrix.
there are nbxmax * nbymax * nbzmax positions in an array.

sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)

lifetime = 4.47772E-06 + or - 8.99083E-08 generation time = 5.31792E-06 + or - 1.24913E-07
nu bar = 2.50608E+00 + or - 1.40421E-03 average fission group = 1.41476E+01 + or - 9.90757E-02
energy(ev) of the average lethargy causing fission = 1.85333E+02 + or - 1.66111E+01
self multiplication = 1.01225E+00 + or - 5.92558E-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	deviation of variance (per cent)
3	1.01592	+ or - .00550	1.01042 to 1.02141	1.00492 to 1.02691	.99942 to 1.03241	30000	13.2575
4	1.01600	+ or - .00555	1.01045 to 1.02156	1.00489 to 1.02711	.99934 to 1.03266	29700	13.2262
5	1.01711	+ or - .00550	1.01161 to 1.02261	1.00611 to 1.02811	1.00061 to 1.03360	29400	13.4186
6	1.01772	+ or - .00552	1.01220 to 1.02324	1.00668 to 1.02876	1.00116 to 1.03429	29100	13.5929
7	1.01732	+ or - .00556	1.01175 to 1.02288	1.00619 to 1.02844	1.00063 to 1.03401	28800	13.6570
8	1.01644	+ or - .00555	1.01088 to 1.02199	1.00533 to 1.02754	.99978 to 1.03309	28500	13.9239
9	1.01552	+ or - .00553	1.00999 to 1.02105	1.00445 to 1.02659	.99892 to 1.03212	28200	14.2077
10	1.01544	+ or - .00559	1.00984 to 1.02103	1.00425 to 1.02662	.99866 to 1.03221	27900	14.1736
11	1.01398	+ or - .00546	1.00852 to 1.01944	1.00306 to 1.02490	.99760 to 1.03036	27600	13.8415
12	1.01484	+ or - .00545	1.00939 to 1.02029	1.00394 to 1.02574	.99849 to 1.03119	27300	14.1486
17	1.01493	+ or - .00569	1.00925 to 1.02062	1.00356 to 1.02631	.99788 to 1.03199	25800	14.4451
22	1.01663	+ or - .00586	1.01076 to 1.02249	1.00490 to 1.02836	.99903 to 1.03422	24300	15.1754
27	1.01763	+ or - .00592	1.01170 to 1.02355	1.00578 to 1.02948	.99986 to 1.03540	22800	15.9811
32	1.02079	+ or - .00603	1.01476 to 1.02682	1.00872 to 1.03286	1.00269 to 1.03889	21300	16.7464
37	1.01942	+ or - .00623	1.01319 to 1.02564	1.00697 to 1.03187	1.00074 to 1.03809	19800	17.2931
42	1.01695	+ or - .00647	1.01048 to 1.02342	1.00402 to 1.02988	.99755 to 1.03635	18300	18.2226
47	1.01332	+ or - .00682	1.00650 to 1.02014	.99967 to 1.02697	.99285 to 1.03379	16800	19.2924
52	1.01363	+ or - .00682	1.00681 to 1.02045	1.00000 to 1.02726	.99318 to 1.03408	15300	20.3986
57	1.01228	+ or - .00751	1.00477 to 1.01979	.99726 to 1.02731	.98974 to 1.03482	13800	20.2128
62	1.00982	+ or - .00776	1.00207 to 1.01758	.99431 to 1.02534	.98655 to 1.03310	12300	22.7435
67	1.00828	+ or - .00830	.99998 to 1.01658	.99168 to 1.02488	.98337 to 1.03318	10800	25.4805
72	1.00519	+ or - .00906	.99614 to 1.01425	.98708 to 1.02331	.97803 to 1.03236	9300	27.6581
77	.99953	+ or - .00972	.98981 to 1.00924	.98009 to 1.01896	.97038 to 1.02867	7800	28.2542
82	.99388	+ or - .01097	.98291 to 1.00485	.97194 to 1.01582	.96097 to 1.02679	6300	31.7664
87	.99721	+ or - .01267	.98454 to 1.00988	.97187 to 1.02255	.95920 to 1.03522	4800	33.3090
92	1.00796	+ or - .01442	.99354 to 1.02238	.97912 to 1.03681	.96469 to 1.05123	3300	49.1956
97	1.03128	+ or - .01785	1.01343 to 1.04912	.99558 to 1.06697	.97773 to 1.08482	1800	60.5636


sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)
plot of average k-effective by generation run.
the line represents k-eff = 1.0159 + or - .0055 which occurs for 103 generations run.
.9575
.9978
1.0382
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 line represents k-eff = 1.0173 + or - .0058 which occurs for 8 generations skipped.
.9910
1.0123
1.0336
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
k-effective satisfies the chi**2 test for normality at the 95 % level

