LA-5138

C.3

DIRECT: A Neutron-Transport Cross-Section and Trial-Solution Data Retrieval System

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

Printed in the United States of America. Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.95

LA-5138
UC-32
ISSUED: February 1973

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

DIRECT: A Neutron-Transport Cross-Section and Trial-Solution Data Retrieval System

by

K. L. Walters
R. E. Alcouffe

Work supported by the U.S. AEC Division of Reactor Development and Technology.

DIRECT: A NEUTRON-TRANSPORT CROSS-SECTION AND TRIAL-SOLUTION DATA RETRIEVAL SYSTEM

by

K. L. Walters and R. E. Alcouffe

ABSTRACT

The generalized diffusion method for studying transport of particles in a multidimensional system requires extensive cross-section and trial-solution data. Through the use of numerous computer filesets, DIRECT supplies the needed data base, provides software protection for both itself and the data, and eliminates cross-section and trial-solution data tapes. Additional software protection is provided through extensive FØRTRAN STØP statements and through fileset manipulation via the control card deck. The master data base can be updated easily through a variety of input options, but only the number and location of individual data sets can be changed. No alterations of any kind are performed on the data sets.

I. INTRODUCTION

When using the generalized diffusion equation method (GDE)[1] to study the transport of particles in a multidimensional system, it is convenient to have a library of multigroup cross sections plus a library of one-dimensional trial solutions consisting of the fluxes and currents. For complex systems, these data are extensive, so there is need for a method for cataloguing and updating them, as well as for a data directory to help the user select the proper cross sections and trial solutions. To this end, the program DIRECT has been written to store and update the required data and associated directories on both the CDC 6600 and 7600 computers.

DIRECT allows for diskfile storage and manipulation of large blocks of cross-section and trial-solution data. Requiring less than 1 minute of central processor time and 61,000 octal words of storage, DIRECT manipulates the master fileset STØRE which can then be used in place of cross-section and trial-solution magnetic tapes normally used by the GDE method. No alterations of any kind are performed on the data sets; only their number and position can be changed.

II. DISK STORAGE

DIRECT uses several filesets stored on disks to perform the necessary manipulations. Their purpose and construction are as follows.

A. PSTØRE. Fileset PSTØRE contains the computer code in UPDATE format,[2] complete with a slightly altered version of the LASL routine CRØS76.[3] The alteration consists of a dummy variable instead of the normal numerical reference to the physical-record unit size. This allows the code to convert the disk-stored equivalent to CDC 7600 tape or tapes without becoming confused by the numerical reference to the magnetic tape physical-record size normally found in CRØS76. The need for this conversion will be discussed in detail later.

B. STØRE. Fileset STØRE contains the cross-section and trial-solution data sets along with the associated directories on which PSTØRE operates in the following order: (1) cross-section directory, (2) cross-section data, (3) trial-solution directory, and (4) trial-solution data. Each of these blocks is followed by a filemark to facilitate file manipulation. Because the directories are built from data blocks (2) and (4), STØRE can be constructed

from empty files for blocks (1) and (3), and the first program execution for each major option (see Section V, A.1) will generate the appropriate directory. The exact nature of the manipulations for the cross-section and trial-solution data sets differs; therefore, two sets of control cards are required, depending on which kind of data is being handled. (Most file manipulation is handled externally by the use of appropriate control cards.)

Directories are constructed by extracting title portions from the respective data blocks. Cross-section data are read, and the title portion is then copied to both the new directory, tape 7, and the new data block, tape 6. Cross-section data blocks are copied to tape 6. Each cross-section portion consists of two records, one single card title and one large cross-section data record. Trial-solution data also use this scheme, with titles being transcribed to both tape 3, the directory, and tape 5, the data set. Here, however, three records, containing in order (1) a single card title, (2) flux data, and (3) current data, are used.

C. WHYNØT. Fileset WHYNØT contains the diskfile equivalent of one or more CDC 7600 trial-solution magnetic tapes generated by the GDE code. This is done for several reasons. First, because the tape or tapes must be mounted once anyway, this eliminates any future mountings and can be done by a short, simple, tape-copying job (see Appendix A). Second, because the diskfile equivalent of the tape is readily available, one can make a preliminary run with DIRECT as a test and verification without making the changes permanent on STØRE. This allows one to inspect the changes requested and to double check that the actual changes requested were those desired, an option that is particularly useful to the new user. Finally, because the code was originally intended primarily for a CDC 6600, where it now resides, allowance had to be made for the 7600 data tapes.

D. GLITCH. Fileset GLITCH contains the cross-section data sets from which data are to be added or transferred to STØRE. GLITCH serves the same basic function for cross-section data that WHYNØT does for trial-solution information.

Actual fileset construction, both of binary filesets like STØRE and GLITCH, and UPDATE filesets like PSTØRE, depends on the computer system being used. In the specific case of the LASL filesets mentioned, details can be obtained from the LASL Programmer's Information Manuals. The method for constructing WHYNØT is shown in Appendix A.

Note that DIRECT tries to guard the master fileset STØRE against operator error. This was briefly alluded to in the discussion of the data filesets WHYNØT and GLITCH and will be expanded now.

III. SOFTWARE PROTECTION

Two forms of disk-file software protection are provided to aid the user. The first is the use of temporary filesets to handle all intermediate processing, as mentioned previously. Thus, by removing one of the control cards, as pointed out in the appendixes, one makes all required changes without effecting any permanent changes in the disk filesets. Though this increases the complexity of the control card deck, it guards against undesirable changes and is strongly recommended for all but the continuous user.

The second form of software protection is provided through extensive use of octal stops to cease program execution when an error is detected. (Octal numbers are used on FØRTRAN STØP statements on CDC 6600 and 7600 computers.) Because proper fileset positioning is critical, 63 octal STØP statements are provided to block execution of paths known not to conform to expected execution routes. These stops are used to indicate several possible errors, such as trying to overstore a dimensioned variable, improperly leaving a DØ loop, and encountering an end of file mark in a read where none was expected. The approximate location of all stops is indicated in the comment part of the code (see Appendix E), along with some indication of the potential trouble. If, for instance, one encounters a STØP 15, he can easily determine that the location is in the subsection of the subroutine CRØSS following the title CAUSING DATA SET DELETIØN. With this technique and the use of the temporary filesets, the code attempts to guard its disk filesets from damage.

IV. CONTROL CARDS

Execution of DIRECT to alter the cross-section data sets or the trial-solution data blocks involves fundamentally different operations, as indicated earlier. Therefore, two different sets of control

cards are required. The significance of the master filesets involved has already been discussed. The comments added to the individual control cards illustrate their purpose (see Appendixes B and C). In addition, a brief discussion follows.

In updating a cross-section data set, numerous temporary filesets are used to protect the master filesets. In addition to filesets 8 and 11 used by the code software directly, six others, Nos. 6, 7, 9, 10, 12, and 30, are referenced in the control card deck. Their use is as follows.

Temporary filesets 6 and 7 are produced during program execution, and contain the new cross-section data set and directory, respectively. Fileset 10 is a composite of 6 and 7 in final format, and is assembled totally through the control card deck from the various other files. Tape 12 is simply the working copy of GLITCH, and tape 30 is a dummy file used for proper positioning of the other filesets.

Tape 30 deserves special consideration. As can be seen from the example in Appendix B, its first real use is to position STØRE past the old cross-section directory before execution. This allows DIRECT to construct the new cross-section directory from the new data. Thus if a mistake was made in the old directory it will not be saved in the new version.

The trial-solution update procedure uses temporary filesets 3, 5, 9, 10, 30, and 40. Three and five, produced by the code, contain the trial-solution directory and data set. Ten and thirty serve as before, and forty is the temporary copy of WHYNØT.

Two other data files, ØUTPUT and tape 9, are of interest. Tape 9 is best explained after discussion of the Input Parameters. ØUTPUT contains a partial indication of job progress followed by a copy of the various directories and the contents of tape 9. If printing is done at a remote terminal, this allows examination of the initial portion to ensure proper code execution, with the bulk of the printing then being diverted to the central printers.

V. INPUT PARAMETERS

Because two basically different operations are being performed on STØRE, the exact nature of the input parameters depends on what options are being executed, as well as whether cross-section or trial-solution information is being manipulated. The input parameters are discussed in the next four subsections; those in the first are common to any execution.

A. Input Cards Common to All Decks. The first three sets of cards specify (1) which major part of the code is being executed, (2) a job title, and (3) the ISKIP (I) and IØPTIØN (I) parameters needed.

1. Card 1, FORMAT 15. A number of 1 or less causes execution of the trial-solution data manipulator. Any value equal to or greater than 2 causes manipulation of cross-section data.

2. Card 2, FØRMAT 8A10. Title desired. Printed at the beginning of the output file.

3. Sequence of cards 3, 4, ..., FØRMAT (1X, I4, I5, 2X, I3). ISKIP (I), IØPTIØN (I), and TAPENØ parameters. ISKIP is the sequence number of the cross-section or trial-solution entry to be altered. Entries must occur in increasing number to avoid a STØP 4. IØPTIØN values indicate which of the four possible changes is to occur at the corresponding ISKIP number. (CAUTIØN: IØPTIØN = 4 will mismatch directory entries and data unless used correctly. This is primary reason for making preliminary checkout runs.) Reading is terminated when both ISKIP and IØPTIØN are blank. On this card, TAPENØ is entered. For cross-section manipulations using data obtained from GLITCH, as well as any trial-solution execution, TAPENØ must be set equal to 12. Only when cross-section changes are desired and all the necessary information is in the input stack should TAPENØ be set equal to 2.

Four basic operations can be performed on data fileset STØRE as specified by input parameters ISKIP (I) and IØPTIØN (I). The significance of the allowable IØPTIØN values is as indicated below, with any other value resulting in an octal stop.

ISKIP (I) specifies the sequential position of the information contained in fileset STØRE, either cross-section or trial-solution data, to be changed. IØPTIØN (I) indicates which type of change is desired at the corresponding ISKIP (I) value. Up to 49 ISKIP/IØPTIØN pairs can be specified sequentially (see Appendix B) using essentially a 1X, I4, I5 format.

