(1)

109.9

JUL 07 1982

WMHL: 3104.2

MEMORANDUM FOR: Malcolm R. Knapp
High-Level Waste Licensing
Management Branch
Division of Waste Management

FROM: Stewart A. Silling
High-Level Waste Licensing
Management Branch
Division of Waste Management

SUBJECT: RATRAP DOCUMENTATION

The attached document is the User's Manual for RATRAP, my post-processor for non-statistical graphics for use with NWFT/DVM.

ORIGINAL SIGNED BY

Stewart A. Silling
High-Level Waste Licensing
Management Branch
Division of Waste Management

Enclosure:
As Stated

Distribution: w/o encl.
WMHL file
WMHL r/f
WM r/f
NMSS r/f
JBMartin
REBrowning
MJBell
SASilling & r/f
HJMiller
JOBunting
PDR
MJWise w/encl.

ORIG NOT rtnd. fm DCS
ACC. 8208030117

Susp 7-13

OFFICE	WMHL						
SURNAME	SASilling:lmc						
DATE	7/7/82						

RATRAP User's Manual

Stewart A. Silling

July 7, 1982

TABLE OF CONTENTS

	Page
1. Introduction	1
2. Plot tape and plot requests	1
3. Producing plots	3
3.1 Generating the plot tape	4
3.2 Running RATRAP	6
3.2.1 Making a survey from a stored plot tape	6
3.2.2 Writing a curve specification	6
3.2.3 Input format	7

Appendices

A. Changes to NWFT/DVM input	18
B. Current limits	19
C. Example	20
D. Modifying a transport model to produce a plot tape	33
E. Updates to NWFT/DVM	35
F. RATRAP code listing	56

3104.2/SAS/82/06/25/0

RATRAP User's Manual

1. Introduction

RATRAP (Radionuclide Transport Plotter) is a code for producing plots of radionuclide discharges and other variables predicted by NRC high-level waste transport models. The code was developed within NRC to act as a post-processor for the transport models developed at Sandia.

Current implementation is on NWFT/DVM, although SWIFT could be easily modified to produce RATRAP plots. In the current implementation on NWFT/DVM, RATRAP produces plots of discharge rate, integrated discharge, EPA ratio, and health effects as a function of time, and also CCDF plots and scatter plots. Source rate can be plotted using the DVM option.

RATRAP interprets files called "plot tapes" produced by a specially modified transport model. These plot tapes contain the numerical data generated by the codes. RATRAP produces plots of this data according to user requests.

The RATRAP code includes the following features:

(1) Self-identifying plot tape data.
(2) Plot requests specified after model run is complete -- no need to rerun.
(3) Overlay plots.
(4) Self-labeling of plots.
(5) Self-scaling consistent between plots.

The code uses the DISSPLA device-independent graphics subroutines to produce plots. Plots can be produced on Calcomp drum plotters or Tektronix CRT terminals.

2. Plot tape and plot requests

The numerical data on the plot tape is in the form shown in Figure 1. Every set of coordinates defining a data point is preceded by a "name tag" which identifies the data point. Since every data point has its own unambiguous name tag, different types of data on the file can be mixed. For example, data on the source rate of Pu242 can be arbitrarily mixed with data on the discharge rate of Tc99.

```
                                                                  name tags
0    1    1    1    1    1    5    1
  9.5250000000E+03     2.5566002170E+03
0    1    1    1    1    1    4    1
  9.7625000000E+03     3.6122291984E+00
0    1    1    1    1    1    5    1
  9.7625000000E+03     2.6437668087E+03
0    1    1    1    1    1    4    1
  1.0000000000E+04     3.7283502802E+00
0    1    1    1    1    1    5    1
  1.0000000000E+04     2.7315996673E+03
```

numerical data

Figure 1. Excerpt from a plot file.

The name tag preceding each data point consists of a sequence of numbers called attributes. Each attribute has a specific meaning defined by the transport model. For example, the third attribute might identify the radionuclide to which the data point refers, while the fourth might identify the model which generated the data. The way that the meanings of attributes are defined is discussed in Appendix D.

The actual numbers contained in each of the attributes are called values. Each value defines which of several predefined possibilities for the attribute applies to the data point. For example, a value of 003 for the third attribute might mean Pu242, while a value of 004 might mean Tc99.

The meaning of each value of each attribute is defined in special title records which are similar to numerical data records. These title records, like the data records, are automatically written on the plot tape by the model which creates the file (see Appendix D).

User curve specification requests (see below) are simply the name tags of the data to be plotted. When RATRAP reads such a request, it searches the plot tape for all data points whose name tags are identical to the one specified by the user.

3. Producing plots

Frequently a modeler will perform a lengthy run of a deterministic code but not know beforehand what variables he wishes to have plotted. Post-processing of plots permits numerical data from the model to be temporarily stored while the modeler looks over the printed results and decides what plots he wants. When he decides, the modeler uses the post-processor to produce the desired plots from the data on the temporary file.

Use of RATRAP therefore involves a three-step procedure:

(1) Run the numerical model, produce and store the plot tape.
(2) Run RATRAP to generate the plots.
(3) Obtain the plots on a graphics device.

Each of these steps is explained below.

3.1 Generating the Plot Tape

Numerical data from the model are stored on the plot tape. The plot tape communicates the data to RATRAP when the user is ready to generate plots.

Special logic is inserted into the Sandia models in order to produce the plot tape. This logic is already included in a version of NWFT/DVM. The logic can also be put in any other code through the use of UPDATE. For users wishing to use RATRAP on other codes, instructions for these modifications are included in Appendix D.

There are some minor input changes that are used with NWFT/DVM to produce a plot tape. These input changes are discussed in Appendix A. When the NWFT/DVM run is complete, the plot tape must be saved. This is normally accomplished through a CATALOG command:

```
REQUEST,TAPE8,*PF.
ABS.
EXIT,U.
CATALOG,TAPE8,BOZOTAPE8,ID=ZZRNRC.
```

In the above control cards, it is assumed that the object code for the numerical model has been attached and is named ABS. Prior to execution of the code, a file named TAPE8 must be requested on a permfile device. This file will contain the plot tape. Execution of NWFT/DVM is initiated with the ABS command. The plot tape is written at this stage. The EXIT command ensures that the plot tape will be saved even if the run aborts. The EXIT command can be omitted, but then the plot tape will be saved only if NWFT/DVM is completed successfully. The CATALOG command stores the plot tape. At BNL, the user should place his own name where "BOZO" occurs above.

In order to find out what is on the plot tape, it is useful to have RATRAP perform a survey of the file immediately following completion of NWFT/DVM. This is accomplished by inserting the following control cards after the CATALOG command:

```
ATTACH,RATRAP,SILLINGRATRAP,ID=ZZRNRC,MR=1.
SKIPF,RATRAP,2,17.
COPYP,RATRAP,ABSR.
ABSR,SURVEY.
```

The above sequence of cards produces a survey of the plot tape. The survey provides information on the contents of the file. (See Figure 2.)

```
SURVEY COMPLETE       886 RECORDS READ


ATTRIBUTE TYPE   2 HAS NAME CODE
    VALUE   1  HAS TITLE NWFT/DVM$

ATTRIBUTE TYPE   3 HAS NAME DATE
    VALUE   1  HAS TITLE  06/13/82 $

ATTRIBUTE TYPE   4 HAS NAME PROBLEM
    VALUE   1  HAS TITLE CHU-PEPPING POINT VAL PROBLEM$

ATTRIBUTE TYPE   5 HAS NAME VECTOR
    VALUE   1  HAS TITLE VECTOR  1$
    VALUE   2  HAS TITLE VECTORS WITHIN 10CFR60$
    VALUE   3  HAS TITLE VECTORS WITHIN EPA LIM$
    VALUE   4  HAS TITLE VECTORS NOT WITHIN EPA LIM$

ATTRIBUTE TYPE   6 HAS NAME COORD1
    VALUE   1  HAS TITLE TIME (YR)$
    VALUE   2  HAS TITLE EPA RELEASE RATIO$
    VALUE   3  HAS TITLE EPA HEALTH EFFECTS$
    VALUE   4  HAS TITLE LEACH TIME (YR)$

ATTRIBUTE TYPE   7 HAS NAME COORD2
    VALUE   1  HAS TITLE INTEGRATED DISCHARGE (CI)$
    VALUE   2  HAS TITLE DISCHARGE RATE (CI/DAY)$
    VALUE   3  HAS TITLE SOURCE RATE (CI/DAY)$
    VALUE   4  HAS TITLE EPA RELEASE RATIO$
    VALUE   5  HAS TITLE EPA HEALTH EFFECTS$
    VALUE   6  HAS TITLE PROBABILITY$
    VALUE   7  HAS TITLE TRAVEL TIME (YR)$
    VALUE   8  HAS TITLE CANISTER LIFE (YR)$

ATTRIBUTE TYPE   8 HAS NAME ISOTOPE
    VALUE   1  HAS TITLE  C 14 $
    VALUE   2  HAS TITLE TC 99 $
    VALUE   3  HAS TITLE SN126 $
    VALUE   4  HAS TITLE  U238 $
    VALUE   5  HAS TITLE ALL NUCLIDES$
```

Figure 2. Sample survey from a plot tape produced by NWFT/DVM.

The use of this information in constructing plot requests is discussed in the following section.