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	.0123			1.24655E-02	6.6666	3.88820E-03	6.1950	7.83278E-03	5.9845
2	.0625			6.35159E-02	2.1827	2.33323E-02	2.0541	6.08748E-02	2.3655
3	.0793			8.05211E-02	2.3803	3.11063E-02	2.3604	7.25412E-02	2.1864
4	.0458			4.65431E-02	2.7027	1.89026E-02	2.6884	4.09507E-02	2.6661
5	.0677			6.87411E-02	2.2946	2.92850E-02	2.2917	5.64617E-02	2.3303
6	.0962			9.77272E-02	1.9496	4.46347E-02	1.9449	8.60874E-02	1.7621
7	.0750			7.62107E-02	2.2772	3.78283E-02	2.2767	6.36809E-02	2.1848
8	.0217			2.20593E-02	3.4673	1.23315E-02	3.4638	2.74690E-02	3.5364
9	.0096			9.77563E-03	4.9486	5.62997E-03	4.9260	1.53279E-02	3.8018
10	.0132			1.33839E-02	3.5625	7.86698E-03	3.5310	1.29909E-02	5.5904
11	.0218			2.21227E-02	2.1970	1.39416E-02	2.1617	1.09830E-02	5.1337
12	.0248			2.52000E-02	1.9682	1.65951E-02	1.9094	7.98579E-03	6.7602
13	.0216			2.19422E-02	1.9776	1.64050E-02	1.8446	6.23545E-03	6.5188
14	.0179			1.81605E-02	1.9966	1.54396E-02	1.8081	7.31300E-03	6.3897
15	.0042			4.27608E-03	4.7513	3.03728E-03	4.3635	3.26536E-03	9.7744
16	.0027			2.79214E-03	6.6016	1.66086E-03	5.9505	1.70334E-03	12.3468
17	.0038			3.83612E-03	3.7370	2.21248E-03	3.6515	5.05704E-04	21.1090
18	.0047			4.77205E-03	3.7644	2.44824E-03	3.7120	4.63957E-04	25.5494
19	.0059			5.97526E-03	3.1278	2.92542E-03	3.0292	1.32827E-03	13.6647
20.	.0243			2.47236E-02	1.9082	1.21004E-02	1.8661	3.62457E-03	8.3223
21	.0125			1.26708E-02	3.0670	6.49619E-03	3.0389	9.44617E-04	15.5483
22	.0275			2.79321E-02	2.1938	1.48221E-02	2.1833	1.72465E-03	10.5998
23	.0718			7.29737E-02	1.7029	3.76834E-02	1.6914	3.53760E-03	7.3288
24	.0974			9.89196E-02	1.5358	4.99746E-02	1.5292	3.61187E-03	6.8975
25	.0708			7.19368E-02	1.5435	3.62318E-02	1.5338	1.85105E-03	8.8398
26	.0777			7.89842E-02	1.8086	3.98216E-02	1.8039	1.64156E-03	10.5867
27	.0273			2.77547E-02	2.7373	1.40494E-02	2.7153	4.04768E-04	20.5533
system total =				1.01592E+00	.5413	5.00651E-01	.5574	5.01342E-01	.5180

elapsed time 1.10933 minutes

random number= 0000760C32983B88

```
sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)
*****************************************************************************************************************************************

 position k-effective=  1.01345E+00  + or -  1.04790E-02
 the position k-effective is the largest eigenvalue of the fission production by position index matrix.
*****************************************************************************************************************************************

elapsed time   1.10933 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) 9.01E-01  ( 1,  2) 1.86E-01  ( 1,  3) 0.00E+00

  ( 2,  1) 1.04E-01  ( 2,  2) 8.42E-01  ( 2,  3) 0.00E+00

  ( 3,  1) 0.00E+00  ( 3,  2) 0.00E+00  ( 3,  3) 0.00E+00
```

sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)