Data set deletions at a specific ISKIP (I) value are obtained by setting IØPTIØN (I) = 1. The subroutine section entitled CAUSING DATA SET DELETION

removes data by positioning STØRE past indicated sets without transcription to any of the important temporary filesets. (Filesets 3, 5, 7, and 6 contain trial-solution and cross-section directories and data blocks to be saved.)

Title changes at a specific ISKIP (I) value are obtained by setting IØPTIØN (I) = 2. The subroutine section entitled CAUSING TITLE CHANGE positions STØRE past the title to be dropped and matches the new title to the old data set.

New cross-section titles can be supplied from physical tapes by setting TAPENØ = 12 and specifying through the RECORD (I) parameter the new title's sequential position in the physical tape. Alternatively, for cross-section titles from cards, TAPENØ is set equal to 2 and the title card (singular) is put in the appropriate position in the input deck. Trial-solution titles must be supplied from cards with TAPENØ = 12. Detailed examples are shown in Appendixes B, C, and D.

Data set replacements at a specific ISKIP (I) value are obtained by setting IØPTIØN (I) = 3. The subroutine section entitled CAUSING DATA SET REPLACEMENT simply positions STØRE past the information to be dropped and inserts the new title and data at this position. A new cross-section title and data can be supplied from the input deck by setting TAPENØ = 2 and appropriately positioning the new data in the input stream (see Appendix D). Either type of input data can be supplied from physical tapes by specifying through the RECORD (I) parameter the desired data's sequential position on the tape and setting TAPENØ = 12 (see Appendixes B and C).

Addition of new data sets at a specific ISKIP (I) value is obtained by setting IØPTIØN (I) = 4. The subroutine section entitled CAUSING ADDITION OF DATA SET AT THE END OF THE OLD DIRECTORY simply transmits the new data to the next position in the temporary fileset. (CAUTIØN: IØPTIØN = 4 will mismatch directory entries and data unless used correctly. As the resulting error is not immediately obvious, one should make a preliminary checkout first.) Data additions here are done exactly like those just discussed.

B. Cross-Section Interaction with Input File. If all changes to be made on STØRE are from data contained in the program data deck, the sequence of information required after TAPENØ is specified is that implied in the changes requested. Consider, for example, ISKIP (I) = 1, 10, 15, and 24 with corresponding IØPTIØN (I) = 1, 2, 3, 4 (Appendix D). This sequence of instructions indicates (1) a data-set deletion at position 1, (2) a title change at position 10, (3) a data-set replacement at position 15, and (4) an addition at (new) position 24. The corresponding data cards must therefore be (1) the new title for position 10, (2) the new title and data set to replace position 15, and (3) the new title and data set added at position 24. All titles are read using an 8A10 format with 6E12.5 formats for the data.

C. Cross-Section Interaction with Fileset GLITCH. As all required information is to be supplied from GLITCH, the sequence of information required after TAPENØ is specified is the position of the data sets in GLITCH from which information is to be taken. Read in by a 2X, I3 format and terminated by a blank card, the parameter RECØRD (I) accomplishes the desired changes.

RECØRD (I) when germane specifies up to 50 sequential positions of information contained on GLITCH or WHYNØT from which data are to be taken for changes to STØRE. Examples are shown in Appendixes B and C, and corresponding text description is given here and in Section D. Note that all changes are performed in subroutine CROSS, and that section headings in the following discussion correspond to comment card captions therein.

Consider the following example. ISKIP (I) = 12, 13, 18, 24; IØPTIØN (I) = 1, 2, 3, 4; and RECØRD (I) = 2, 10, 20. From this input, data set 12 is dropped. The new title for position 13 comes from GLITCH position 2. The new replacement data for position 18 come from GLITCH position 10, and the new position 24 comes from GLITCH position 20 (Appendix B).

D. Trial-Solution Changes. In changing the trial-solution data sets, a combination of the previous changes is used. Deletions are as before, but new single card titles, IØPTIØN (I) = 2 with 8A10 formats, are assumed to come from the input deck. Data replacements and additions at the end are obtained from WHYNØT at the RECORD (I) values referenced (Appendix C). TAPENØ is always set equal to 12 for trial-solution changes. Here again, except as noted above for titles, I/Ø is taken from the G. D. E. method.

E. Temporary Fileset Nine. A negative TAPENØ value causes a complete printout of this fileset containing all trial-solution or cross-section data present on STØRE at the end of job execution. In addition, some of the IØPTIØN parameters cause partial printout of this information. For trial-solution data, either a replacement or addition will cause the printout of that particular trial-solution data set on tape 9. In cross-section sets, title changes, replacements, or additions are printed out for the particular element referenced. If these data are not wanted, the corresponding control cards can be removed.

REFERENCES

1. R. E. Alcouffe, "Generalized Finite-Differenced Diffusion Equation for Neutron-Transport Computations," Los Alamos Scientific Laboratory report LA-4938-MS (July 1972).
2. A. M. Salem, Los Alamos Scientific Laboratory, personal communication, May 1971.
3. J. F. Bem, Los Alamos Scientific Laboratory, personal communication, May 1971.

APPENDIX A

FILESET WHYNØT

Fileset WHYNØT contains the diskfile equivalent of one or more CDC 7600 trial-solution magnetic tapes. Constructed as indicated below, WHYNØT requires that the first record on each tape, the tape label, be copied to a dummy file so that only the actual data are transmitted. Note that the alphanumerics, B13, B14, B15, and B16, on the right of the assigned cards are comments indicating the sequence number of the problem from which the data originated.

```
$JOB(NAME=KKWALT,CL=U,TL=5M,PL=100,UA=8111D463,CAT=02,AC=C11KLW,SC=*600008)
ASSIGN AB,WHYNOT(,C11WHYNOT)
REWIND(DUMMY)
ASSIGN MT,TAPE1(PUL,XX004894,SMB)  B 13
COPYCF(TAPE1,DUMMY)
COPYBF(TAPE1,WHYNOT)
RELTAPE(TAPE1)
ASSIGN MT,TAPE3(PUL,XX005568,SMB)  B15
COPYCF(TAPE3,DUMMY)
COPYBF(TAPE3,WHYNOT)
RELTAPE(TAPE3)
ASSIGN MT,TAPE4(PUL,XX006242,SMB)  B16
COPYCF(TAPE4,DUMMY)
COPYBF(TAPE4,WHYNOT)
RELTAPE(TAPE4)
ASSIGN MT,TAPE5(PUL,XX003031,SMB)  B17
COPYCF(TAPE5,DUMMY)
COPYBF(TAPE5,WHYNOT)
RELTAPE(TAPE5)
REWIND(WHYNOT)
CATALOG(WHYNOT,,0)
000000000000000000000
000000000000000000000
```

APPENDIX B
FILESET GLITCH

The control card deck used in changing the cross-section data from fileset GLITCH is as follows. Comments added to the individual control cards illustrate their respective purposes.

```
$JOB(NAME=C11DIRECT,CL=U,TL=1M,PL=50,UA=8111D463,AC=C11KLW,SC=61000B)
ASSIGN AB,PSTORE(,C11PSTORE)          ASSIGNING AND
UPDATE(F,P=PSTORE,L=DUM)              UPDATING PROGRAM FILESET.
ASSIGN AB,STORE(,C11STORE)                 ASSIGNING MASTER DATA FILESET.
ASSIGN AB,GLITCH(,C11GLITCH)          ASSIGNING CROSS-SECTIONS.
REWIND(TAPE30)                             ASSIGNING TEMPORARY
REWIND(TAPE12)                             WORKING FILESETS.
COPYCR(GLITCH,TAPE12)                 SETTING UP WORKING COPY
REWIND(TAPE12)                        OF CROSS-SECTIONS.
COPYBF(STORE,TAPE30,1)                     SKIPPING OLD CROSS-SECTION DIRECTORY.
MAP(OFF)
RUN(G,I=COMPILE)
REWIND(STORE)
COPYBF(STORE,OUTPUT,1)                LISTING OLD CROSS-SECTION DIRECTORY.
COPYBF(STORE,TAPE30,1)                     SKIPPING OLD CROSS-SECTIONS.
REWIND(TAPE7)                         POSITIONING NEW CROSS-SECTION DIRECTORY AND
REWIND(TAPE10)                        MASTER TEMPORARY FILESET.
COPYBF(TAPE7,TAPE10,1)                     SAVING NEW DIRECTORY.
COPYBF(TAPE6,TAPE10,1)                SAVING NEW CROSS-SECTION DATA SET.
COPYBF(STORE,TAPE10,1)                     SECOND (UNALTERED) DIRECTORY AND
COPYBF(STORE,TAPE10,1)                     DATA SET SAVED.
REWIND(TAPE10)
COPYSBF(TAPE10,OUTPUT,1)              TO EXAMINE NEW CROSS-SECTION DIRECTORY.
COPYBF(TAPE10,TAPE30,1)                    OMITS FIRST DATA SET FROM LISTING ONLY.
COPYSBF(TAPE10,OUTPUT,1)              PRINTS OUT SECOND DIRECTORY.
REWIND(TAPE10)                             SETTING UP FILESETS TO SAVE TEMPORARY
REWIND(STORE)                              COPY IN TAPE10.
COPYBF(TAPE10,STORE,4)                PULL THIS CARD DURING TESTING.
REWIND(TAPE9)
COPYSBF(TAPE9,OUTPUT)
EXIT.
REWIND(TAPE7)  **********
REWIND(STORE)            CROSS-SECTION DATA CONTROL CARDS
COPYSBF(STORE,OUTPUT,1)                                        **********
COPYSBF(TAPE7,OUTPUT)
REWIND(TAPE9)
COPYSBF(TAPE9,OUTPUT)                      PULL TO ELIMINATE FILESET NINE PRINTOUT.
0000000000000000000000
0000000000000000000000
    2
 TAPE/PERMFILE EXAMPLE PROBLEM.
   12    1
   13    2
   18    3
   24    4
            -12
    2
   10
   20

0000000000000000000000
0000000000000000000000
```