3.2 Running RATRAP

3.2.1 Making a survey from a stored plot tape

The specification of plot requests requires a survey of the plot tape. A survey can be produced immediately after execution of the model, as shown above. If this survey is not available, another is easily produced:

```
ATTACH,TAPE8,BOZOTAPE8,ID=ZZRNRC,MR=1.
ATTACH,RATRAP,SILLINGRATRAP,MR=1.
SKIPF,RATRAP,2,17.
COPYP,RATRAP,ABSR.
ABSR,SURVEY.
```

The above control cards produce a survey identical to the one made using the control cards discussed in the previous section. Figure 2 shows an example of a survey.

3.2.2 Writing a curve specification

The most complicated user input required is the curve specification request. The exact format is given in the next section, but its meaning is discussed here.

The curve specification request is a series of 3-digit numbers containing the attribute values of the data desired. The value of the first attribute is always 000.

The values of the remaining attributes are read off the survey. As an aid to the user, the survey also contains a list of all the name tags on the plot tape. Consider the plot tape whose survey is shown in Figure 2. Suppose the user wants a plot of the discharge rate of C14. The attributes would be selected as follows:

Attribute	Value	Reason
1	000	always 000
2	001	only choice
3	001	only choice

4	001	only choice
5	001	vector 1
6	001	time on x-axis
7	002	discharge rate on y-axis
8	001	C14

The curve specification request would therefore be composed as follows:

000 001 001 001 001 001 002 001

The curve resulting from this curve specification is shown in Figure 3.

3.2.3 Input format

A RATRAP input deck is composed of cards described on the following pages.

The input deck structure is shown in Figure 4. The first system logical record consists, of course, of the control cards. The control cards are terminated with an end-of-record card (7/8/9 for punch input, *EOR for Brookhaven Intercom files).

The second system logical record consists of the RATRAP input cards. Aside from the first 2 cards, which specify the output device and user name, the input is organized into groups of cards, each group requesting a plot. These groups are called "frames." The frames are separated by blank cards.

Within each frame, each card specifies a curve to be drawn on the plot. A plot can consist of several curves. The curve specification cards also contain requests for certain options.

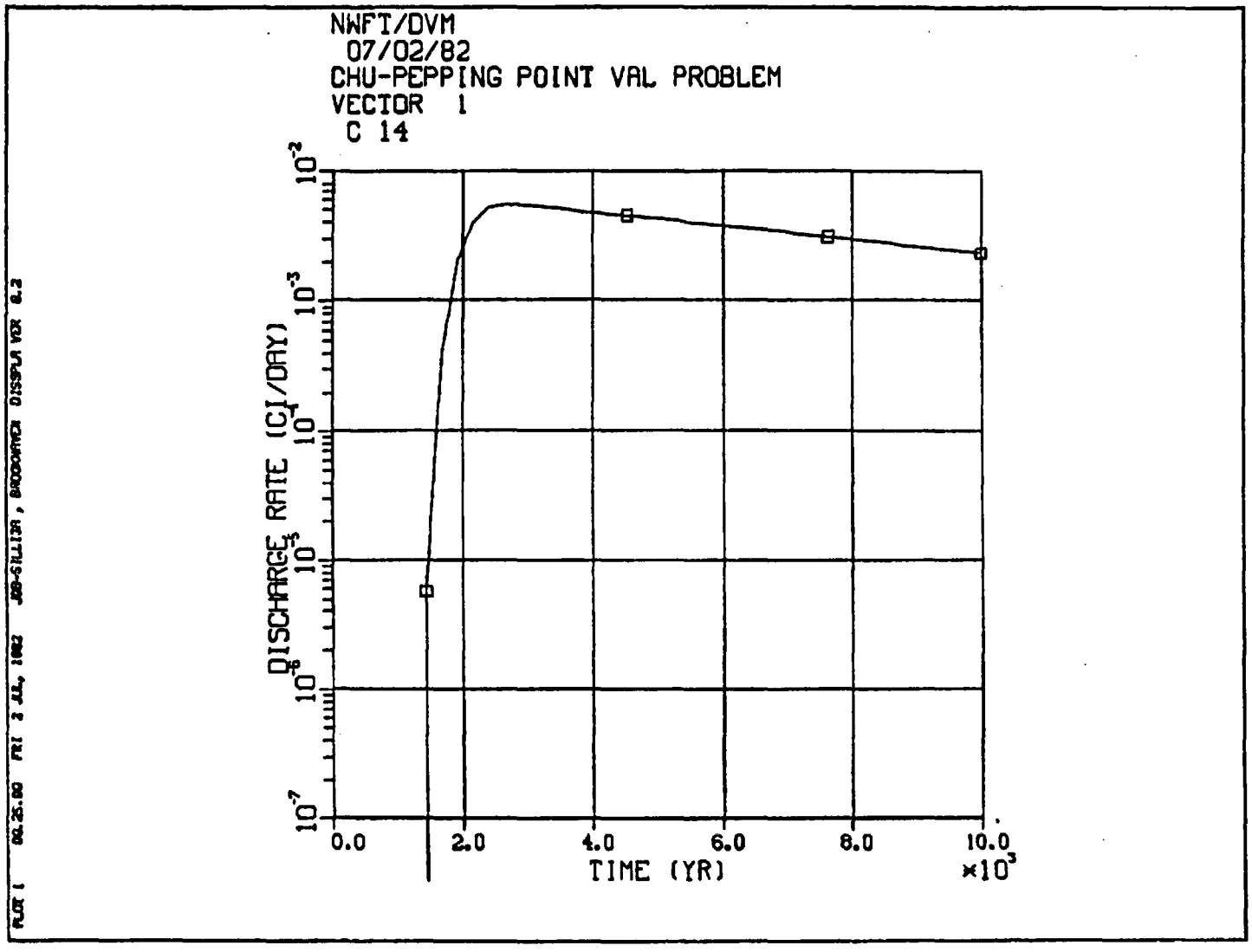

Figure 3. Curve produced from curve specification request discussed in text.

```
control cards {BOZO,STMFZ,TPO,T37.
               ACCOUNT,BOZO,9999.
               COPYSP,INPUT,OUTPUT.
               REWIND,INPUT.
               ATTACH,TAPE8,BOZOTAPE8,ID=ZZRNRC,MR=1.
               ATTACH,RATRAP,SILLINGRATRAP,ID=ZZRNRC,MR=1.
               SKIPF,RATRAP,2,17.
               COPYP,RATRAP,ABSR.
               FILE,TAPE99,RT=W,BT=I.
               ABSR.
               end-of-record (7/8/9 punch or *EOR on INTERCOM)
input         {CALCOMP
               BOZO x74173
               000 001 001 002 001 004

               000 001 001 001 002 003     LINY
               000 001 001 001 002 004

               -1
```

Figure 4. Sample deck structure.

Examples: (for the Brookhaven system):

1) Drum plotter output

```
BOZO,STMFZ,TPO,T37.
ACCOUNT,BOZO,9999.
COPYSP,INPUT,OUTPUT.
REWIND,INPUT.
ATTACH,TAPE8,BOZOTAPE8,ID=ZZRNRC,MR=1.
ATTACH,RATRAP,SILLINGRATRAP,ID=ZZRNRC,MR=1.
SKIPF,RATRAP,2,17.
COPYP,RATRAP,ABSR.
FILE,TAPE99,RT=W,BT=I.
ABSR.
end-of-record
```

2) Tektronix output

```
BOZO,STMFZ,TPO,T37.
ACCOUNT,BOZO,9999.
COPYSP,INPUT,OUTPUT.
REWIND,INPUT.
ATTACH,TAPE8,BOZOTAPE8,ID=ZZRNRC,MR=1.
ATTACH,RATRAP,SILLINGRATRAP,ID=ZZRNRC,MR=1.
SKIPF,RATRAP,3,17.
COPYP,RATRAP,ABSR.
FILE,TAPE99,RT=W,BW=I.
ABSR.
EXIT,U.
CATALOG,TAPE99,PLOTFILE,ID=BOZO,ST=MFA.
end-of-record
```

The above commands will cause the Tektronix file to be stored as a permfile on MFA. The plots would then be produced during an interactive session on MFA at the Tektronix terminal according to the instructions in NUREG/BR-0022, "Computer User's Guide," page III-33.

3) Survey only (no plots)

```
BOZO,STMFZ,TPO,T37.
ACCOUNT,BOZO,9999.
ATTACH,TAPE8,BOZOTAPE8,ID=ZZRNRC,MR=1.
ATTACH,RATRAP,SILLINGRATRAP,ID=ZZRNRC,MR=1.
SKIPF,RATRAP,2,17.
COPYP,RATRAP,ABSR.
ABSR,SURVEY.
end-of-record
```

CARD TYPE 1
DEVICE TYPE

Required: 1 per run
Limit: 1 per run

Format: A10

Place the word "TEKTRONIX" or "CALCOMP" starting in column 1, depending on the device desired.

Note:

Different device types will require different control cards.

CARD TYPE 2
PROGRAMMER NAME

Required: 1 per run
Limit: 1 per run

Format: A20

Place your name and phone number in columns 1 through 20.

CARD TYPE 3
CURVE SPECIFICATION

Required: 1 per frame
Limit: LIMAVAL per frame (see App. B)

Format: 10(I3, 1X)

Each curve to be drawn in a frame is specified by a series of 3-digit numbers separated by blanks. This series of numbers is found from the survey which appears in the printout. The first 3-digit number (attribute 1) is always 000. The remaining numbers correspond to the attributes of the data desired, as found from the survey.

The total number of 3-digit numbers on the card depends on the number of attributes by which numerical data was described on the plot tape. This equals the number of attributes listed in the survey.