source vector by position index

index	vector
1	4.79801E-01
2	5.20199E-01
3	0.00000E+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 8.69919E-01 + or - 7.11086E-03

```
sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)
************************************************************************************************************************************************************

 unit k-effective=  1.01345E+00  + or -  1.04790E-02
 the unit k-effective is the largest eigenvalue of the fission production by unit number matrix.
************************************************************************************************************************************************************

elapsed time   1.10933 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) 8.42E-01  (  1,  2) 1.04E-01  (  1,  3) 0.00E+00  (  1,  4) 0.00E+00  (  1,  5) 0.00E+00
  (  2,  1) 1.86E-01  (  2,  2) 9.01E-01  (  2,  3) 0.00E+00  (  2,  4) 0.00E+00  (  2,  5) 0.00E+00
  (  3,  1) 0.00E+00  (  3,  2) 0.00E+00  (  3,  3) 0.00E+00  (  3,  4) 0.00E+00  (  3,  5) 0.00E+00
  (  4,  1) 0.00E+00  (  4,  2) 0.00E+00  (  4,  3) 0.00E+00  (  4,  4) 0.00E+00  (  4,  5) 0.00E+00
  (  5,  1) 0.00E+00  (  5,  2) 0.00E+00  (  5,  3) 0.00E+00  (  5,  4) 0.00E+00  (  5,  5) 0.00E+00
```

sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)

source vector by unit

unit	vector
1	5.20199E-01
2	4.79801E-01
3	0.00000E+00
4	0.00000E+00
5	0.00000E+00

average self multiplication by unit

the number of next generation neutrons produced in a unit by
a neutron born in that same unit is 8.69919E-01 + or - 5.02814E-03

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	1.327E-01	1.20	5.308E-01
	2	0.000E+00	.00	0.000E+00
	3	0.000E+00	.00	0.000E+00
2	1	1.213E-01	1.26	4.851E-01
	2	0.000E+00	.00	0.000E+00
3	1	0.000E+00	.00	0.000E+00
4	1	0.000E+00	.00	0.000E+00
	2	0.000E+00	.00	0.000E+00
global unit				
5	1	0.000E+00	.00	0.000E+00

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	1.936E-02	8.05	4.241E-03	12.04	7.563E-03	8.78
2	1.773E-01	2.61	4.086E-02	3.69	7.166E-02	3.45
3	2.110E-01	2.09	5.217E-02	3.13	8.127E-02	2.43
4	1.207E-01	2.13	2.860E-02	3.56	4.843E-02	3.59
5	1.707E-01	1.84	4.018E-02	3.24	6.887E-02	3.40
6	2.627E-01	1.46	6.492E-02	2.44	1.058E-01	2.40
7	2.279E-01	1.29	5.955E-02	2.39	8.608E-02	2.67
8	1.647E-01	1.44	3.641E-02	2.51	5.073E-02	3.32
9	1.243E-01	1.68	2.404E-02	3.34	2.905E-02	4.28
10	1.107E-01	1.62	2.109E-02	2.97	2.603E-02	4.76
11	1.035E-01	1.53	1.994E-02	3.14	2.181E-02	4.71
12	6.447E-02	1.70	1.253E-02	3.94	1.567E-02	6.10
13	5.359E-02	1.69	9.955E-03	4.18	1.301E-02	5.77
14	5.395E-02	1.85	1.036E-02	4.03	1.401E-02	5.44
15	2.484E-02	2.43	4.633E-03	5.52	5.885E-03	8.15
16	1.475E-02	3.25	2.962E-03	7.14	3.446E-03	10.19
17	6.547E-03	4.05	1.128E-03	10.35	1.263E-03	18.39
18	5.177E-03	3.53	7.991E-04	13.97	1.047E-03	20.38
19	9.996E-03	3.10	1.724E-03	7.78	2.640E-03	13.36
20	3.251E-02	1.95	6.452E-03	4.25	8.227E-03	8.06
21	9.356E-03	3.00	1.902E-03	7.71	2.183E-03	13.61
22	1.528E-02	2.52	3.056E-03	6.88	3.711E-03	9.90
23	3.657E-02	1.71	8.851E-03	4.06	9.758E-03	6.33
24	3.254E-02	1.68	9.871E-03	4.42	1.127E-02	6.36
25	1.658E-02	1.68	6.142E-03	4.89	6.634E-03	6.83
26	1.237E-02	2.07	5.194E-03	5.76	5.511E-03	8.59
27	2.245E-03	3.29	1.235E-03	9.93	1.494E-03	14.98