APPENDIX C

CHANGES TØ TRIAL SØLUTIØN DATA

The control card deck used in changing the trial-solution data is as follows. Comments added to the individual control cards illustrate their respective purposes.

```
$JOB(NAME=C11DIRECT,CL=U,TL=1M,PL=50,UA=8111D463,AC=C11KLW,SC=61000B)
ASSIGN AB,WHYNOT(,C11WHYNOT)          SETTING UP
REWIND(TAPE40)                        TEMPORARY COPY
COPYBF(WHYNOT,TAPE40,30)              OF PERMFILE
REWIND(TAPE40)                        WHYNOT.
ASSIGN AB,PSTORE(,C11PSTORE)               ASSIGNING AND UPDATING
UPDATE(F,P=PSTORE,L=DUM)                   PROGRAM FILESET.
REWIND(TAPE3)                         ASSIGNING TEMPORARY
REWIND(TAPE30)                        WORKING FILESETS.
ASSIGN AB,STORE(,C11STORE)                 ASSIGNING AND POSITIONING
COPYBF(STORE,TAPE30,3)                     MASTER DATA FILESET.
MAP(OFF)
RUN(G,I=COMPILE)
REWIND(TAPE3)                         INSURING
REWIND(TAPE5)                         PROPER
REWIND(TAPE10)                        FILESET
REWIND(STORE)                         POSITIONING.
COPYBF(STORE,TAPE10,2)                     SAVING UNALTERED CROSS-SECTION VALUES.
COPYBF(TAPE3,TAPE10,1)                SAVING NEW TRIAL SOLUTION DIRECTORY AND
COPYBF(TAPE5,TAPE10,1)                ASSOCIATED DATA SETS.
REWIND(TAPE10)                             PRINTING UNALTERED
COPYSBF(TAPE10,OUTPUT)                     CROSS-SECTION DIRECTORY.
COPYBF(TAPE10,TAPE30)                 SKIPPING UNALTERED DATA SET.
REWIND(STORE)
COPYBF(STORE,TAPE30,2)                     PRINTING OLD
COPYSBF(STORE,OUTPUT)                      TRIAL SOLUTION DIRECTORY.
COPYSBF(TAPE10,OUTPUT)                PRINTING NEW DIRECTORY.
REWIND(TAPE10)                             SETTING UP FILESETS TO SAVE TEMPORARY
REWIND(STORE)                              COPY IN TAPE10.
COPYBF(TAPE10,STORE,4)                PULL THIS CARD DURING TESTING.
REWIND(TAPE9)
COPYSBF(TAPE9,OUTPUT)
EXIT.
REWIND(TAPE3)  **********
COPYSBF(TAPE3,OUTPUT)     TRIAL SOLUTION DATA CONTROL CARDS
REWIND(TAPE9)                                          **********
COPYSBF(TAPE9,OUTPUT)                      PULL TO ELIMINATE FILESET NINE PRINTOUT.
0000000000000000000000
0000000000000000000000
   1
 TRIAL SOLUTION EXAMPLE PROBLEM.
   2    1
   3    2
   4    3
   5    4
   6    4
   7    4
          -12
   1
   2
   3
   4

 NEW TITLE FOR DIRECTORY ENTRY NUMBER 3.
0000000000000000000000
0000000000000000000000
```

APPENDIX D

CHANGES TØ CRØSS-SECTIØN DATA

The control card deck used to change the cross-section data from the input file is as follows.

```
$JOB(NAME=KKWALT,CL=U,TL=01M,PL=5,UA=8111D463,CAT=02,AC=C11KLW,SC=610008)
ASSIGN AB,PSTORE(,C11PSTORE)
UPDATE(F,P=PSTORE,L=DUM)
ASSIGN AB,STORE(,C11STORE)
ASSIGN AB,GLITCH(,C11GLITCH)
REWIND(TAPE30)
REWIND(TAPE12)
COPYCR(GLITCH,TAPE12)
REWIND(TAPE12)
COPYBF(STORE,TAPE30,1)
MAP(OFF)
RUN(G,I=COMPILE)
REWIND(STORE)
COPYBF(STORE,OUTPUT,1)
COPYBF(STORE,TAPE30,1)
REWIND(TAPE7)
REWIND(TAPE10)
COPYBF(TAPE7,TAPE10)  SAVING NEW DIRECTORY.                              *
COPYBF(TAPE6,TAPE10)  SAVING FIRST DATA SET.                             *
COPYBF(STORE,TAPE10)  SECOND (UNALTERED) DIRECTORY SAVED.                *
COPYBF(STORE,TAPE10)  SECOND (UNALTERED) DATA SET SAVED.
REWIND(TAPE10)  COMPLETED DIRECTORY WITH DATA SET.
COPYSBF(TAPE10,OUTPUT,1)  TO EXAMINE NEW CROSS-SECTION DIRECTORY.
COPYBF(TAPE10,TAPE30)  OMITTS FIRST DATA SET FROM OUTPUT LISTING.
COPYSBF(TAPE10,OUTPUT)  PRINTS OUT SECOND DIRECTORY.
REWIND(TAPE9)
COPYSBF(TAPE9,OUTPUT)
EXIT.
REWIND(TAPE7)  ************
REWIND(STORE)                CROSS-SECTION DATA
COPYSBF(STORE,OUTPUT,1)                          ************
COPYSBF(TAPE7,OUTPUT)
REWIND(TAPE9)
COPYSBF(TAPE9,OUTPUT)
0000000000000000000000
0000000000000000000000
   2
 ATTEMPTING TO CHECK ALL CROSS-SECTION DATA SET OPTIONS.
   1    1
  10    2
  15    3
  24    4
          -2
 NEW TITLE FOR MN CROSS-SECTION SET.
 ATTEMPTING TO REPLACE U238R CROSS-SECTION DATA WITH SM.
 0.          -4.28993E-01 0.           2.91842E+00 0.           0.
 0.           0.          0.           8.76561E-01 0.           0.
 0.           0.          0.           0.          3.74210E-02 0.
 3.90487E+00  0.          0.           0.          0.           0.
 2.28502E+00  6.87018E-01 0.           0.          0.           0.
 0.           9.28300E-02 0.           5.34422E+00 0.           0.
 0.           0.          0.           3.22301E+00 1.36013E+00  5.82052E-01
 0.           0.          0.           0.          1.96240E-01  0.
 6.96221E+00  0.          0.           0.          0.           0.
 5.32686E+00  1.83798E+00 2.22290E-01  7.89150E-01 0.           0.
 0.           5.34890E-01 0.           8.67875E+00 0.           0.
 0.           0.          0.           7.78168E+00 1.23362E+00  1.87950E-01
 0.           3.76522E-01 0.           0.          1.22140E+00  0.
 1.17949E+01  0.          0.           0.          0.           0.
 1.04734E+01  3.38304E-01 1.98350E-01  2.45450E-03 0.           3.61102E-02
 0.           2.93960E+00 0.           1.71257E+01 0.           0.
 0.           0.          0.           1.41161E+01 8.78800E-02  2.36460E-02
 7.14250E-03  0.          1.15117E-03  0.          6.34840E+00  0.
 2.89713E+01  0.          0.           0.          0.           0.
```

```
2.21024E+01 7.00360E-02 5.30420E-03 2.29500E-04 0.          1.64406E-05
0.          9.48520E+00 0.          3.62604E+01 0.          0.
0.          0.          0.          2.61605E+01 5.20490E-01 0.
3.21870E-03 0.          0.          0.          1.40650E+01 0.
4.56290E+01 0.          0.          0.          0.          0.
3.08408E+01 6.14660E-01 0.          0.          1.95090E-03 0.
0.          2.06710E+01 0.          5.76597E+01 0.          0.
0.          0.          0.          3.61442E+01 7.23210E-01 0.
0.          0.          1.10380E-03 0.          3.00940E+01 0.
7.31162E+01 0.          0.          0.          0.          0.
4.20439E+01 8.44530E-01 0.          0.          0.          4.29510E-04
0.          4.33780E+01 0.          9.29702E+01 0.          0.
0.          0.          0.          4.84710E+01 9.78310E-01 0.
0.          0.          1.58120E-04 0.          6.18890E+01 0.
1.18475E+02 0.          0.          0.          0.          0.
5.53123E+01 1.12120E+00 0.          0.          0.          5.81190E-05
0.          1.85450E+02 0.          6.22867E+02 0.          0.
0.          0.          0.          4.26306E+02 1.27320E+00 0.
0.          0.          2.13880E-05 0.          2.01200E+02 0.
3.66122E+02 0.          0.          0.          0.          0.
1.64922E+02 1.11110E+01 0.          0.          0.          7.86620E-06
0.          1.31880E+02 0.          1.56479E+02 0.          0.