To the right of the series of 3-digit numbers appear options for the plot format. The options on any curve specification card apply to the entire frame which contains the card. Available options and their meanings are listed below:

	LOGX	x-axis is logarithmic
or	LINX (default)	x-axis is linear
	LOGY (default)	y-axis is logarithmic
or	LINY	y-axis is linear
	SCAT	scatter plot (data points not connected)
or	CURVE (default)	connect the data points with a smooth curve

The options may be placed anywhere to the right of the series of 3-digit numbers, in any order.

<u>Examples</u>:

000 001 001 005 001

000 001 001 001 003 002 001 SCAT LOGX

If more than one curve specification card appears in a frame, the curves are overlaid. In this case, only one attribute may differ between the curve specification cards for that frame.

Examples:

```
000 001 001 005 001
000 001 001 008 001

000 001 001 001 003 002 001     SCAT LOGX
000 001 001 001 003 002 002
000 001 001 001 003 002 004
```

CARD TYPE 4
END OF FRAME

Required: 1 per frame
Limit: 1 per frame

This is a blank card which is placed after all the curve specifications for the frame.

CARD TYPE 5
END OF INPUT

Put a card with a -1 starting in column 1 to terminate the input deck for the run.

Appendix A. Changes to NWFT/DVM input

There are four modifications to an NWFT/DVM input deck:

(1) A dollar sign ($) must follow the title on the title card.

(2) The first isotope card must include a list of the numbers of all the nuclides for which data should be included in the plot tape. Up to 9 such nuclides may be specified.

The format of the first isotope card is now as follows:

```
variables:  NOISO,(ISOPLT(I),I=1,9)
format:     10I5
```

The numbers used to specify the nuclides on the plot tape correspond to the NCP values on the remainder of the isotope cards.

Note that there are certain non-nuclide-specific data (e.g., total EPA discharge rate) which are always written on the plot tape regardless of user input.

The number of nuclides for which data is written to the plot tape should be chosen so that the length of the plot tape will not be excessive (about 10,000 records). This is usually only a consideration for runs with many vectors.

(3) There is a third field, REFMAS, on the leach/dispersivity card. This variable contains the total repository MTHM which corresponds to the initial inventories used in the run. Default is 1000 MTHM.

The format of the leach/dispersivity card is now as follows:

```
variables:  LEACH,ALPHA,REFMAS
format:     3E10.0
```

(4) The end of the NWFT/DVM input deck should be signified by placing a 2 in column 1.

Appendix B. Current limits.

Variable	Meaning	Current Limit
LIMPNT	Maximum number of points that can be plotted on a curve.	3100
LIMAVAL	Maximum number of attribute values.	510
LIMATYP	Maximum number of attributes.	10
LIMCOO	Maximum number of coordinates.	4 (but currently only x-y plots can be made)

Appendix C. Example.

This example illustrates most of the capabilities of the current implementation on NWFT/DVM.

The NWFT/DVM run uses 10 vectors from an LHS sample which varies the leach time and groundwater travel time. The nuclides modeled are C14, Tc99, Sn126, and U238.

Of the four nuclides, only C14, Sn126, and Tc99 are written to the plot tape (see the first isotope card).

The following plots are produced by RATRAP:

(1) Discharge rate of C14 as function of time in vector 4. (Figure C.1)

(2) Overlay of integrated discharge of C14, Tc99, and Sn126, in vector 10. (Figure C.2)

(3) Overlay of total EPA ratio in vectors 1, 5, 4, 7, and 10. (Figure C.3)

(4) Overlay of total health effects in vectors 1, 5, 4, 7, and 10. (Figure C.4)

(5) Scatter plot for vectors which do or do not fail the EPA standard, with leach time and travel time variable. (Figure C.5)

(6) CCDF of EPA ratios for vectors which do or do not fail 10 CFR 60. (Figure C.6)

```
SILLING,STMFZ,TP0,T37.                          STDT2
ACCOUNT,SILLING,1349.
FILE,M,RT=Z,BT=C,FL=90.
ATTACH,M,STDUPDATES,ID=SAS,ST=MFA,MR=1.
COPYSP,M,OUTPUT.
REWIND,M.
COPYSP,INPUT,OUTPUT.
REWIND,INPUT.
COPY,M,MODS.
REWIND,MODS.
ATTACH,OLDPL,ENGDVM,ID=ZZRNRC,MR=1.
UPDATE,F,I=MODS,L=0.
FTN,I,L=0,LCM=I,PL=10000.
RETURN,COMPILE,OLDPL.
ATTACH,IMSLX,MR=1.
LIBRARY,IMSLX.
REQUEST,TAPE8,*PF.
ATTACH,TAPE10,SILLINGSTDT2SAMPLE,ID=ZZRNRC,MR=1.
LDSET,PRESET=ZERO.
LGO.
PURGE,S,SILLINGTAPE8,ID=ZZRNRC.
EXIT,U.
CATALOG,TAPE8,SILLINGTAPE8,ID=ZZRNRC.
EXIT,U.
ATTACH,RATRAP,SILLINGRATRAP,ID=ZZRNRC,MR=1.
SKIPF,RATRAP,3,17.
COPYP,RATRAP,ABS.
REQUEST,TAPE99,*PF.
FILE,TAPE99,RT=W,BT=I.
ABS,SURVEY.
EXIT,U.
COMMENT. CATALOG,TAPE99,PLOTFILE,ID=SAS,ST=MFA.
*EOR
VARY LEACH, TRAVEL$                                                    PRO
0110011010000100000010 0                                               OPT
        0.  671.2383  -22.7122                                         PRE
```

```
37)          .2829      .2829      .2829      .2829  2.835E-6    1.E-20     1.E-20     1.E-20CON
38)         1.E-20  2.835E-6  2.835E-6     1.E-20      100.     1.E-20     1.E-20           CON
39)         1.29E6    1.29E6    1.29E6     1.29E6    8.61E6         1.         1.         1.ARE
40)             1.    8.61E6    8.61E6         1.    8.61E6         1.         1.           ARE
41)            .01       .01     5248.        .01       .01      200.      5248.        .01LEN
42)           705.      705.      705.       705.      200.     1410.      1410.           LEN
43)             0.    -1410.        0.         0.        0.     -705.      -705.    -1410.ELE
44)         -1410.        0.    -1410.     -1410.                                           ELE
45)            .15       .15       .15        .15     1.E-4     1.E-4      1.E-4     1.E-4POR
46)          1.E-4     1.E-4     1.E-4      1.E-4     1.E-4     1.E-4      1.E-4           POR
47)                                                                                          BRI
48)                                                                                          BRI
49)         4   13   10    3    4                                                           MIG
50)         4    1    2    3                                                                ISO
51)            14.     C 14         1         0    5.74E3     2.8E4                        ISO   1
52)            99.    TC 99         2         0    5.12E5     1.4E6                        ISO   2
53)           126.    SN126         3         0    1.00E5     5.6E4                        ISO   3
54)           238.     U238         4         0    4.52E9     3.1E4                        ISO   4
55)                                 0.        0.                                            KD    1
56)                                 0.        0.                                            KD    2
57)                             .009343   .009343                                           KD    3
58)                                 0.        0.                                            KD    4
59)           1.E4      100.      1.E5                                                      LEA
60)         2                                                                                DVM
61)            10.     1.E-9     1.E-6       10.                                            SOL
62)             10      1.E4     1000.        0      1.E4                                   TIM
63)     2                                                                                    END
```

```
SURVEY COMPLETE      6736 RECORDS READ


ATTRIBUTE TYPE    2 HAS NAME CODE
    VALUE    1  HAS TITLE NWFT/DVM$

ATTRIBUTE TYPE    3 HAS NAME DATE
    VALUE    1  HAS TITLE  07/06/82 $

ATTRIBUTE TYPE    4 HAS NAME PROBLEM
    VALUE    1  HAS TITLE VARY LEACH, TRAVEL$

ATTRIBUTE TYPE    5 HAS NAME VECTOR
    VALUE    1  HAS TITLE VECTOR  1$
    VALUE    2  HAS TITLE VECTOR  2$
    VALUE    3  HAS TITLE VECTOR  3$
    VALUE    4  HAS TITLE VECTOR  4$
    VALUE    5  HAS TITLE VECTOR  5$
    VALUE    6  HAS TITLE VECTOR  6$
    VALUE    7  HAS TITLE VECTOR  7$
    VALUE    8  HAS TITLE VECTOR  8$
    VALUE    9  HAS TITLE VECTOR  9$
    VALUE   10  HAS TITLE VECTOR 10$
    VALUE   11  HAS TITLE ALL VECTORS$
    VALUE   12  HAS TITLE VECTORS WITHIN 10CFR60$
    VALUE   13  HAS TITLE VECTORS WITHIN EPA LIM$
    VALUE   14  HAS TITLE VECTORS NOT WITHIN EPA LIM$

ATTRIBUTE TYPE    6 HAS NAME COORD1
    VALUE    1  HAS TITLE TIME (YR)$
    VALUE    2  HAS TITLE EPA RELEASE RATIO$
    VALUE    3  HAS TITLE EPA HEALTH EFFECTS$
    VALUE    4  HAS TITLE LEACH TIME (YR)$

ATTRIBUTE TYPE    7 HAS NAME COORD2
    VALUE    1  HAS TITLE INTEGRATED DISCHARGE (CI)$
    VALUE    2  HAS TITLE DISCHARGE RATE (CI/DAY)$
    VALUE    3  HAS TITLE SOURCE RATE (CI/DAY)$
    VALUE    4  HAS TITLE EPA RELEASE RATIO$
    VALUE    5  HAS TITLE EPA HEALTH EFFECTS$
    VALUE    6  HAS TITLE PROBABILITY$
    VALUE    7  HAS TITLE TRAVEL TIME (YR)$
    VALUE    8  HAS TITLE CANISTER LIFE (YR)$

ATTRIBUTE TYPE    8 HAS NAME ISOTOPE
    VALUE    1  HAS TITLE  C 14 $
    VALUE    2  HAS TITLE TC 99 $
    VALUE    3  HAS TITLE SN126 $
    VALUE    4  HAS TITLE ALL NUCLIDES$
```

```
 1)     SILLING,STMFZ,TP0,T277,P2.
 2)     ACCOUNT,SILLING,1349.
 3)     COPYSP,INPUT,OUTPUT.
 4)     REWIND,INPUT.
 5)     ATTACH,TAPE8,SILLINGTAPE8,ID=ZZRNRC,MR=1.
 6)     ATTACH,RATRAP,SILLINGRATRAP,ID=ZZRNRC,MR=1.
 7)     SKIPF,RATRAP,3,17.
 8)     COPYP,RATRAP,ABSR.
 9)     FILE,TAPE99,RT=W,BT=I.
10)     ABSR.
11)     CATALOG,TAPE99,PLOTFILE,ID=SAS,ST=MFA.
12)     *EOR
13)     TEKTRONIX
14)     S.SILLING X74133
15)     000 001 001 001 004 001 002 001
16)
17)     000 001 001 001 010 001 001 001
18)     000 001 001 001 010 001 001 002
19)     000 001 001 001 010 001 001 003
20)
21)     000 001 001 001 001 001 004 004
22)     000 001 001 001 004 001 004 004
23)     000 001 001 001 005 001 004 004
24)     000 001 001 001 007 001 004 004
25)     000 001 001 001 010 001 004 004
26)
27)     000 001 001 001 001 001 005 004
28)     000 001 001 001 004 001 005 004
29)     000 001 001 001 005 001 005 004
30)     000 001 001 001 007 001 005 004
31)     000 001 001 001 010 001 005 004
32)
33)     000 001 001 001 013 004 007 004    SCAT LOGX
34)     000 001 001 001 014 004 007 004
35)
36)     000 001 001 001 011 002 006 004    LOGX LINY
```

```
37)     000 001 001 001 012 002 006 004
38)
39)     -1
```