fluxes for unit 2

group	region 1 flux	region 1 percent deviation	region 2 flux	region 2 percent deviation
1	1.203E-02	6.25	6.181E-03	7.72
2	1.026E-01	2.12	5.081E-02	3.26
3	1.286E-01	2.42	6.315E-02	2.83
4	8.009E-02	2.97	3.866E-02	3.42
5	1.228E-01	2.36	5.663E-02	2.48
6	1.955E-01	1.90	9.139E-02	2.17
7	1.369E-01	2.19	6.312E-02	2.64
8	2.626E-02	4.02	1.545E-02	4.96
9	6.403E-03	8.48	7.113E-03	8.48
10	3.387E-03	9.92	4.885E-03	9.77
11	2.375E-03	9.91	3.864E-03	10.69
12	1.726E-03	13.74	2.539E-03	13.62
13	1.522E-03	14.96	2.485E-03	14.55
14	1.573E-03	11.51	2.731E-03	13.60
15	5.363E-04	21.66	9.046E-04	23.10
16	1.946E-04	36.75	4.997E-04	26.74
17	8.655E-05	35.63	3.108E-04	39.22
18	7.330E-05	43.54	1.030E-04	52.15
19	1.520E-04	34.80	2.838E-04	30.54
20	4.567E-04	17.72	1.518E-03	16.56
21	1.608E-04	28.36	3.514E-04	32.36
22	2.633E-04	22.37	4.960E-04	28.56
23	7.266E-04	14.47	1.161E-03	12.76
24	7.437E-04	15.44	1.302E-03	14.52
25	3.899E-04	18.31	7.765E-04	19.39
26	2.024E-04	26.95	3.610E-04	22.70
27	1.118E-04	46.80	1.666E-04	38.36

fluxes for unit 3
region 1

group	flux	percent deviation
1	0.000E+00	.00
2	0.000E+00	.00
3	0.000E+00	.00
4	0.000E+00	.00
5	0.000E+00	.00
6	0.000E+00	.00
7	0.000E+00	.00
8	0.000E+00	.00
9	0.000E+00	.00
10	0.000E+00	.00
11	0.000E+00	.00
12	0.000E+00	.00
13	0.000E+00	.00
14	0.000E+00	.00
15	0.000E+00	.00
16	0.000E+00	.00
17	0.000E+00	.00
18	0.000E+00	.00
19	0.000E+00	.00
20	0.000E+00	.00
21	0.000E+00	.00
22	0.000E+00	.00
23	0.000E+00	.00
24	0.000E+00	.00
25	0.000E+00	.00
26	0.000E+00	.00
27	0.000E+00	.00

fluxes for unit 4

group	region 1 flux	region 1 percent deviation	region 2 flux	region 2 percent deviation
1	0.000E+00	.00	2.634E-02	10.16
2	0.000E+00	.00	2.284E-01	4.20
3	0.000E+00	.00	2.791E-01	3.80
4	0.000E+00	.00	1.567E-01	4.31
5	0.000E+00	.00	2.400E-01	3.82
6	0.000E+00	.00	3.648E-01	3.04
7	0.000E+00	.00	2.713E-01	3.05
8	0.000E+00	.00	9.770E-02	5.89
9	0.000E+00	.00	5.388E-02	8.49
10	0.000E+00	.00	3.566E-02	10.61
11	0.000E+00	.00	3.305E-02	11.25
12	0.000E+00	.00	2.370E-02	12.02
13	0.000E+00	.00	1.898E-02	13.74
14	0.000E+00	.00	2.018E-02	12.85
15	0.000E+00	.00	9.610E-03	18.29
16	0.000E+00	.00	5.653E-03	24.94
17	0.000E+00	.00	1.627E-03	40.56
18	0.000E+00	.00	1.575E-03	43.50
19	0.000E+00	.00	3.614E-03	27.28
20	0.000E+00	.00	1.252E-02	15.49
21	0.000E+00	.00	1.874E-03	35.37
22	0.000E+00	.00	3.835E-03	24.48
23	0.000E+00	.00	8.211E-03	16.59
24	0.000E+00	.00	1.047E-02	15.09
25	0.000E+00	.00	6.181E-03	20.99
26	0.000E+00	.00	3.063E-03	24.76
27	0.000E+00	.00	1.972E-03	41.78