0.          0.          0.          2.45964E+01 0.          0.
0.          0.          0.          0.          1.28970E+02 0.
1.41760E+02 0.          0.          0.          0.          0.
1.27900E+01 2.45600E-03 0.          0.          0.          0.
0.          4.15600E+02 0.          4.76893E+02 0.          0.
0.          0.          0.          6.11940E+01 0.          0.
0.          0.          0.          0.          1.21290E+02 0.
1.24369E+02 0.          0.          0.          0.          0.
2.22563E+00 9.93330E-02 0.          0.          0.          0.
0.          3.86790E+02 0.          3.98251E+02 0.          0.
0.          0.          0.          1.13090E+01 8.53060E-01 0.
0.          0.          0.          0.          1.73600E+01 0.
2.61463E+01 0.          0.          0.          0.          0.
8.55754E+00 1.52080E-01 0.          0.          0.          0.
0.          9.15390E+01 0.          1.06546E+02 0.          0.
0.          0.          0.          1.45019E+01 2.28760E-01 0.
0.          0.          0.          0.          4.73100E+03 0.
4.80713E+03 0.          0.          0.          0.          0.
7.57367E+01 5.04860E-01 0.          0.          0.          0.
0.          1.12390E+03 0.          1.14956E+03 0.          0.
0.          0.          0.          2.47435E+01 3.97070E-01 0.
0.          0.          0.          0.          2.76430E+04 0.
2.79461E+04 0.          0.          0.          0.          0.
2.82702E+02 9.15077E-01 0.          0.          0.          0.
0.          7.98930E+04 0.          8.04505E+04 0.          0.
0.          0.          0.          5.52595E+02 2.04191E+01 0.
0.          0.          0.          0.          4.35321E+04 0.
4.37254E+04 0.          0.          0.          0.          0.
1.91833E+02 4.96761E+00 0.          0.          0.          0.
0.          5.38764E+04 0.          5.40035E+04 0.          0.
0.          0.          0.          1.27115E+02 1.39886E+00 0.
0.          0.          0.
SM149
0.         -4.28993E-01 0.          2.91842E+00 0.          0.
0.          0.          0.          8.76561E-01 0.          0.
0.          0.          0.          0.          3.74210E-02 0.
3.90487E+00 0.          0.          0.          0.          0.
2.28502E+00 6.87018E-01 0.          0.          0.          0.
0.          9.28300E-02 0.          5.34422E+00 0.          0.
0.          0.          0.          3.22301E+00 1.36013E+00 5.82052E-01
0.          0.          0.          0.          1.96240E-01 0.
6.96221E+00 0.          0.          0.          0.          0.
5.32686E+00 1.83798E+00 2.22290E-01 7.89150E-01 0.          0.
0.          5.34890E-01 0.          8.67875E+00 0.          0.
0.          0.          0.          7.78168E+00 1.23362E+00 1.87950E-01
0.          3.76522E-01 0.          0.          1.22140E+00 0.
1.17949E+01 0.          0.          0.          0.          0.
```

```
1.04734E+01 3.38304E-01 1.98350E-01 2.45450E-03 0.          3.61102E-02
0.          2.93960E+00 0.          1.71257E+01 0.          0.
0.          0.          0.          1.41161E+01 8.78800E-02 2.36460E-02
7.14250E-03 0.          1.15117E-03 0.          6.34840E+00 0.
2.89713E+01 0.          0.          0.          0.          0.
2.21024E+01 7.00360E-02 5.30420E-03 2.29500E-04 0.          1.64406E-05
0.          9.48520E+00 0.          3.62604E+01 0.          0.
0.          0.          0.          2.61605E+01 5.20490E-01 0.
3.21870E-03 0.          0.          0.          1.40650E+01 0.
4.56290E+01 0.          0.          0.          0.          0.
3.08408E+01 6.14660E-01 0.          0.          1.95090E-03 0.
0.          2.06710E+01 0.          5.76597E+01 0.          0.
0.          0.          0.          3.61442E+01 7.23210E-01 0.
0.          0.          1.10380E-03 0.          3.00940E+01 0.
7.31162E+01 0.          0.          0.          0.          0.
4.20439E+01 8.44530E-01 0.          0.          0.          4.29510E-04
0.          4.33780E+01 0.          9.29702E+01 0.          0.
0.          0.          0.          4.84710E+01 9.78310E-01 0.
0.          0.          1.58120E-04 0.          6.18890E+01 0.
1.18475E+02 0.          0.          0.          0.          0.
5.53123E+01 1.12120E+00 0.          0.          0.          5.81190E-05
0.          1.85450E+02 0.          6.22867E+02 0.          0.
0.          0.          0.          4.26306E+02 1.27320E+00 0.
0.          0.          2.13880E-05 0.          2.01200E+02 0.
3.66122E+02 0.          0.          0.          0.          0.
1.64922E+02 1.11110E+01 0.          0.          0.          7.86620E-06
0.          1.31880E+02 0.          1.56479E+02 0.          0.
0.          0.          0.          2.45964E+01 0.          0.
0.          0.          0.          0.          1.28970E+02 0.
1.41760E+02 0.          0.          0.          0.          0.
1.27900E+01 2.45600E-03 0.          0.          0.          0.
0.          4.15600E+02 0.          4.76893E+02 0.          0.
0.          0.          0.          6.11940E+01 0.          0.
0.          0.          0.          0.          1.21290E+02 0.
1.24369E+02 0.          0.          0.          0.          0.
2.22563E+00 9.93330E-02 0.          0.          0.          0.
0.          3.86790E+02 0.          3.98251E+02 0.          0.
0.          0.          0.          1.13090E+01 8.53060E-01 0.
0.          0.          0.          0.          1.73600E+01 0.
2.61463E+01 0.          0.          0.          0.          0.
8.55754E+00 1.52080E-01 0.          0.          0.          0.
0.          9.15390E+01 0.          1.06546E+02 0.          0.
0.          0.          0.          1.45019E+01 2.28760E-01 0.
0.          0.          0.          0.          4.73100E+03 0.
4.80713E+03 0.          0.          0.          0.          0.
7.57367E+01 5.04860E-01 0.          0.          0.          0.
0.          1.12390E+03 0.          1.14956E+03 0.          0.
0.          0.          0.          2.47435E+01 3.97070E-01 0.
0.          0.          0.          0.          2.76430E+04 0.
2.79461E+04 0.          0.          0.          0.          0.
2.82702E+02 9.15077E-01 0.          0.          0.          0.
0.          7.98930E+04 0.          8.04505E+04 0.          0.
0.          0.          0.          5.52595E+02 2.04191E+01 0.
0.          0.          0.          0.          4.35321E+04 0.
4.37254E+04 0.          0.          0.          0.          0.
1.91833E+02 4.96761E+00 0.          0.          0.          0.
0.          5.38764E+04 0.          5.40035E+04 0.          0.
0.          0.          0.          1.27115E+02 1.39886E+00 0.
0.          0.          0.
00000000000000000000
00000000000000000000
```