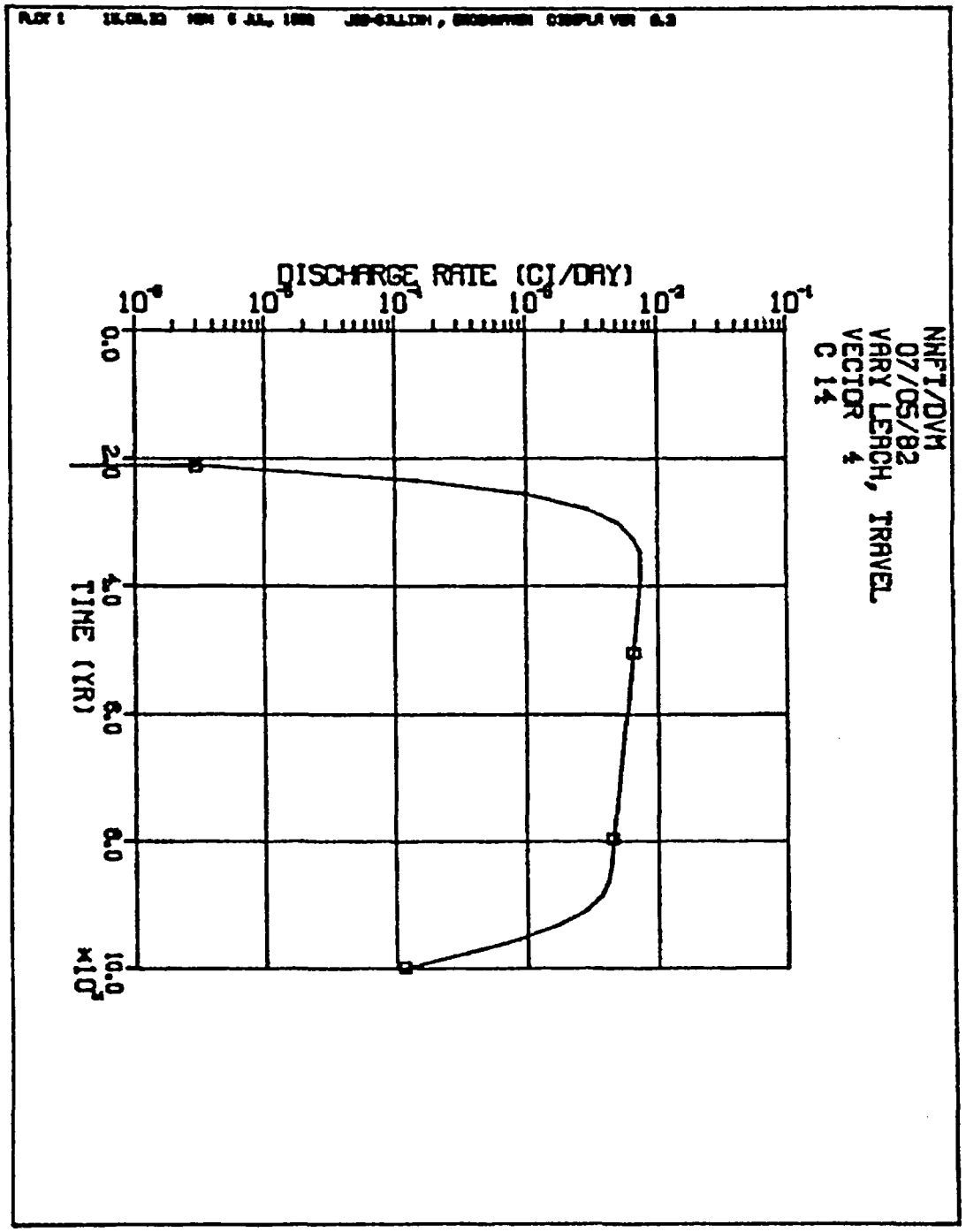
NWFT/DVM
07/05/82
VARY LEACH, TRAVEL
VECTOR 4
C 14
DISCHARGE RATE (CI/DAY)
TIME (YR)
0.0
2.0
4.0
6.0
8.0
10.0

Figure C.1.

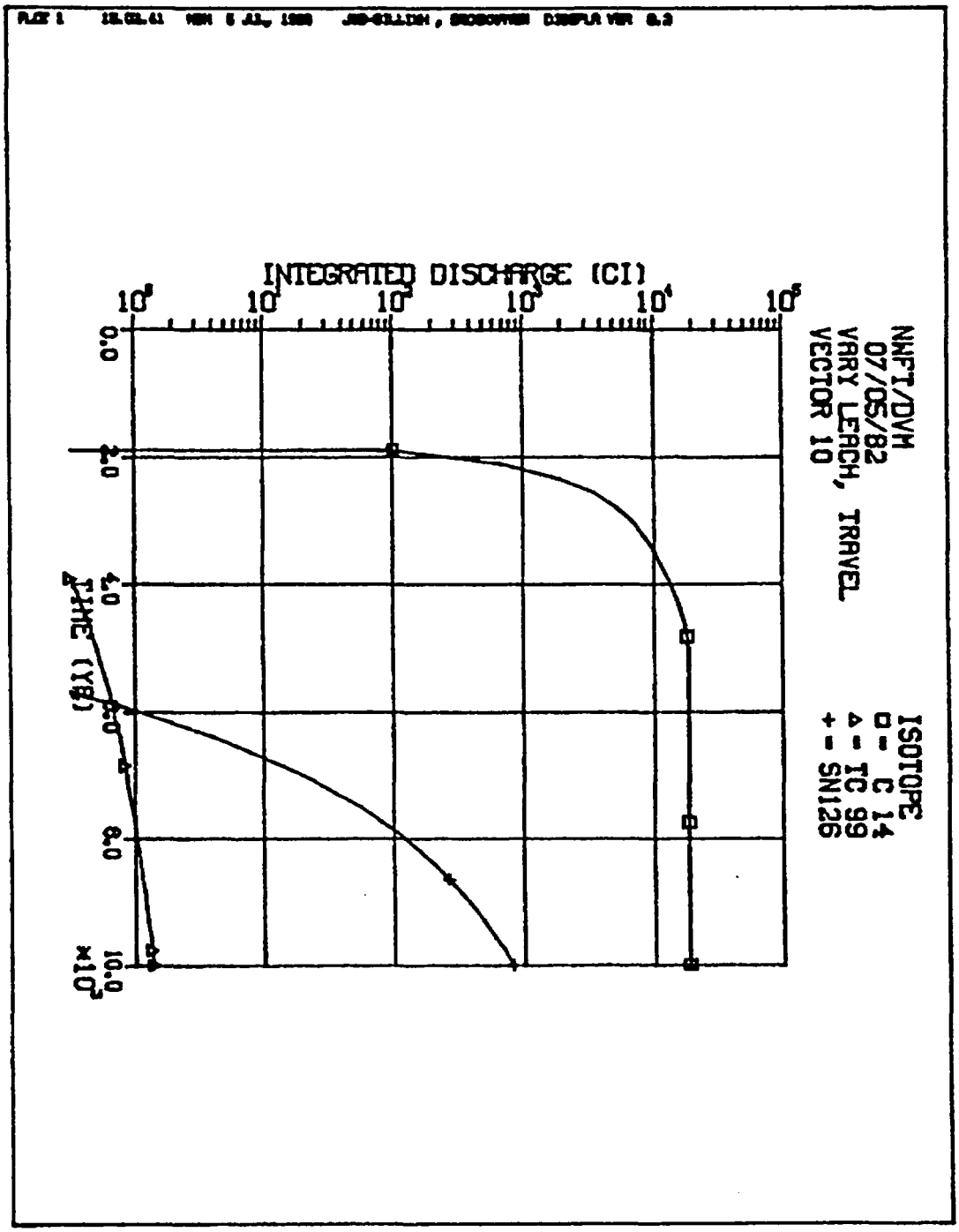
INTEGRATED DISCHARGE (CI)
NWFT/DVM
07/05/82
VARY LEACH, TRAVEL
VECTOR 10
ISOTOPE
□ - C 14
Δ - TC 99
+ - SN126
TIME (YR)
0.0
2.0
4.0
6.0
8.0
10.0
×10³

Figure C.2.