fluxes for global unit
region 1

group	flux	percent deviation
1	0.000E+00	.00
2	0.000E+00	.00
3	0.000E+00	.00
4	0.000E+00	.00
5	0.000E+00	.00
6	0.000E+00	.00
7	0.000E+00	.00
8	0.000E+00	.00
9	0.000E+00	.00
10	0.000E+00	.00
11	0.000E+00	.00
12	0.000E+00	.00
13	0.000E+00	.00
14	0.000E+00	.00
15	0.000E+00	.00
16	0.000E+00	.00
17	0.000E+00	.00
18	0.000E+00	.00
19	0.000E+00	.00
20	0.000E+00	.00
21	0.000E+00	.00
22	0.000E+00	.00
23	0.000E+00	.00
24	0.000E+00	.00
25	0.000E+00	.00
26	0.000E+00	.00
27	0.000E+00	.00

```
sample problem 19 4 aqueous 4 metal array of arrays (samp prob 12)

                              frequency for generations    4 to  103
 .8759 to  .8990     *
 .8990 to  .9221     *****
 .9221 to  .9452     *****
 .9452 to  .9683     **********
 .9683 to  .9914     *********
 .9914 to 1.0145     ***************
1.0145 to 1.0376     ************************
1.0376 to 1.0607     ***********
1.0607 to 1.0838     **********
1.0838 to 1.1068     *******
1.1068 to 1.1299     **
1.1299 to 1.1530     **

                              frequency for generations   29 to  103
 .8759 to  .8990     *
 .8990 to  .9221     **
 .9221 to  .9452     ***
 .9452 to  .9683     ******
 .9683 to  .9914     ********
 .9914 to 1.0145     ************
1.0145 to 1.0376     ******************
1.0376 to 1.0607     ********
1.0607 to 1.0838     *********
1.0838 to 1.1068     ****
1.1068 to 1.1299     **
1.1299 to 1.1530     *

                              frequency for generations   54 to  103
 .8759 to  .8990     *
 .8990 to  .9221     **
 .9221 to  .9452     **
 .9452 to  .9683     ***
 .9683 to  .9914     *****
 .9914 to 1.0145     ***********
1.0145 to 1.0376     ************
1.0376 to 1.0607     *****
1.0607 to 1.0838     *****
1.0838 to 1.1068     ***
1.1068 to 1.1299     *
1.1299 to 1.1530

                              frequency for generations   79 to  103
 .8759 to  .8990     *
 .8990 to  .9221     **
 .9221 to  .9452     *
 .9452 to  .9683     **
 .9683 to  .9914     ***
 .9914 to 1.0145     ******
1.0145 to 1.0376     *******
1.0376 to 1.0607     *
1.0607 to 1.0838     *
1.0838 to 1.1068     *
1.1068 to 1.1299
1.1299 to 1.1530

*****************************************************************************************************************************************
          congratulations!  you have  successfully traversed the perilous path through keno vi in   1.10933 minutes
*****************************************************************************************************************************************
```

F17.F LIST OF KENO V.a VALIDATION REPORTS

This section contains a list of KENO validation reports in reverse chronological order. This list includes validation reports for KENO V.a and older versions of KENO. These reports form the basis of the validation reports to be done on KENO-VI.

ORNL-6512 — *Validation Studies Based on Data From Criticality Experiments Performed With Mixed Oxide Fuel Pins Moderated by Plutonium-Uranium-Gadolinium Nitrate Solutions*, G. R. Smolen, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab. (to be published).

ORNL-6511 — *Validation Studies Based on Data From Plutonium-Uranium Nitrate Critical Experiments Conducted in Annular Geometry*, G. R. Smolen, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab. (to be published).

ORNL-6510 — *Validation Studies Based on Data From Plutonium-Uranium Nitrate Critical Experiments Conducted in Slab and Cylindrical Geometries*, G. R. Smolen, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., December 1989.

NEACRP-L-306 — *Standard Problem Exercise on Criticality Codes for Dissolving Fissile Oxides in Acids*, OECD, Paris, France, 1989.

Y/DD-419 — *Validation Check Cases of SCALE77 on the ORGDP IBM-3083*, W. C. Jordan, H. R. Dyer, J. C. Turner, Martin Marietta Energy Systems, Inc., Oak Ridge Y-12 Plant, January 1989.