APPENDIX E

DIRECT CODE LISTING

```
      PROGRAM DIRECT(INPUT=101,TAPE1=INPUT,STORE=101,TAPE2=STORE,TAPE3=1
     101,OUTPUT=101,TAPE4=OUTPUT,TAPE5=101,TAPE6=101,TAPE7=101,TAPE8=101
     2,TAPE9=101,TAPE11=101,TAPE12=513,TAPE40=513)
C
C     CARD ONE.  OPTION.  FORMAT(I5).
C     OPTION .LE. 1 FOR TRIAL SOLUTION DIRECTORY.
C            .GE. 2 FOR CROSS-SECTION DIRECTORY.
C
C     CARD TWO.  TITLE.  FORMAT(8A10)
C     TITLE FOR PARTICULAR JOB, NOT FOR THE DIRECTORY.
C
C
C     CARD THREE.  ISKIP(IP), IOPTION(I), TAPENO.  FORMAT(1X,I4,I5,2X,I3
C     TAPENO TAKEN OFF LAST CARD WITH OTHER TWO ENTRIES BLANK.
C
C     CARD FOUR.  RECORD(I).  FORMAT(2X,I3).
C     ONLY IF TAPENO .GE. 12.  USE BLANK CARD TO TERMINATE.
C
C     CARD FIVE.  TITLE.  FORMAT(8A10).  MAY NOT BE NEEDED.
C
C     CARD SIX.  DATA.  FORMAT(6E12.5)  MAY NOT BE NEEDED.
C
C     STOP  1  =  CROSS.  CAUSING DATA SET TO BE SKIPPED OR MODIFIED.
C                 UNEXPECTED EOF.
C     STOP  2  =  CROSS.  CAUSING DATA SET TO BE SKIPPED OR MODIFIED.
C                 IMPROPER EXIT.
C     STOP  3  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                 UNEXPECTED IOPTION VALUE ENCOUNTERED.
C     STOP  4  =  CROSS.  CHECKING ISKIP(IP) ORDER.
C                 ERROR IN ORDER OF ISKIP PARAMETERS.
C     STOP  5  =  CROSS.  SETTING UP RECORD OPTION.
C                 SEQUENCE ERROR.
C     STOP  6  =  CROSS.  CAUSING ADDITION OF CROSS-SECTION SET.
C                 IMPROPER EXIT.
C     STOP  7  =  CROSS.  SETTING UP RECORD OPTION.
C                 TO MANY VALUES.
C     STOP 10  =  CROSS.  COMPLETE CROSS-SECTION DATA SET PRINTOUT.
C                 UNEXPECTED EOF IN TITLE READ.
C     STOP 11  =  CROSS.  CAUSING PROPER CHANGE TO OCCUR.
C                 UNEXPECTED EOF IN TAPENO.
C     STOP 12  =  CROSS.  CAUSING DATA SET TO BE SKIPPED OR MODIFIED.
C                 IMPROPER IOPTION.
C     STOP 13  =  CROSS.  CAUSING DATA SET TO BE SKIPPED OR MODIFIED.
C                 UNEXPECTED TAPENO, TOO LARGE FOR PROGRAM CARD.
C     STOP 14  =  CROSS.  CASE OF TAPENO = 2 AND NO CHANGE DESIRED.
C                 UNEXPECTED EOF.
C     STOP 15  =  CROSS.  CAUSING DATA SET DELETION.
C                 UNEXPECTED EOF.
C     STOP 16  =  CROSS.  CASE OF TAPENO = 2 AND NO CHANGE DESIRED.
C                 UNEXPECTED EOF IN TITLE READ.
C     STOP 17  =  CROSS.  CASE OF TAPENO = 2 AND NO CHANGE DESIRED.
C                 UNEXPECTED EOF IN DATA READ.
C     STOP 20  =  CROSS.  CAUSING DATA SET DELETION.
C                 UNEXPECTED EOF IN TITLE READ.
C     STOP 21  =  CROSS.  CAUSING DATA SET DELETION.
C                 UNEXPECTED EOF IN TITLE READ.
C     STOP 22  =  CROSS.  IF ALL INCOMING DATA COMING FROM ACTUAL TAPE.
C                 UNEXPECTED EOF IN TITLE READ.
C     STOP 23  =  CROSS.  IF ALL INCOMING DATA COMING FROM ACTUAL TAPE.
C                 UNEXPECTED EOF IN TITLE READ.
C     STOP 24  =  CROSS.  CAUSING PROPER CHANGE TO OCCUR.
C                 UNEXPECTED EOF IN DATA READ.
```

```
C      STOP 25  =  CROSS.  CAUSING TITLE CHANGE.
C                  UNEXPECTED EOF IN TITLE READ.
C      STOP 26  =  CROSS.  CAUSING TITLE CHANGE.
C                  UNEXPECTED EOF IN DATA READ.
C      STOP 27  =  CROSS.  CAUSING DATA SET REPLACEMENT.
C                  UNEXPECTED EOF IN TITLE READ.
C      STOP 30  =  CROSS.  CAUSING DATA SET REPLACEMENT.
C                  UNEXPECTED EOF IN DATA READ.
C      STOP 31  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                  UNNECESSARY POSITIONING OF TAPE FILE.
C      STOP 32  =  CROSS.  CROSS CHECKING ISKIP AND RECORD OPTIONS.
C                  UNBALANCED INPUT.
C      STOP 33  =  CROSS.  CAUSING TITLE CHANGE.
C                  UNEXPECTED EOF IN ATAPE TITLE READ.
C      STOP 34  =  CROSS.  CAUSING PROPER CHANGE TO OCCUR.
C                  ERROR IN COMPUTED GO TO.
C      STOP 35  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                  UNEXPECTED EOF IN TITLE READ.
C      STOP 36  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                  UNEXPECTED EOF IN TITLE READ.
C      STOP 37  =  CROSS.  CAUSING PROPER CHANGE TO OCCUR.
C                  ERROR IN RANGE OF COMPUTED GO TO STATEMENT.
C      STOP 40  =  CROSS.  CASE OF TAPENO = 2 AND NO CHANGE DESIRED.
C                  UNEXPECTED EOF IN BINARY FLUX READ.
C      STOP 41  =  CROSS.  CASE OF TAPENO = 2 AND NO CHANGE DESIRED.
C                  UNEXPECTED EOF IN BINARY CURRENT READ.
C      STOP 42  =  CROSS.  MULTIPLE ENTRIES.
C                  CURRENT OR FLUX NUMBER LARGER THAN DIMENSION STATEMENT
C      STOP 43  =  CROSS.  MULTIPLE ENTRIES.
C                  NUMBER OF GROUPS LARGER THAN DIMENSION STATEMENT.
C      STOP 44  =  CROSS.  CAUSING DATA SET REPLACEMENT.
C                  UNEXPECTED EOF IN DATA(I) READ.
C      STOP 45  =  CROSS.  COMPLETE TRIAL SOLUTION DATA SET PRINTOUT.
C                  UNDEFINED PATH IN PARTIAL PRINTOUT.
C      STOP 46  =  CROSS.  CAUSING DATA SET REPLACEMENT.
C                  UNEXPECTED EOF IN FLUX READ.
C      STOP 47  =  CROSS.  CAUSING DATA SET REPLACEMENT.
C                  UNEXPECTED EOF IN CURRENT READ.
C      STOP 50  =  CROSS.  CAUSING ADDITION OF DATA SET.
C                  UNEXPECTED EOF IN DATA(I) READ.
C      STOP 51  =  CROSS.  CAUSING ADDITION OF DATA SET.
C                  UNEXPECTED EOF IN FLUX READ.
C      STOP 52  =  CROSS.  CAUSING ADDITION OF DATA SET.
C                  UNEXPECTED EOF IN CURRENT READ.
C      STOP 53  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                  UNEXPECTED EOF IN TITLE READ.
C      STOP 54  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                  INVALID POSITIONING OF TAPE FILE.
C      STOP 55  =  CROSS.  IF ALL INCOMING DATA FROM ACTUAL TAPE.
C                  INVALID TAPENO VALUE ENCOUNTERED.
C      STOP 55  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                  INVALID POSITIONING OF TAPE FILE.
C      STOP 56  =  CROSS.  SETTING POSITION OF TAPE FILE.
C                  TO MANY EOFS ENCOUNTERED BETWEEN DATA BLOCKS.
C      STOP 57  =  CROSS.  7600 TAPE CONVERSION.
C                  TO MANY EOFS ENCOUNTERED.
C      STOP 60  =  CROSS.  7600 TAPE CONVERSION.
C                  STRANGE PATH.
C      STOP 61  =  CROSS.  COMPLETE TRIAL SOLUTION DATA SET PRINTOUT.
C                  UNEXPECTED EOF IN FLUX READ.
C      STOP 62  =  CROSS.  COMPLETE TRIAL SOLUTION DATA SET PRINTOUT
C                  UNEXPECTED EOF IN DATA(I) READ.
C      STOP 63  =  CROSS.  COMPLETE TRIAL SOLUTION DATA SET PRINTOUT.
C                  UNEXPECTED EOF IN CURRENT READ.
C      STOP 777 =  NORMAL EXIT.
C
```

```
      DIMENSION TITLE(8)
      INTEGER OPTION
      INTEGER ATAPE,TAPENO
      READ(1,10) OPTION
      IF(OPTION .LE. 1) OPTION = 1
      IF(OPTION .GE. 2) OPTION = 2
      WRITE(4,15) OPTION
      READ(1,5) TITLE
      WRITE(4,5) TITLE
      CALL CROSS(OPTION)
      STOP 777
    5 FORMAT(8A10)
   10 FORMAT(I5)
   15 FORMAT(1H1,//* OPTION = *,I3,//)
      END
C     **********
      SUBROUTINE CROSS (OPTION)
C
C
C     READ ALL MESH VALUES BY GROUP.
      DIMENSION TITLE(1,8), C(1,15,29), ISKIP(50), IOPTION(50), RECORD(5
     10), DATA(50), LABEL(50), FLUX(29,119), CURRENT(29,120), BUF(3500)
      INTEGER TAPENO,ATAPE,RECORD,OPTION,DATASET