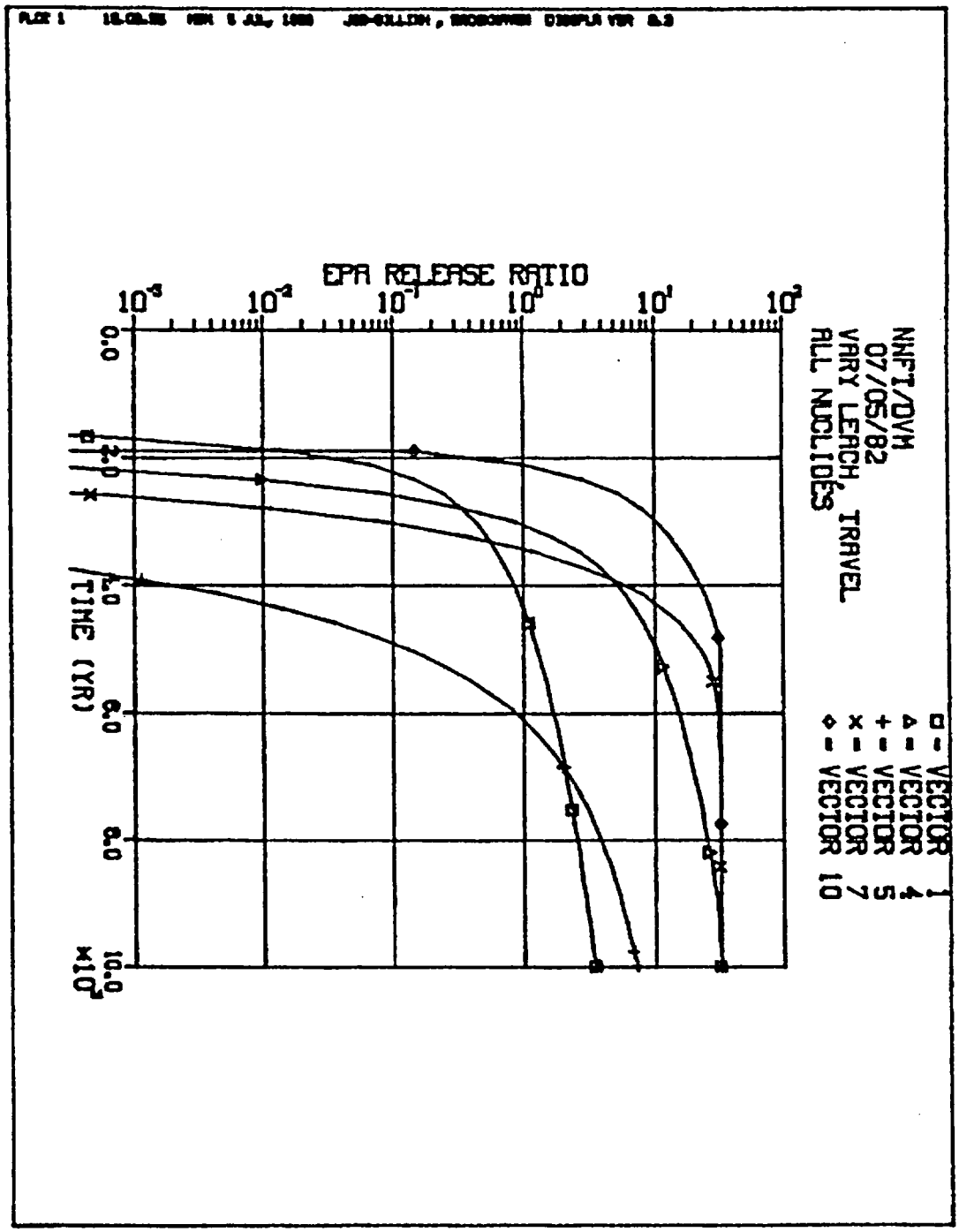
NNFI/DVH
07/05/82
VARY LEACH, TRAVEL
ALL NUCLIDES
□ -- VECTOR 1
△ -- VECTOR 4
+ -- VECTOR 5
× -- VECTOR 7
◇ -- VECTOR 10
EPA RELEASE RATIO
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
0.0
2.0
4.0
6.0
8.0
10.0
TIME (YR)
$\times 10^{4}$

Figure C.3.

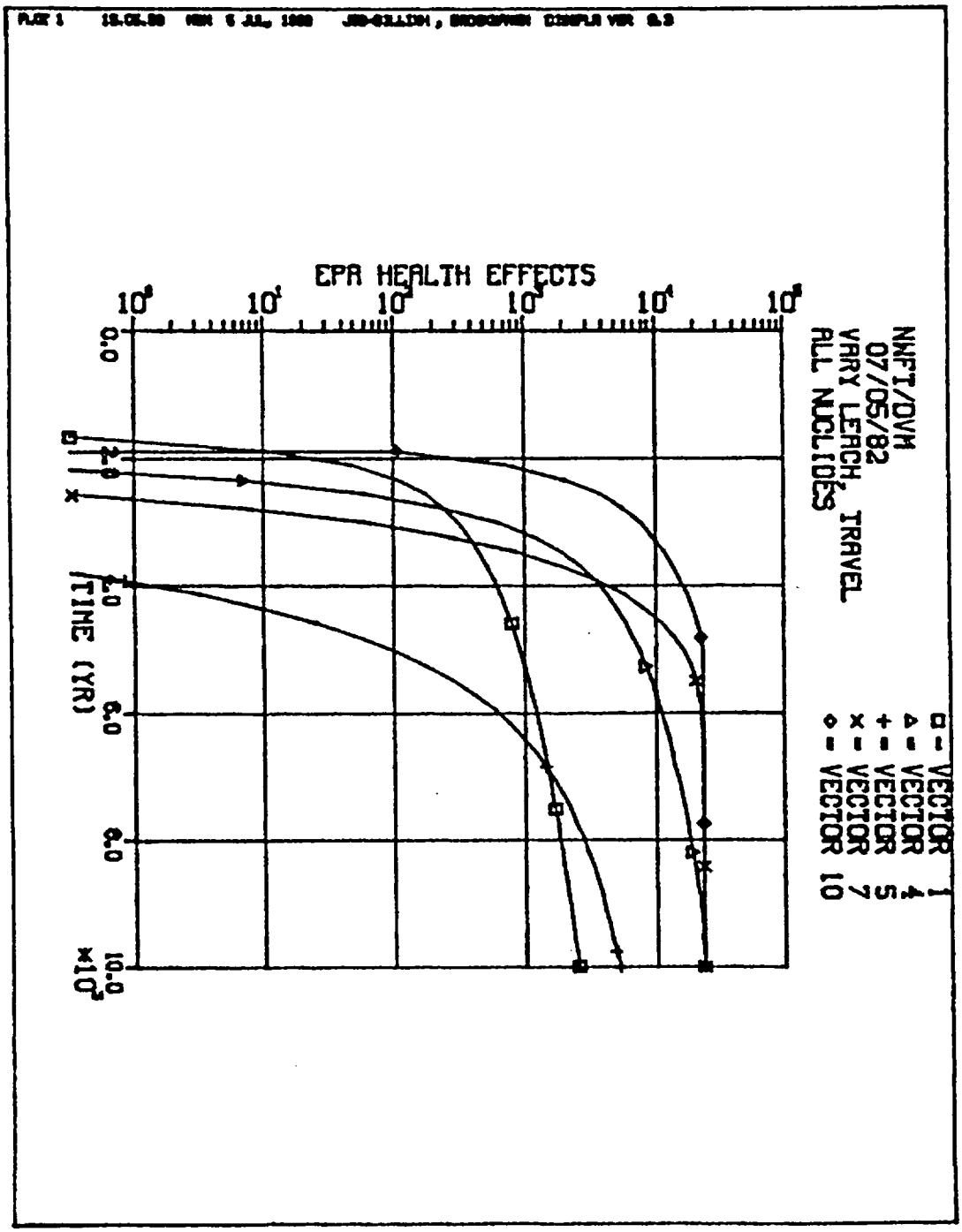
NNFT/OVH
07/05/82
VARY LEACH, TRAVEL
ALL NUCLIDES
□ -- VECTOR 1
Δ -- VECTOR 4
+ -- VECTOR 5
x -- VECTOR 7
◇ -- VECTOR 10
EPA HEALTH EFFECTS
TIME (YR)
0.0
2.0
4.0
6.0
8.0
10.0
x10^5

Figure C.4.