ORNL-6449 — Validation Studies Based on Data From Low Concentration Mixed Pu + U Aqueous Critical Experiments," G. R. Smolen, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., August 1988.
Notice: APPLIED TECHNOLOGY INFORMATION
Receipt of this report requires approval from:
D. E. Bailey
Director, Division of Fuels and Reprocessing
NE-551
U.S. Department of Energy
Washington, DC 20545

ORNL-6443 *Validation Studies Performed With Water- and Organic-Moderated and Reflected Mixed Oxide Fuel Pin Critical Experiments*, G. R. Smolen, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., June 1988.
Notice: APPLIED TECHNOLOGY INFORMATION
Receipt of this report requires approval from:
D. E. Bailey
Director, Division of Fuels and Reprocessing
NE-551
U.S. Department of Energy
Washington, DC 20545

ORNL/CSD/TM-242 *Recalculation of a Few Bare Plutonium Critical Arrays*, H. R. Dyer, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., April 1987.

ORNL/CSD/TM-238 *Validation of KENO-V.a Comparison with Critical Experiments*, W. C. Jordan, N. F. Landers, L. M. Petrie, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., December 1986.

ORNL/TM-9668 *Validation Studies for KENO-IV with Mixed Plutonium-Uranium Critical Experiments*, R. T. Primm III, Martin Marietta Energy Systems Inc., Oak Ridge Natl. Lab., November 1985.
Notice: APPLIED TECHNOLOGY INFORMATION
Receipt of this report requires approval from
D. E. Bailey
Director, Division of Fuels and Reprocessing
NE-551
U.S. Department of Energy
Washington, DC 20545

ORNL/CFRP-84/20 *Validation of the SCALE Code System and One Cross-Section Library for Plutonium and Gadolinium Solutions*, R. L. Sanders, University of Tennessee, July 1985.
Notice: APPLIED TECHNOLOGY INFORMATION
Receipt of this report requires approval from
D. E. Bailey
Director, Division of Fuels and Reprocessing
NE-551
U.S. Department of Energy
Washington, DC 20545

ORNL/CSD/TM-223 *Validation of KENO-V.a and Two Cross-Section Libraries for Criticality Calculations of Low-Enriched Uranium Systems*, M. E. Easter, Martin Marietta Energy Systems Inc., Oak Ridge Natl. Lab., July 1985.

ORNL/TM-9402 *Validation of the SCALE Code System and Two Cross-Section Libraries for Plutonium Benchmark Experiments,* M.E. Easter, University of Tennessee, R. T. Primm III, Martin Marietta Energy Systems Inc., Oak Ridge Natl. Lab., January 1985.
Notice: APPLIED TECHNOLOGY INFORMATION
Receipt of this report requires approval from
D. E. Bailey
Director, Division of Fuels and Reprocessing
NE-551
U.S. Department of Energy
Washington, DC 20545

ORNL/CSD/TM-224 *Assessment of Computational Performance in Nuclear Criticality,* L. M. Petrie, J. T. Thomas, Martin Marietta Energy Systems Inc., Oak Ridge Natl. Lab., January 1985.

ORNL/CSD/TM-221 *Validation of the Monte Carlo Criticality Program KENO-V.a for Highly Enriched Uranium Systems,* J. R. Knight, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., November 1984.

CSNI Report No. 78 *Standard Problem Exercise on Criticality Codes for Large Arrays of Packages on Fissile Materials,* CSNI Working Group, OECD, Paris, France, August 1984.

Trans. Am. Nucl. Soc. V44, p. 291-293 (US) (1983) "Validation of the NITAWL-KENO Methodology in Modeling New-Fuel Storage Criticality," D. G. Napolitano, Dr. Harris, P. F. Rose, E. Schmidt, E. Schmidt, M. Divadeennam, Yankee Atomic Electric Co., Framingham, Ma. 01701 (1983).

CSNI Report No. 71 *Standard Problem Exercise on Criticality Codes for Spent LWR Fuel Transport Containers,* CSNI Group of Experts on Nuclear Criticality Safety Computations, OECD, Paris, France, May 1982.

NUREG/CR-1917 *"Validation of Three Cross-Section Libraries Used with the SCALE System for Criticality Safety Analysis,"* ORNL/NUREG/CSD/TM-19, A. M. Hathout et al., U.S. Nuclear Regulatory Commission, June 1981.