C     TAPE NUMBERS USED IN CROSS-SECTION DIRECTORY = 1,2,4,6,7,8,9,10,11
C                                                    12,30.
C     TAPE NUMBERS USED IN TRIAL SOLUTION DIRECTORY= 1,2,3,4,5,10,11,12,
C                                                    30,40.
C     IGM = THE NUMBER OF GROUPS PER DATA BLOCK.  FIXED AT 29.
C     IHM = THE NUMBER OF CROSS-SECTIONS PER DATA BLOCK.  FIXED AT 15.
C     IOPTION = 1 FOR DATA SET DELETION.
C             = 2 FOR TITLE CHANGE.
C             = 3 FOR DATA SET REPLACEMENT.
C             = 4 FOR ADDITION OF NEW CROSS-SECTION SET AT THE END OF TH
C                 CURRENT DATA SET.
C     IP      = INDEX FOR THE ISKIP(IP) OPTION.
C     ISKIP(I) = NUMBER OF CROSS-SECTION ENTRY FOR WHICH ACTION IS NECESSARY.
C     IZ      = NUMBER OF THE DATA SET CURRENTLY UNDER CONSIDERATION.
C     IZIP    = THE NUMBER OF ISKIP CARDS READ.
C     TAPENO = TAPE NUMBER FROM WHICH CROSS-SECTION DATA IS TO BE READ.
C             = 1 FOR INPUT FILE.  DEFAULT VALUE.
C             = 2 FOR PERMFILE STORE.
C             =  11 FOR ALL INCOMING DATA FROM ACTUAL TAPE.
C             =  ANYTHING GREATER THAN 11 FOR ACTUAL PHYSICAL TAPE FOR PATIAL
C                DATA ADDITION.
C     TAPE(3) = TRIAL SOLUTION DIRECTORY.
C     TAPE(5) = TRIAL SOLUTION DATA.
C     TAPE(7) = CROSS-SECTION DIRECTORY.
C     TAPE(6) = DATA SET FOR CROSS-SECTION DIRECTORY.
      ITAPE=4
      IGM=29 $ IHM=15
      IZIP=ILOG=IPRINT=ILUCK=IREC=IZI=0
C
C     CLEARING OUT OF ARRAYS.
C
      DO 10 I=1,50
      DATA(I)=0.0
      IOPTION(I)=0
      RECORD(I)=0
      ISKIP(I)=0
10    CONTINUE
      DO 20 I=1,100
      DO 20 J=1,29
      CURRENT(J,I)=0.00
20    CONTINUE
      IF (OPTION.EQ.1) GO TO 30
C
```

```
C     CREATING CROSS-SECTION DIRECTORY TITLE.
C
      WRITE (7,1560)
C
C     CAUSING DATA SET TO BE SKIPPED OR MODIFIED.
C     EXECUTED REGARDLESS OF SPECIFIED VALUE OF TAPENO.
C
30    CONTINUE
      DO 50 I=1,50
      READ (1,1570)ISKIP(I),IOPTION(I),TAPENO
      IF (EOF,1) 60,40
40    CONTINUE
      IF (ISKIP(I).LE.0.AND.IOPTION(I).LE.0) GO TO 70
      IF (IOPTION.LE.0.OR.IOPTION.GT.4) STOP 12
      IZIP=IZIP+1
50    CONTINUE
      STOP 2
60    CONTINUE
      STOP 1
70    CONTINUE
      IF (TAPENO.GT.0) GO TO 80
      IPRINT=1
      WRITE (4,1900)IPRINT
80    CONTINUE
      TAPENO=IABS(TAPENO)
      IZI=IZIP-1
      IF (TAPENO.LE.0) TAPENO=2
      IP=1
      IZ=0
      IF (TAPENO.GT.12) STOP 13
      IF (IZI.LE.0) IZI=0
      IF (IZIP.LT.1) GO TO 110
      IF (IZI.EQ.0) GO TO 100
C
C     CHECKING ISKIP(IP) ORDER.
C
      DO 90 II=1,IZI
      IV=II+1
      IF (ISKIP(IV).LE.ISKIP(II)) STOP 4
90    CONTINUE
C
C     ECHO-CHECKING ISKIP AND IOPTION PARAMETERS.
C
      WRITE (4,1580)
      DO 100 I=1,IZI
      WRITE (4,1590)ISKIP(I),IOPTION(I)
100   CONTINUE
      IF (IZI.EQ.0) WRITE (4,1580)
      WRITE (4,1600)ISKIP(IZIP),IOPTION(IZIP),TAPENO
C
C     SETTING UP RECORD OPTION.
C
110   CONTINUE
      IF (TAPENO.LE.11) GO TO 160
      DO 140 I=1,50
      READ (1,1610)RECORD(I)
      IF (EOF,1) 120,130
120   CONTINUE
      STOP 5
130   CONTINUE
      IF (RECORD(I).LE.0) GO TO 150
      IREC=IREC+1
140   CONTINUE
      STOP 7
150   CONTINUE
      WRITE (4,1620)
```

```
      WRITE (4,1630)(RECORD(I),I=1,IREC)
160   CONTINUE
      ATAPE=TAPENO
      IF (OPTION.EQ.1.AND.TAPENO.EQ.12) TAPENO=2
C
C     CROSS CHECKING ISKIP AND RECORD OPTIONS
C
      IF (ATAPE.EQ.12.AND.OPTION.EQ.1.AND.TAPENO.EQ.2) GO TO 170
      IF (ATAPE.LT.10) GO TO 190
170   CONTINUE
      IRECORD=IREC
      DO 180 IKLW=1,IZIP
      IF (IOPTION(IKLW).EQ.1) IRECORD=IRECORD+1
      IF (OPTION.EQ.1.AND.IOPTION(IKLW).EQ.2) IRECORD=IRECORD+1
180   CONTINUE
      IF (IZIP.EQ.IRECORD) GO TO 190
      WRITE (4,1550)IZIP,IRECORD,IREC
      STOP 32
190   CONTINUE
      IF (OPTION.NE.1.OR.(OPTION.EQ.1.AND.ATAPE.NE.12)) GO TO 230
C
C     7600 TAPE CONVERSION.
C
      REWIND 12
      ISTOP=0
      REWIND 40
      ITAPES=0
      DO 220 I=1,ITAPE
      CALL CROS76 (40,12,1,BUF,2)
      READ (40)
      IF (EOF,40) 200,210
200   CONTINUE
      ITAPES=ITAPES+1
      IF (ITAPES.GT.ITAPE) STOP 57
      GO TO 220
210   CONTINUE
      ISTOP=ISTOP+1
220   CONTINUE
      REWIND 12
      END FILE 40
      WRITE (4,1980)ISTOP
230   CONTINUE
C
C     CREATING DIRECTORY.
C
      DATASET=0
      DO 1260 I=1,100
      IST=IZAP=IRETURN=IWHY=IFLUX2=ICUR2=0
      IKK=ITAPE=IFLUX=ICURENT=0
      KK=IZ+1
      DATASET=DATASET+1
      IF (I.EQ.ISKIP(IP)) GO TO 290
      IF (TAPENO.EQ.2) GO TO 990
      IF (TAPENO.EQ.1) GO TO 990
      IF (TAPENO.EQ.12) GO TO 980
C
C     IF ALL INCOMING CROSS-SECTION DATA COMING FROM ACTUAL TAPE
C
      IF (TAPENO.NE.11) STOP 55
      IF (OPTION.EQ.1) GO TO 280
      READ (TAPENO,1640)(TITLE(1,J),J=1,8)
      IF (EOF,TAPENO) 240,250
240   CONTINUE
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      STOP 22
250   CONTINUE
```

```
      READ (TAPENO,1660)((C(1,K,J),K=1,IHM),J=1,IGM)
      IF (EOF,TAPENO) 260,270
260   CONTINUE
      WRITE (4,1700)I
      STOP 23
270   CONTINUE
      IZ=IZ+1
      WRITE (6)(TITLE(1,J),J=1,8)
      WRITE (6)((C(1,K,J),K=1,IHM),J=1,IGM)
      WRITE (7,1650)IZ,(TITLE(1,J),J=1,8)
      GO TO 1260
280   CONTINUE
      GO TO 1260
C
C     CAUSING PROPER CHANGE TO OCCUR.
C
290   CONTINUE
      IF (OPTION.EQ.1) GO TO 400
      TAPENO=1
      IF (ATAPE.EQ.12.AND.IOPTION(IP).GT.1) GO TO 490
300   CONTINUE
      IP=IP+1
      IF (IST.EQ.1) IP=IP-1
      IF (IOPTION(IP-1).EQ.1) GO TO 330
      READ (TAPENO,1640)(TITLE(1,J),J=1,8)
      IF (EOF,TAPENO) 310,320
310   CONTINUE
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      STOP 11
320   CONTINUE
330   CONTINUE
      KLW=IOPTION(IP-1)
      IF (KLW.LT.1.OR.KLW.GT.4) GO TO 390
      GO TO (370,370,340,340), KLW
340   CONTINUE
      IF (OPTION.EQ.1) GO TO 360
      READ (TAPENO,1660)((C(1,K,J),K=1,IHM),J=1,IGM)
      IF (EOF,TAPENO) 350,380
350   CONTINUE
      WRITE (4,1700)I
      STOP 24
360   CONTINUE
370   CONTINUE
380   CONTINUE
      GO TO (410,680,780,1120), KLW
390   CONTINUE
      WRITE (4,1880)KLW,IOPTION(IP),IOPTION(IP-1),ISKIP(IP),ISKIP(IP)
      STOP 34
400   CONTINUE
      IP=IP+1
      KLW=IOPTION(IP-1)
      IF (KLW.GT.4.OR.KLW.LE.0) STOP 37
      GO TO (410,680,780,1120), KLW
C
C     CAUSING DATA SET DELETION.
C
410   CONTINUE
      TAPENO=ATAPE
      IF (TAPENO.EQ.12) TAPENO=2
      IF (OPTION.EQ.1) GO TO 460
      READ (TAPENO)(TITLE(1,J),J=1,8)
      IF (EOF,TAPENO) 420,430
420   CONTINUE
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      STOP 20
430   CONTINUE
```

```
      WRITE (4,1870)
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      READ (TAPENO)((C(1,K,J),K=1,IHM),J=1,IGM)
      IF (EOF,TAPENO) 440,450
440   CONTINUE
      STOP 21
450   CONTINUE
      WRITE (8)(TITLE(1,J),J=1,8)
      WRITE (8)((C(1,K,J),K=1,IHM),J=1,IGM)
      TAPENO=ATAPE
      GO TO 1260
460   CONTINUE
      READ (2)(DATA(I),I=1,12),I1,I2,I3,I4
      IF (EOF,2) 470,480
470   CONTINUE
      STOP 15
480   CONTINUE
      DATASET=DATASET-1
      READ (2)
      READ (2)
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      GO TO 1260
C
C     SETTING POSITION OF TAPE FILE.
C
490   CONTINUE
      WRITE (4,1670)
      TAPENO=ATAPE
      ILUCK=ILUCK+1