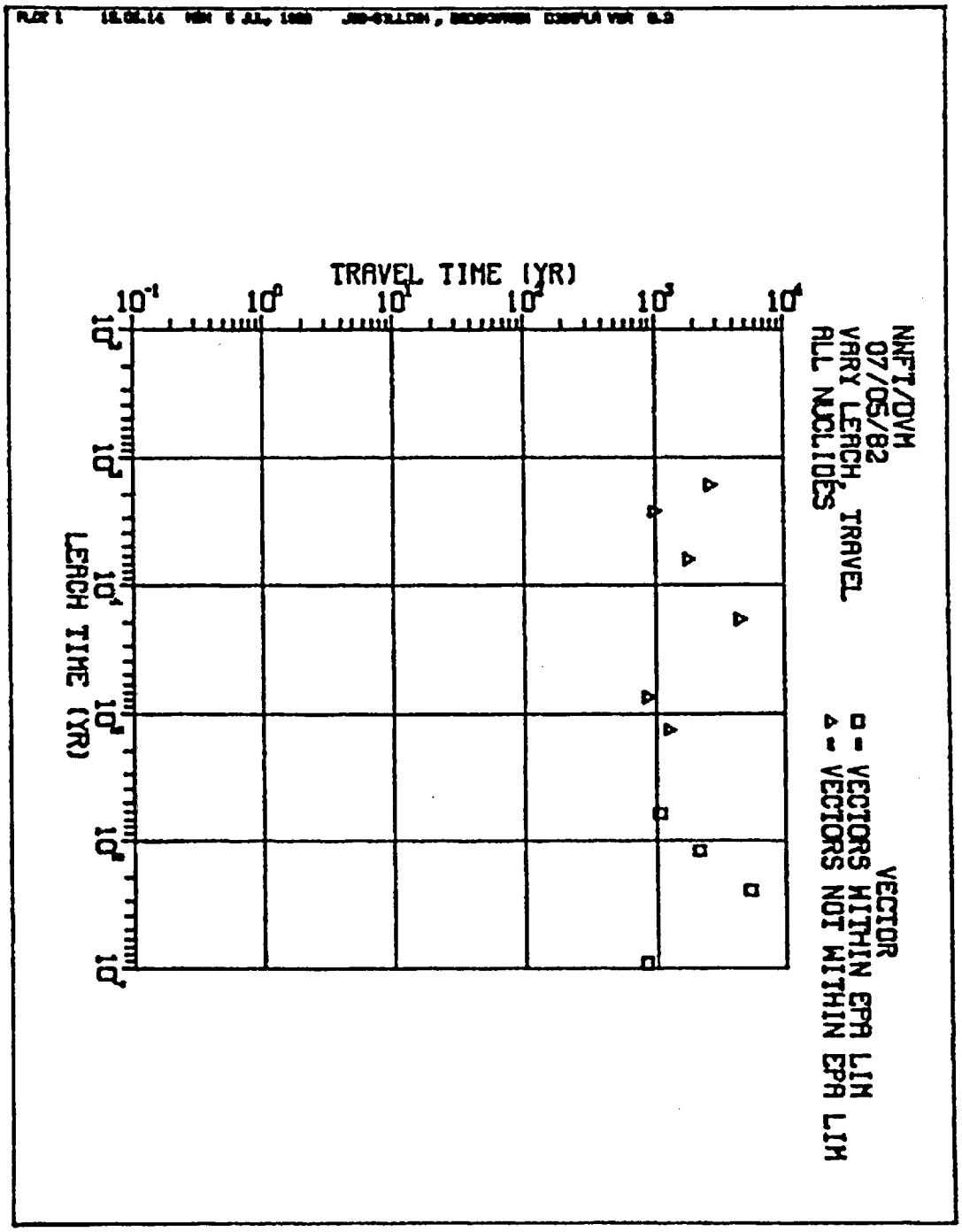
NNFT/OVM
07/06/82
VARY LEACH, TRAVEL
ALL NUCLIDES
VECTOR
□ - VECTORS WITHIN EPA LIM
△ - VECTORS NOT WITHIN EPA LIM
TRAVEL TIME (YR)
LEACH TIME (YR)

Figure C.5.

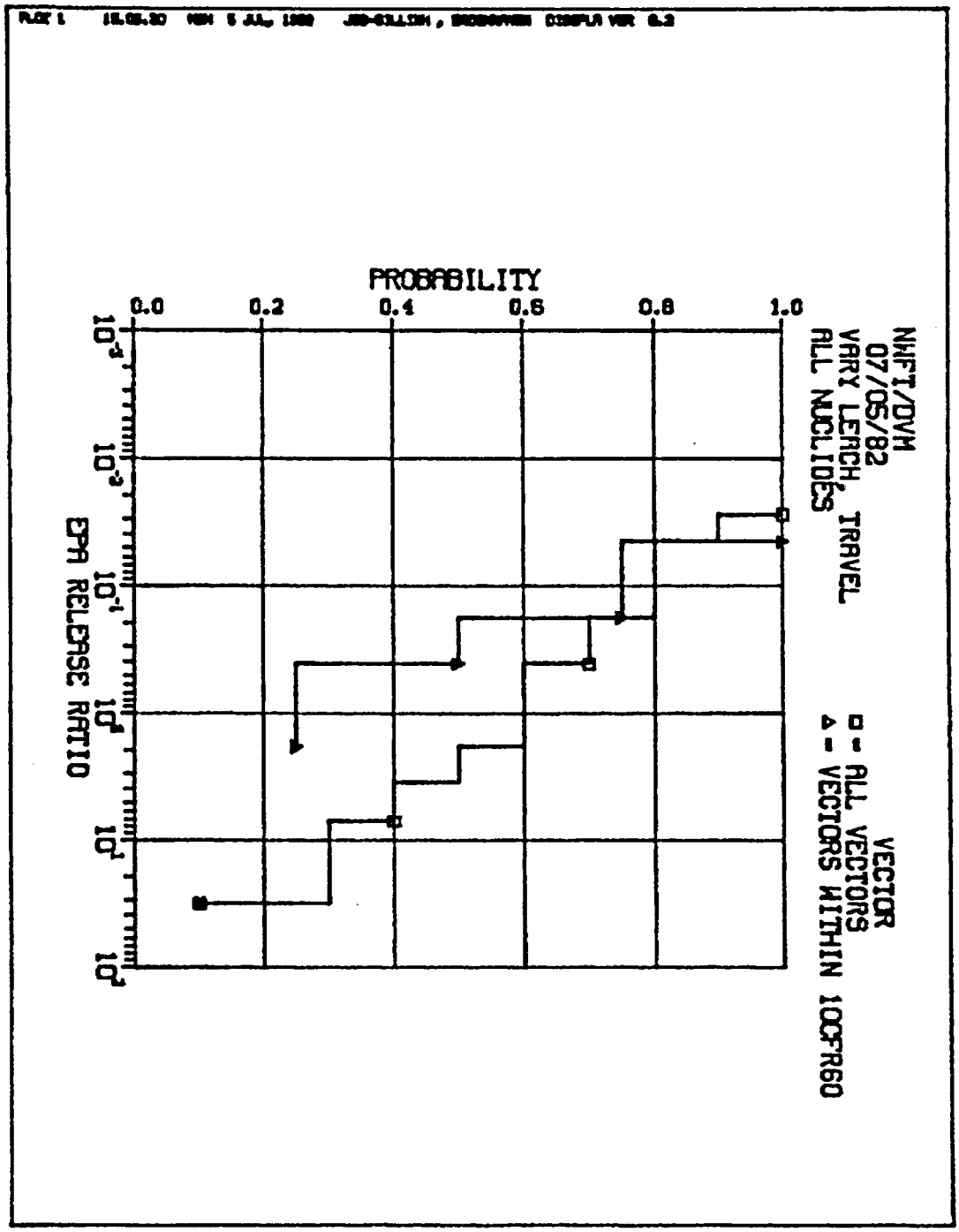
NWFT/DVM
07/06/82
VARY LERCH, TRAVEL
ALL NUCLIDES
VECTOR
□ - ALL VECTORS
▲ - VECTORS WITHIN 10CFR60
PROBABILITY
0.0
0.2
0.4
0.6
0.8
1.0
EPA RELEASE RATIO
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}

Figure C.6.

Appendix D. Modifying a transport model to produce a plot tape.

A transport model can be modified to produce plot tapes containing almost any scalar variables. There are several steps in these modifications.

(1) Declare a file for the plot tape in the PROGRAM card. TAPE8 is the most convenient.

(2) Append the subroutines HEADER, TITLE, and PTREC to the code. (See Appendix E.)

(3) Insert common block PTCOM into every subroutine from which numerical data will be written. (See Appendix E.)

(4) In the first DATA statement in subroutine TITLE, set MAXATYP to the number of attributes to be used.

(5) Change the first data statement in HEADER to reflect the desired meanings of the attributes.

(6) Change the calls to TITLE in subroutine HEADER to define the meaning of every value to be used for the attributes. (TITLE may alternatively be called from any other subroutine if more convenient.)

(7) Find the locations in the code at which point numerical data will be writen on the plot tape. For each such location:

    (a) Set the elements of the AVAL array in common block PTCOM to the attribute values which correctly describe the data point to be written.

    (b) To write a data point on the plot tape (with the name tag just defined) insert CALL PTREC (x,y) where x and y are the x- and y- coordinates.

Note:

When defining a header as described in item 5 above, there are certain restrictions. The first attribute must be called "TITLE." There must be one attribute called "COORD1" and another called "COORD2." (The

significance of these attributes is that their values are the only ones that can appear on the x- and y- axes, respectively.) No attribute may have more than 10 characters. Finally, the entire header string must be terminated with a dollar sign ($).