Y-2234 — *Validation of the Monte Carlo Criticality Program KENO-IV and the Hansen-Roach "Sixteen-Energy Group Cross-Sections for High Assay Uranium Systems,"* G. R. Handley, L. C. Masters, R. V. Stachowiak, Union Carbide Corp., Nuclear Div., Oak Ridge Y-12 Plant, April 1981.

Trans. Am. Nucl. Soc. V27, p. 406-407 US (1977) — "Validation of Criticality Safety Broad-Group Library Using Uranium Systems," N. F. Cross, R. M. Westfall, K. R. Turnbull, P. B. Fox, Union Carbide Corp., Nuclear Div., Oak Ridge Natl. Lab. (1977).

Y-1948 — *"Validation of the KENO code for Nuclear Criticality Safety Calculations of Moderated, Low-Enriched Uranium Systems,"* G. R. Handley and C. M. Hopper, Union Carbide Corp., Nuclear Div., Oak Ridge Y-12 Plant, 1974.

Y-1858 — *Validation Checks of the ANISN and KENO Codes by Correlation with Experimental Data,"* G. R. Handley and C. M. Hopper, Union Carbide Corp., Oak Ridge Y-12 Plant, 1972.

NRC FORM 335 (2-89) NRCM 1102 201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See Instructions on the reverse)

1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, If any.)

NUREG/CR-0200, Rev. 6
ORNL/NUREG/CSD-2R6
Vol. 2, Part 3

2. TITLE AND SUBTITLE

SCALE: A MODULAR CODE SYSTEM FOR PERFORMING STANDARDIZED COMPUTER ANALYSES FOR LICENSING EVALUATION

Functional Modules
F16 – F17

3. DATE REPORT PUBLISHED

MONTH	YEAR
May	2000

4. FIN OR GRANT NUMBER

B0009

5. AUTHOR(S)

S.M. Bowman, B.L. Broadhead, C.B. Bryan,* J.A. Bucholz, K.W. Childs, A.L. Edwards,* M.B. Emmett, P.B. Fox, S.K. Fraley,* G.E. Giles, N.M. Greene, O.W. Hermann, T.J. Hoffman,* D.F. Hollenbach, W.C. Jordan, J.R. Knight,* N.F. Landers,* L.C. Leal, K. Lucius,* C.V. Parks, L.M. Petrie, J.C. Ryman, J.S. Tang,* J.C. Turner,* J.T. West,* R.M. Westfall, P.T. Williams

*Formerly with ORNL

6. TYPE OF REPORT

Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION — NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; If contractor, provide name and mailing address.)*

Oak Ridge National Laboratory
Post Office Box 2008
Oak Ridge, Tennessee 37831-6370

9. SPONSORING ORGANIZATION — NAME AND ADDRESS *(If NRC, type "Same as above"; If contractor, provide NRC Division, Office or Region, U.S. Regulatory Commission, and mailing address.)*

Spent Fuel Project Office
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

11. ABSTRACT *(200 words or less)*

SCALE, a modular code system for Standardized Computer Analyses Licensing Evaluation, has been developed by Oak Ridge National Laboratory at the request of the U.S. Nuclear Regulatory Commission. The SCALE system utilizes well-established computer codes and methods within standard analysis sequences that (1) allow an input format designed for the occasional user and/or novice, (2) automate the data processing and coupling between modules, and (3) provide accurate and reliable results. System development has been directed at problem-dependent cross-section processing and analysis of criticality safety, shielding, heat transfer, and depletion/decay problems. Since the initial release of SCALE in 1980, the code system has been heavily used for evaluation of nuclear fuel facility and package designs. This revision documents Version 4.4 of the system.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*

SCALE, cross sections, criticality safety, shielding, heat transfer, depletion, decay, spent fuel, KENO, ORIGEN, XSDRNPM, MORSE, NITAWL, XSDRN, SAS1, SAS2, SAS3, SAS4, CSAS, ORIGEN-ARP, QADS, HEATING, HTAS1

13. AVAILABILITY STATEMENT

Unlimited

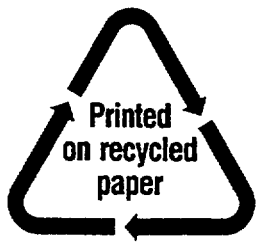
14. SECURITY CLASSIFICATION

(This Page) Unclassified

(This Report) Unclassified

15. NUMBER OF PAGES

16. PRICE



Federal Recycling Program