      IF (ILUCK.GT.1) REWIND TAPENO
      ILOG=ILOG+1
      IF (ILOG.GT.IREC) GO TO 670
      ICOUNT=RECORD(ILOG)-1
      IF (OPTION.EQ.1) GO TO 560
      IF (ICOUNT.EQ.0) GO TO 550
      DO 540 IZZ=1,ICOUNT
      READ (TAPENO,1640)(TITLE(1,K),K=1,8)
      IF (EOF,TAPENO) 500,510
500   CONTINUE
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      STOP 35
510   CONTINUE
      READ (TAPENO,1660)((C(1,K,L),K=1,IHM),L=1,IGM)
      IF (EOF,TAPENO) 520,530
520   CONTINUE
      WRITE (4,1700)I
      STOP 36
530   CONTINUE
      WRITE (4,1640)(TITLE(1,K),K=1,8)
540   CONTINUE
550   CONTINUE
      IP=IP+1
      IST=1
      IF (IOPTION(IP-1).EQ.1) GO TO 660
      IF (IOPTION(IP-1).EQ.2) GO TO 690
      IF (IOPTION(IP-1).EQ.3) GO TO 300
      IF (IOPTION(IP-1).EQ.4) GO TO 300
      WRITE (4,1860)IOPTION(IP-1)
      STOP 3
560   CONTINUE
      IF (ICOUNT.EQ.0) GO TO 630
      DO 620 IZZ=1,ICOUNT
      READ (12)(DATA(I),I=1,12),I1,I2,I3,I4
      IF (EOF,12) 570,580
570   CONTINUE
```

```
      STOP 53
580   CONTINUE
      I5=I3+1
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      WRITE (4,1720)(DATA(I),I=1,12)
      READ (12)
      READ (12)
      READ (12)
      IF (EOF,12) 590,590
590   CONTINUE
      READ (12)
      IF (EOF,12) 610,600
600   CONTINUE
      WRITE (4,1970)
610   CONTINUE
620   CONTINUE
630   CONTINUE
      NUT=IOPTION(IP-1)
      GO TO (640,650,850,1150), NUT
640   CONTINUE
      STOP 54
650   CONTINUE
      STOP 55
660   CONTINUE
      STOP 31
670   CONTINUE
      WRITE (4,1990)ILOG,IREC
      GO TO 1270
C
C     CAUSING TITLE CHANGE.
C
680   CONTINUE
      IF (OPTION.EQ.1) GO TO 770
      IF (ATAPE.EQ.12) GO TO 490
690   CONTINUE
      IF (ATAPE.NE.12) GO TO 710
      READ (ATAPE)(TITLE(1,J),J=1,8)
      IF (EOF,ATAPE) 700,710
700   CONTINUE
      STOP 33
710   CONTINUE
      WRITE (6)(TITLE(1,J),J=1,8)
      WRITE (7,1650)KK,(TITLE(1,J),J=1,8)
      TAPENO=ATAPE
      IF (ATAPE.EQ.12) TAPENO=2
      READ (TAPENO)(TITLE(1,J),J=1,8)
      IF (EOF,TAPENO) 720,730
720   CONTINUE
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      STOP 25
730   CONTINUE
      WRITE (8)(TITLE(1,J),J=1,8)
      READ (TAPENO)((C(1,K,J),K=1,IHM),J=1,IGM)
      IF (EOF,TAPENO) 740,750
740   CONTINUE
      WRITE (4,1700)I
      STOP 26
750   CONTINUE
      WRITE (6)((C(1,K,J),K=1,IHM),J=1,IGM)
      IZ=IZ+1
      TAPENO=ATAPE
      WRITE (9,1890)
      IRETURN=1
      IZAP=1
      GO TO 1290
```

```
760   CONTINUE
      IRETURN=0
      IZAP=0
      GO TO 1260
770   CONTINUE
      READ (1,1720)(LABEL(I),I=1,12)
      WRITE (4,1680)I
      WRITE (3,1710)I
      WRITE (3,1720)(LABEL(I),I=1,12)
      READ (2)(DATA(I),I=1,12),I1,I2,I3,I4
      I5=I3+1
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      WRITE (3,1730)I1,I2,I3,I4
      WRITE (5)(LABEL(I),I=1,12),I1,I2,I3,I4
      READ (2)((FLUX(K,J),J=1,I3),K=1,I4)
      WRITE (5)((FLUX(K,J),J=1,I3),K=1,I4)
      READ (2)((CURRENT(K,J),J=1,I5),K=1,I4)
      WRITE (5)((CURRENT(K,J),J=1,I5),K=1,I4)
      GO TO 1260
C
C     CAUSING DATA SET REPLACEMENT.
C
780   CONTINUE
      IF (OPTION.EQ.1) GO TO 840
      WRITE (6)(TITLE(1,J),J=1,8)
      WRITE (6)((C(1,K,J),K=1,IHM),J=1,IGM)
      WRITE (7,1650)KK,(TITLE(1,J),J=1,8)
      WRITE (9,1690)
      IZAP=1
      GO TO 1290
790   CONTINUE
      IZAP=0
      IZ=IZ+1
      TAPENO=ATAPE
      IF (ATAPE.EQ.12) TAPENO=2
      READ (TAPENO)(TITLE(1,J),J=1,8)
      IF (EOF,TAPENO) 800,810
800   CONTINUE
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      STOP 27
810   CONTINUE
      WRITE (8)(TITLE(1,J),J=1,8)
      READ (TAPENO)((C(1,K,J),K=1,IHM),J=1,IGM)
      IF (EOF,TAPENO) 820,830
820   CONTINUE
      WRITE (4,1700)I
      STOP 30
830   CONTINUE
      WRITE (8)((C(1,K,J),K=1,IHM),J=1,IGM)
      TAPENO=ATAPE
      GO TO 1260
840   CONTINUE
      GO TO 490
850   CONTINUE
      ITAPE=4
      READ (2)(DATA(I),I=1,12),I1,I2,I3,I4
      IF (EOF,2) 860,870
860   CONTINUE
      STOP 44
870   CONTINUE
      I5=I3+1
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      IF (IKK.EQ.0) WRITE (4,1920)
      WRITE (ITAPE,1710)I
```

```
      WRITE (ITAPE,1720)(DATA(I),I=1,12)
      WRITE (ITAPE,1730)I1,I2,I3,I4
      IF (ITAPE.EQ.3) GO TO 910
      IF (IKK.EQ.1) GO TO 900
      READ (12)(DATA(I),I=1,12),I1,I2,I3,I4
      IF (EOF,12) 890,880
880   CONTINUE
890   CONTINUE
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      IKK=1
      GO TO 870
900   CONTINUE
      ITAPE=3
      GO TO 870
910   CONTINUE
      WRITE (5)(DATA(I),I=1,12),I1,I2,I3,I4
      WRITE (9,1710)I
      WRITE (9,1720)(DATA(I),I=1,12)
      WRITE (9,1730)I1,I2,I3,I4
      IKK=0
      READ (2)
      READ (2)
      READ (12)((FLUX(K,J),J=1,I3),K=1,I4)
      IF (EOF,12) 920,930
920   CONTINUE
      STOP 46
930   CONTINUE
      I5=I3+1
      WRITE (5)((FLUX(K,J),J=1,I3),K=1,I4)
      IFLUX=1
      GO TO 1430
940   CONTINUE
      IFLUX=0
      READ (12)((CURRENT(K,J),J=1,I5),K=1,I4)
      IF (EOF,12) 950,960
950   CONTINUE
      STOP 47
960   CONTINUE
      WRITE (5)((CURRENT(K,J),J=1,I5),K=1,I4)
      ICURENT=1
      GO TO 1430
970   CONTINUE
      ICURENT=0
      GO TO 1260
C
C     CASE OF TAPENO = 2 AND NO CHANGE DESIRED.
C
980   CONTINUE
      IF (OPTION.EQ.1) GO TO 1040
      TAPENO=2
990   CONTINUE
      IF (OPTION.EQ.1) GO TO 1040
      IF (I.GT.ISKIP(IZIP)) GO TO 1110
      READ (TAPENO)(TITLE(1,J),J=1,8)
      IF (EOF,TAPENO) 1000,1010
1000  CONTINUE
      WRITE (4,1640)(TITLE(1,J),J=1,8)
      STOP 16
1010  CONTINUE
      READ (TAPENO)((C(1,K,J),K=1,IHM),J=1,IGM)
      IF (EOF,TAPENO) 1020,1030
1020  CONTINUE
      WRITE (4,1700)I
      STOP 17
1030  CONTINUE
```

```
      IZ=IZ+1
      WRITE (6)(TITLE(1,J),J=1,8)
      WRITE (6)((C(1,K,J),K=1,IHM),J=1,IGM)
      WRITE (7,1650)IZ,(TITLE(1,J),J=1,8)
      IF (ATAPE.EQ.12) TAPENO=ATAPE
      GO TO 1260
1040  CONTINUE
      READ (2)(DATA(I),I=1,12),I1,I2,I3,I4
      IF (EOF,2) 1050,1060
1050  CONTINUE
      WRITE (4,2010)
      DATASET=DATASET-1
      GO TO 1270
1060  CONTINUE
      WRITE (3,1710)I
      WRITE (3,1720)(DATA(I),I=1,12)
      WRITE (3,1730)I1,I2,I3,I4
      I5=I3+1
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      WRITE (5)(DATA(I),I=1,12),I1,I2,I3,I4
      READ (2)((FLUX(K,J),J=1,I3),K=1,I4)
      IF (EOF,2) 1070,1080
1070  CONTINUE
      STOP 40
1080  CONTINUE
      WRITE (5)((FLUX(K,J),J=1,I3),K=1,I4)
      READ (2)((CURRENT(K,J),J=1,I5),K=1,I4)
      IF (EOF,2) 1090,1100
1090  CONTINUE
      STOP 41
1100  CONTINUE
      WRITE (5)((CURRENT(K,J),J=1,I5),K=1,I4)
      GO TO 1260
1110  CONTINUE
      WRITE (4,2000)I,ISKIP(IZIP)
      GO TO 1270
C
C     CAUSING ADDITION OF DATA SET AT THE END OF THE OLD DIRECTORY.
C
1120  CONTINUE
      IF (OPTION.EQ.1) GO TO 1140
      WRITE (6)(TITLE(1,J),J=1,8)
      WRITE (6)((C(1,K,J),K=1,IHM),J=1,IGM)
      WRITE (7,1650)KK,(TITLE(1,J),J=1,8)
      WRITE (9,1740)
      IZAP=1
      IRETURN=1
      GO TO 1290
1130  CONTINUE
      IRETURN=0
      IZAP=0
      IZ=IZ+1
      TAPENO=ATAPE
      GO TO 1260
1140  CONTINUE
      GO TO 490
1150  CONTINUE
      READ (12)(DATA(I),1=1,12),I1,I2,I3,I4
      IF (EOF,12) 1160,1170
1160  CONTINUE
      STOP 50
1170  CONTINUE
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      ITAPE=4