Appendix E. Updates to NWFT/DVM

```
*I ISOTOPE.6
C
     3 ,CURLST(120),CURINT(120),ISOPLT(09),NPLT
*INSERT      ISOTOPE.23
C-CURLST- DISCHARGE RATE IN PREVIOUS TIME STEP
C-CURINT- INTEGRATED DISCHARGE
C-ISOPLT- ISOTOPES TO BE PLOTTED BY RATRAP
C-NPLT-   NUMBER OF ISOTOPES TO BE PLOTTED BY RATRAP
*D MISC.4
     1 TTL(8),NLEG,NJCT,X,Y,ALPHA,LEACH,DTPRT
*I MISC.25
C DTPRT   INTERVAL (YEARS) BETWEEN DISCHARGE PRINTOUTS
*INSERT      ANAMOD.2
     *   TAPE2,
     *   TAPE8,
*INSERT      ANAMOD.48
C  TAPE  2 FOR STEPWISE REGRESSION AND TRADE-OFF PLOT FILE
*I ANAMOD.50
C  TAPE  8 FOR RATRAP PLOT TAPE
*INSERT      ANAMOD.55
*CALL EPA
*INSERT      ANAMOD.59
*CALL LEGPROP
*INSERT      ANAMOD.62
*CALL PTCOM
*DELETE      ANAMOD.67,ANAMOD.67
*CALL SANS
      DIMENSION SCRATCH(8)
*DELETE      ANAMOD.68,ANAMOD.68
      DATA MAXVEC, ND, NEQN /1023, 9, 9/
*INSERT      ANAMOD.78
C
C WRITE PLOT TAPE HEADER AND PROBLEM TITLE
C
      CALL HEADER
      LVECT = 0
```

```
37)     *DELETE       ANAMOD.88,ANAMOD.88
38)           READ(5,9001) NOVEC, TUB, TRLSE, NOSKIP, DTPRT
39)           DTPRT = AMAX1(DTPRT, 1.E-90)
40)     *INSERT       ANAMOD.95
41)     C
42)     C SET SPECIAL VECTOR TITLES
43)     C
44)           CALL TITLE(5, NOVEC+1, 'ALL VECTORS$')
45)           CALL TITLE(5, NOVEC+2, 'VECTORS WITHIN 10CFR60$')
46)           CALL TITLE(5, NOVEC+3, 'VECTORS WITHIN EPA LIM$')
47)           CALL TITLE(5, NOVEC+4, 'VECTORS NOT WITHIN EPA LIM$')
48)     *I ANAMOD.96
49)     C
50)     C INITIALIZE NUMBER OF PART 60 VECTORS
51)     C
52)           NPT60 = 0
53)     *INSERT       ANAMOD.101
54)     C
55)     C          WRITE VECTOR TITLE TO PLOT TAPE
56)     C
57)           LVECT = LVECT + 1
58)           ENCODE(10,9400,SCRATCH) LVECT
59)      9400 FORMAT('VECTOR', I3, '$')
60)           CALL TITLE(5,LVECT,SCRATCH)
61)     C
62)     C ZERO OUT DISCHARGE ARRAYS
63)     C
64)           DO 14 IS=1,NOISO
65)           CUROUT(IS) = 0.
66)           CURLST(IS) = 0.
67)           CURINT(IS) = 0.
68)        14 CONTINUE
69)     *INSERT       ANAMOD.128
70)     C
71)     C SET FLAG IF THIS VECTOR IS WITHIN 10 CFR 60 CRITERIA
72)     C
```

```
      INPT60(NORV) = 0
      IF(LEACH.GT.0.99999E5.AND.TRLSF.GT.999.99.AND.
     *   TRTIME.GT.999.99) INPT60(NORV) = 1
      IF(INPT60(NORV).EQ.1) NPT60 = NPT60 + 1
*INSERT      ANAMOD.133
C
C WRITE RECORD FOR STEPWISE REGRESSION AND TRADE-OFF PLOT FILE
C
      WRITE(2) (RV(I),I=1,NRV), LEACH, TRLSE, TRTIME, EPASUM,
     *   (EPAINT(IS),IS=1,NOISO)
C
C WRITE PLOT RECORD FOR SCATTER PLOTSS
C
      IF(EPASUM.LE.1.0) AVAL(5) = NOVEC + 3
      IF(EPASUM.GT.1.0) AVAL(5) = NOVEC + 4
      AVAL(6) = 4
      AVAL(8) = NPLT + 1
      AVAL(7) = 7
      CALL PTREC(LEACH, TRTIME)
      AVAL(7) = 8
      CALL PTREC(LEACH, TRLSE)
*INSERT      ANAMOD.143
C
C SAVE EPA RELEASE FRACTION AND HEALTH EFFECTS
C
      SAMSUM(NORV) = EPASUM
      SAMRSK(NORV) = EPARSK
*INSERT      ANAMOD.159
C
C SORT RELEASE FRACTIONS
C
      IF(NOVEC.LE.1) GO TO 73
      DO 52 I=1,NOVEC
      VECNUM(I) = I
   52 CONTINUE
      IF(NOVEC.GT.1) CALL SSORT(SAMSUM, VECNUM, NOVEC, 2)
```

```
      WRITE(6,9500)
 9500 FORMAT(//// ' EPA RELEASE FRACTIONS BY VECTOR' //)
      WRITE(6,9201) (I, VECNUM(I), SAMSUM(I), I=1,NOVEC)
C
C WRITE CCDF PLOT RECORDS FOR EPA RELEASE FRACTIONS
C
      AVAL(5) = NOVEC + 1
      AVAL(6) = 2
      AVAL(7) = 6
      AVAL(8) = NPLT + 1
      DO 53 I=1,NOVEC
      PROB = 1.0 - (I - 1) / FLOAT(NOVEC)
      TEMP = SAMSUM(I)
      CALL PTREC(TEMP, PROB)
      IF(I.NE.NOVEC) CALL PTREC(TEMP, PROB-1.0/NOVEC)
   53 CONTINUE
C
C WRITE CCDF PLOT RECORDS FOR EPA FRACTIONS OF VECTORS WITHIN
C 10 CFR 60
C
      AVAL(5) = NOVEC + 2
      AVAL(6) = 2
      AVAL(7) = 6
      AVAL(8) = NPLT + 1
      J = 0
      DO 71 I=1,NOVEC
      K = IFIX(VECNUM(I) + 0.1)
      IF(INPT60(K).NE.1) GO TO 71
      J = J + 1
      PROB = 1.0 - (J - 1) / FLOAT(NPT60)
      TEMP = SAMSUM(I)
      CALL PTREC(TEMP, PROB)
      IF(J.NE.NPT60) CALL PTREC(TEMP, PROB-1.0/NPT60)
   71 CONTINUE
      WRITE(6,9502) NPT60
 9502 FORMAT(' NUMBER OF VECTORS WITHIN 10CFR60=', I5)
```

```
C
C SORT HEALTH EFFECTS
C
      DO 54 I=1,NOVEC
      VECNUM(I) = I
   54 CONTINUE
      IF(NOVEC.GT.1) CALL SSORT(SAMRSK, VECNUM, NOVEC, 2)
      WRITE(6,9501)
 9501 FORMAT(//// ' EPA HEALTH EFFECTS BY VECTOR' //)
      WRITE(6,9201) (I, VECNUM(I), SAMRSK(I), I=1,NOVEC)
C
C WRITE CCDF PLOT RECORDS FOR EPA HEALTH EFFECTS
C
      AVAL(5) = NOVEC + 1
      AVAL(6) = 3
      AVAL(7) = 6
      AVAL(8) = NPLT + 1
      DO 55 I=1,NOVEC
      PROB = 1.0 - (I - 1) / FLOAT(NOVEC)
      TEMP = SAMRSK(I)
      CALL PTREC(TEMP, PROB)
      IF(I.NE.NOVEC) CALL PTREC(TEMP, PROB=1.0/NOVEC)
   55 CONTINUE
C
C WRITE CCDF PLOT RECORDS FOR EPA HEALTH EFFECTS FROM VECTORS
C WITHIN 10 CFR 60
C
      AVAL(5) = NOVEC + 2
      AVAL(6) = 3
      AVAL(7) = 6
      AVAL(8) = NPLT + 1
      J = 0
      DO 72 I=1,NOVEC
      K = IFIX(VECNUM(I) + 0.1)
      IF(INPT60(K).NE.1) GO TO 72
      J = J + 1
```

```
      PROB = 1.0 - (J - 1) / FLOAT(NPT60)
      TEMP = SAMRSK(I)
      CALL PTREC(TEMP, PROB)
      IF(J.NE.NPT60) CALL PTREC(TEMP, PROB-1.0/NPT60)
   72 CONTINUE
   73 CONTINUE
*DELETE      ANAMOD.182,ANAMOD.182
 9001 FORMAT(I10, 2E10.0, I10, 2E10.0)
*DELETE      ANAMOD.188,ANAMOD.188
 9201 FORMAT(/5(* RANK RVECTOR  DISCHARGE  *)/5(1X,I4,F6.0,2X,
     *   E11.4,2X))
*DELETE          ET.10,ET.10
      EXPARG = -D(1) * T
      EXPARG = AMAX1(-600.0, EXPARG)
      DF = EXP(EXPARG)
*INSERT      FLOWIN.3
*CALL EPA
*D FLOWIN.23
      IF(EOF(5).EQ.0.AND.INP(1).LE.1) GO TO 1
*DELETE      FLOWIN.123,FLOWIN.123
      READ(5,9002) NOISO, (ISOPLT(I),I=1,09)
C
C COUNT THE NUMBER OF ISOTOPES TO BE PLOTTED
C
      NPLT = 0
      DO 13 I=1,09
      IF(ISOPLT(I).LE.0) GO TO 14
      NPLT = I
   13 CONTINUE
   14 CONTINUE
*I FLOWIN.144
C
C INITIALIZE EPA LIMIT ARRAYS
C
      CALL EPAINI
*DELETE      FLOWIN.165,FLOWIN.165
```

```
217)          READ(5,9000) LEACH, ALPHA, REFMAS
218)     *DELETE       FLOWIN.167,FLOWIN.167
219)          WRITE(6,9006) LEACH, ALPHA, REFMAS