```

```
      WRITE (4,1930)
1180  CONTINUE
      WRITE (ITAPE,1710)I
      WRITE (ITAPE,1720)(DATA(I),I=1,12)
      WRITE (ITAPE,1730)I1,I2,I3,I4
      IF (ITAPE.EQ.3) GO TO 1190
      ITAPE=3
      GO TO 1180
1190  CONTINUE
      WRITE (5)(DATA(I),I=1,12),I1,I2,I3,I4
      WRITE (9,1710)I
      WRITE (9,1720)(DATA(I),I=1,12)
      WRITE (9,1730)I1,I2,I3,I4
      READ (12)((FLUX(K,J),J=1,I3),K=1,I4)
      IF (EOF,12) 1200,1210
1200  CONTINUE
      STOP 51
1210  CONTINUE
      WRITE (5)((FLUX(K,J),J=1,I3),K=1,I4)
      IFLUX2=1
      GO TO 1430
1220  CONTINUE
      IFLUX2=0
      I5=I3+1
      READ (12)((CURRENT(K,J),J=1,I5),K=1,I4)
      IF (EOF,12) 1230,1240
1230  CONTINUE
      STOP 52
1240  CONTINUE
      WRITE (5)((CURRENT(K,J),J=1,I5),K=1,I4)
      ICUR2=1
      GO TO 1430
1250  CONTINUE
      ICUR2=0
      GO TO 1260
1260  CONTINUE
      STOP 6
1270  CONTINUE
      IF (OPTION.EQ.2) GO TO 1280
      WRITE (4,1910)
      GO TO 1390
1280  CONTINUE
      WRITE (4,1750)
C
C     COMPLETE CROSS-SECTION DATA SET PRINTOUT.
C
      IF (IPRINT.NE.1) GO TO 1380
      WRITE (9,1760)
      REWIND 6
      WRITE (9,1770)
1290  CONTINUE
      DO 1370 ITIMES=1,KK
      IF (IZAP.EQ.1) GO TO 1310
      READ (6)(TITLE(1,J),J=1,8)
      IF (EOF,6) 1300,1310
1300  CONTINUE
      STOP 10
1310  CONTINUE
      WRITE (9,1780)(TITLE(1,J),J=1,8)
      WRITE (9,1790)
      IF (IZAP.EQ.1) GO TO 1330
      READ (6)((C(1,K,J),K=1,IHM),J=1,29)
      IF (EOF,6) 1320,1330
1320  CONTINUE
      WRITE (9,1800)KK
1330  CONTINUE
```

```
      DO 1340 IQ=1,15
      WRITE (9,1810)IQ,(C(1,IQ,J),J=1,10)
1340  CONTINUE
      WRITE (9,1820)
      DO 1350 IQ=1,15
      WRITE (9,1810)IQ,(C(1,IQ,J),J=11,20)
1350  CONTINUE
      WRITE (9,1830)
      DO 1360 IQ=1,15
      WRITE (9,1840)IQ,(C(1,IQ,J),J=21,29)
1360  CONTINUE
      WRITE (4,1850)
      IF (IZAP.EQ.1.AND.IRETURN.EQ.1) GO TO 760
      IF (IRETURN.EQ.1) GO TO 1130
      IF (IZAP.EQ.1) GO TO 790
1370  CONTINUE
1380  CONTINUE
      WRITE (9,1850)
1390  CONTINUE
      IWHY=1
C
C     COMPLETE TRIAL SOLUTION DATA SET PRINTOUT.
C
      IF (OPTION.EQ.2) GO TO 1540
      IF (IPRINT.NE.1) GO TO 1540
      WRITE (9,1940)
      REWIND 5
      DO 1530 IX=1,DATASET
      IF (IWHY.EQ.0) GO TO 1410
      READ (5)(DATA(I),I=1,12),I1,I2,I3,I4
      IF (EOF,5) 1400,1410
1400  CONTINUE
      STOP 62
1410  CONTINUE
      I5=I3+1
      IF (I5.GT.120) STOP 42
      IF (I4.GT.29) STOP 43
      WRITE (9,1710)IX
      WRITE (9,1720)(DATA(I),I=1,12)
      WRITE (9,1730)I1,I2,I3,I4
      IF (IWHY.EQ.0) GO TO 1440
      READ (5)((FLUX(K,J),J=1,I3),K=1,I4)
      IF (EOF,5) 1420,1440
1420  CONTINUE
      STOP 61
1430  CONTINUE
      IF (IFLUX.EQ.1) GO TO 1440
      IF (ICURENT.EQ.1) GO TO 1490
      IF (IFLUX2.EQ.1) GO TO 1440
      IF (ICUR2.EQ.1) GO TO 1490
      STOP 45
1440  CONTINUE
      WRITE (9,1950)
      WRITE (9,1790)
      DO 1450 IZK=1,I3
      WRITE (9,1810)IZK,(FLUX(J,IZK),J=1,10)
1450  CONTINUE
      WRITE (9,1820)
      DO 1460 IZK=1,I3
      WRITE (9,1810)IZK,(FLUX(J,IZK),J=11,20)
1460  CONTINUE
      WRITE (9,1830)
      DO 1470 IZK=1,I3
      WRITE (9,1840)IZK,(FLUX(J,IZK),J=21,29)
1470  CONTINUE
      IF (IFLUX.EQ.1) GO TO 940
```

```
      IF (IFLUX2.EQ.1) GO TO 1220
      IF (IWHY.EQ.0) GO TO 1490
      READ (5)((CURRENT(K,J),J=1,I5),K=1,I4)
      IF (EOF,5) 1480,1490
1480  CONTINUE
      STOP 63
1490  CONTINUE
      WRITE (9,1960)
      WRITE (9,1790)
      DO 1500 IZK=1,I5
      WRITE (9,1810)IZK,(CURRENT(J,IZK),J=1,10)
1500  CONTINUE
      WRITE (9,1820)
      DO 1510 IZK=1,I5
      WRITE (9,1810)IZK,(CURRENT(J,IZK),J=11,20)
1510  CONTINUE
      WRITE (9,1830)
      DO 1520 IZK=1,I5
      WRITE (9,1840)IZK,(CURRENT(J,IZK),J=21,29)
1520  CONTINUE
      IF (ICURENT.EQ.1) GO TO 970
      IF (ICUR2.EQ.1) GO TO 1250
1530  CONTINUE
1540  CONTINUE
      RETURN
C
C
C
1550  FORMAT (//* IZIP = *,I3,* IRECORD = *,I3,* AND IREC = *,I3,//)
1560  FORMAT (//* TREAT CROSS-SECTION SET WITH FIVE THERMAL GROUPS.*,//,
     1* IGM = 29, IHM = 15, IHT = 4, IHS = 10.*,//,* ENERGY STRUCTURE*,/
     2/* E07, 3E06, 1.4E06, 9E05, 4E05, 1E05, 1.7E04, 3.355E03, 2.035E03
     3,*,//,* 1234., 748.5, 454., 275.4, 167., 101.3, 61.44, 37.27, 22.6
     4,*,//,* 13.71, 8.315, 5.043, 3.059, 1.855, 1.125, 0.77487, 0.2814,
     5 *,//,* 0.1226, 0.04617, 0.01834*//)
1570  FORMAT (1X,I4,I5,2X,I3)
1580  FORMAT (//* ISKIP AND IOPTION PARAMETER PRINTOUT.*//)
1590  FORMAT (5X,I3,5X,I3)
1600  FORMAT (3(5X,I3))
1610  FORMAT (2X,I3)
1620  FORMAT (//* RECORD PARAMETER PRINTOUT.*//)
1630  FORMAT (5X,20(I3,2X))
1640  FORMAT (8A10)
1650  FORMAT (25X,I3,8A10)
1660  FORMAT (6E12.5)
1670  FORMAT (//* SETTING POSITION OF TAPE FILE.*//)
1680  FORMAT (//* TRIAL SOLUTION NUMBER *,I3,* HAS A NEW TITLE.*//)
1690  FORMAT (//* NEW DATA SET AFTER REPLACEMENT.*//)
1700  FORMAT (//* I = *,I3,//)
1710  FORMAT (//* TRIAL SOLUTION NUMBER         =  *,I6,//)
1720  FORMAT (12A6)
1730  FORMAT (//* THE X COORDINATE              = *,I4,//* THE Y COORDINA
     1TE              = *,I4,//* THE NUMBER OF MESH INTERVALS = *,I4,//*
     2THE NUMBER OF GROUPS          = *,I4,//)
1740  FORMAT (//* NEW DATA SET ADDITION.*//)
1750  FORMAT (//* ORIGINAL CROSS-SECTION DIRECTORY PRINTED OUT BEFORE NE
     1WLY ALTERED VERSION.*//)
1760  FORMAT (1H0,5X,* COMPLETE CROSS-SECTION DATA SET PRINTOUT.*,1H0)
1770  FORMAT (1H1)
1780  FORMAT (1H0,5X,8A10,1H0)
1790  FORMAT (//*  I      J=1          J=2          J=3          J=4
     1      J=5          J=6          J=7          J=8          J=9
     2     J=10     *//)
1800  FORMAT (//,* AT STOP 15 KK = *,I3,//)
1810  FORMAT (1X,I2,1X,10(1PE13.5))
1820  FORMAT (//*  I      J=11         J=12         J=13         J=14
```

```
     1       J=15          J=16          J=17          J=18          J=19
     2     J=20     *//)
1830  FORMAT (//*  I       J=21          J=22          J=23          J=24
     1       J=25          J=26          J=27          J=28          J=29 *//
     2)
1840  FORMAT (1X,I2,1X,9(1PE13.5))
1850  FORMAT (/////)
1860  FORMAT (//* IOPTION = *,I3,//)
1870  FORMAT (///* TITLE OF DELETED DATA SET *,//)
1880  FORMAT (//* KLW = *,I3,//,* IOPTION(IP) = *,I3,//,* IOPTION(IP-1)
     1= *,I3,//,* ISKIP(IP) = *,I3,//,* ISKIP(IP-1) = *,I3,//)
1890  FORMAT (//* TITLE CHANGE.  DATA SET WITH OLD TITLE. *//)
1900  FORMAT (//* BECAUSE OF NEGATIVE TAPENO IPRINT SET EQUAL TO*,I3,//)
1910  FORMAT (//* OLD TRIAL SOLUTION DIRECTORY PRINTED OUT BEFORE ALTERE
     1D VERSION.*,//)
1920  FORMAT (//* OLD DIRECTORY ENTRY AND COMPLETE NEW REPLACEMENT.*//)
1930  FORMAT (//* NEW DATA SET.*,//)
1940  FORMAT (1H0,5X,* COMPLETE TRIAL SOLUTION DATA SET PRINTOUT.*,1H0)
1950  FORMAT (//* FLUX PRINTOUT.*//)
1960  FORMAT (//* CURRENT PRINTOUT.*//)
1970  FORMAT (//* THE EQUIVALENT TO A STOP 56 HAS BEEN ENCOUNTERED.*//)
1980  FORMAT (//* IN 7600 TAPE CONVERSION THE EQUIVALENT TO A STOP 60 WA
     1S ENCOUNTERED*,I3,* TIMES.*//)
1990  FORMAT (//* EXIT FROM MAIN LOOP WITH ILOG =*,I4,* AND IREC =*,I4,/
     1/)
2000  FORMAT (//* EXIT FROM MAIN LOOP WITH I =*,I4,* AND ISKIP(IZIP) =*,
     1I4,//)
2010  FORMAT (//* EXIT FROM MAIN LOOP BECAUSE EOF FOUND IN TRIAL SOLUTIO
     1N TITLE READ.*//)
```

☆ U.S. GOVERNMENT PRINTING OFFICE: 1973—784-275/2