220)     *DELETE       FLOWIN.219,FLOWIN.219
221)         *= 1PE12.3, ' FT', 5X, 'ASSUMED MASS FOR EPA FRACTIONS',
222)         *1PE13.6, ' MTHM')
223)     *I GETRV.23
224)     C
225)     C READ RANDOM VARIABLES FOR TEST PROBLEM 2
226)     C
227)          LEACH = RV(1)
228)          PHI(10) = RV(2)
229)     *DELETE          GIT.24,GIT.24
230)          EXPARG = X / AL - (Z + B) ** 2
231)          EXPARG = AMAX1(-600.0, EXPARG)
232)          TERM2 = POLYETA * EXP(EXPARG) * TRPI
233)     *DELETE          GIT.28,GIT.28
234)          EXPARG = -(Z - B) ** 2
235)          EXPARG = AMAX1(-600.0, EXPARG)
236)          TERM1 = POLYETA * EXP(EXPARG) * TRPI
237)     *INSERT      PATHLEN.3
238)     *CALL EPA
239)     *INSERT      PATHLEN.8
240)     *CALL OUTVC
241)     *INSERT      PATHLEN.106
242)          TRTIME = PATH(10) / PORE(10) + PATH(3) / PORE(3) +
243)         *   PATH(4) / PORE(4)
244)          WRITE(6,1000) TRTIME, LEACH, TRLSE
245)      1000 FORMAT(' TRAVEL TIME=', 1PE13.6, 3X,
246)         *   ' LEACH TIME=', 1PE13.6, 3X,
247)         *   ' RELEASE TIME=', 1PE13.6)
248)     *DELETE          PRP.13,PRP.13
249)          EXPARG = ZZ
250)          EXPARG = AMAX1(-600.0, EXPARG)
251)          F1 = EXP(EXPARG)
252)     *INSERT        SSORT.53
```

```
253)          LEVEL 2, X, Y
254)     *INSERT      SOURCE.6
255)     *CALL PTCOM
256)     *INSERT      SOURCE.7
257)     *CALL TIME
258)     *INSERT      SOURCE.184
259)     C
260)     C WRITE SOURCE RATE PLOT RECORD
261)     C
262)          IF(IC.EQ.0) GO TO 831
263)          AVAL(6) = 1
264)          AVAL(7) = 3
265)          IF(NPLT.EQ.0) GO TO 831
266)          DO 830 LR=1,NPLT
267)          IR = ISOPLT(LR)
268)          AVAL(8) = LR
269)          SRATE = S(IR) / (365. * DT * CI2ATOM(IR))
270)          CALL PTREC(TIME+TRLSE, SRATE)
271)      830 CONTINUE
272)      831 CONTINUE
273)     *INSERT       TPPRT.2
274)     *CALL EPA
275)     *INSERT       TPPRT.5
276)     *CALL MISC
277)     *CALL PTCOM
278)     C
279)          DATA INIT /1/
280)     C
281)     C          WRITE ISOTOPE NAMES TO PLOT TAPE
282)     C          NPLT+1 REPRESENTS THE SUM OF ALL ISOTOPES
283)     C
284)          IF(INIT.EQ.0) GO TO 17
285)          INIT = 0
286)          IF(NPLT.EQ.0) GO TO 16
287)          DO 15 LS=1,NPLT
288)          IS = ISOPLT(LS)
```

```
      ENCODE(7,1010,SCRATCH) ISONAME(IS)
 1010 FORMAT(A6, 1H$)
      CALL TITLE(8,LS,SCRATCH)
   15 CONTINUE
C
C          SET SPECIAL ISOTOPE NAMES
C
   16 CALL TITLE(8,NPLT+1,'ALL NUCLIDES$')
   17 CONTINUE
*INSERT        TPPRT.13
C
C          WRITE DISCHARGE RATE PLOT RECORD
C
      AVAL(6) = 1
      AVAL(7) = 2
      IF(NPLT.EQ.0) GO TO 151
      DO 150 LS=1,NPLT
      IS = ISOPLT(LS)
      AVAL(8) = LS
      CALL PTREC(T,CUROUT(IS))
  150 CONTINUE
  151 CONTINUE
C
C FIND INTEGRATED DISCHARGE
C
      DO 155 IS=1,NOISO
      CURINT(IS) = CURINT(IS) + 0.5 * DT * (CURLST(IS) +
     *   CUROUT(IS)) * 365.
      CURLST(IS) = CUROUT(IS)
  155 CONTINUE
C
C WRITE INTEGRATED DISCHARGE PLOT RECORD
C
      AVAL(6) = 1
      AVAL(7) = 1
      IF(NPLT.EQ.0) GO TO 161
```

```
      DO 160 LS=1,NPLT
      IS = ISOPLT(LS)
      AVAL(8) = LS
      CALL PTREC(T,CURINT(IS))
  160 CONTINUE
  161 CONTINUE
C
C CALCULATE EPA RATIOS AND SUM
C
      EPASUM = 0.
      DO 170 IS=1,NOISO
      EPAINT(IS) = CURINT(IS) * 1000. / (EPALIM(IS) * REFMAS)
      EPASUM = EPASUM + EPAINT(IS)
  170 CONTINUE
C
C WRITE EPA RATIO PLOT RECORD
C
      AVAL(6) = 1
      AVAL(7) = 4
      IF(NPLT.EQ.0) GO TO 176
      DO 175 LS=1,NPLT
      IS = ISOPLT(LS)
      AVAL(8) = LS
      CALL PTREC(T, EPAINT(IS))
  175 CONTINUE
  176 CONTINUE
C
C WRITE PLOT RECORD FOR SUM OF EPA RATIOS
C
      AVAL(8) = NPLT + 1
      CALL PTREC(T, EPASUM)
C
C CALCULATE EPA HEALTH EFFECTS AND SUM
C
      EPARSK = 0.
      DO 180 IS=1,NOISO
```

```
361)          EPAHEF(IS) = CURINT(IS) * EPATOX(IS)
362)          EPARSK = EPARSK + EPAHEF(IS)
363)      180 CONTINUE
364)    C
365)    C WRITE EPA HEALTH EFFECTS PLOT RECORD
366)    C
367)          AVAL(6) = 1
368)          AVAL(7) = 5
369)          IF(NPLT.EQ.0) GO TO 186
370)          DO 185 LS=1,NPLT
371)          IS = ISOPLT(LS)
372)          AVAL(8) = LS
373)          CALL PTREC(T, EPAHEF(IS))
374)      185 CONTINUE
375)      186 CONTINUE
376)    C
377)    C WRITE EPA TOTAL RISK PLOT RECORD
378)    C
379)          AVAL(8) = NPLT + 1
380)          CALL PTREC(T, EPARSK)
381)    *DELETE       TPPRT.18,TPPRT.20
382)          FAC = AMOD(T,DTPRT) / DT
383)          IF(FAC.LT.0..OR.FAC.GE.1.) GO TO 100
384)    *DELETE       TPPRT.21,TPPRT.21
385)    *DELETE       TPPRT.23,TPPRT.23
386)          WRITE(6,2000) T, (ISONAME(J), CUROUT(J), J=1,NOISO)
387)     2000 FORMAT(//// ' TIME=', 1PE13.6, ' YEARS',
388)         *   T40, 'DISCHARGE RATE (CI/DAY)' /
389)         *   5(1X, A6, 1PE14.6, 4X) )
390)          WRITE(6,2010) T, (ISONAME(J), CURINT(J), J=1,NOISO)
391)     2010 FORMAT(/ ' TIME=', 1PE13.6, ' YEARS',
392)         *   T40, 'INTEGRATED DISCHARGE (CI)' /
393)         *   5(1X, A6, 1PE14.6, 4X) )
394)          WRITE(6,2020) T, (ISONAME(J), EPAINT(J), J=1,NOISO)
395)     2020 FORMAT(/ ' TIME=', 1PE13.6, ' YEARS',
396)         *   T40, 'RATIO OF INTEGRATED DISCHARGE TO EPA LIMIT' /
```

```
     *    5(1X, A6, 1PE14.6, 4X) )
      WRITE(6,2030) EPASUM
 2030 FORMAT(' SUM OF EPA RATIOS=' 1PE13.6)
      WRITE(6,2040) T, (ISONAME(J), EPAHEF(J), J=1,NOISO)
 2040 FORMAT(/ ' TIME=', 1PE13.6, ' YEARS',
     *    T40, 'EPA HEALTH EFFECTS' /
     *    5(1X, A6, 1PE14.6, 4X) )
      WRITE(6,2050) EPARSK
 2050 FORMAT(' SUM OF HEALTH EFFECTS=', 1PE13.6)
*DELETE        TPPRT.35,TPPRT.35
*INSERT      TRNSPRT.6
*CALL TIME
*INSERT      TRNSPRT.11
      TIME = T
*INSERT      TRNSPRT.98
      TIME = T
*I USEINP.51
      DTSAVE = DT
*I USEINP.74
      DT = DTSAVE
*INSERT         WORK.46
*COMDECK SANS
      LEVEL 2, SANS
      COMMON /SANS/
     *    SANS(1023,120), ZERO(121)
C-SANS-   SCRATCH ARRAY FOR SORTING DISCHARGE FROM DIFFERENT VECTORS
*DECK HEADER
      SUBROUTINE HEADER
C
C          PURPOSE -- WRITE TO TAPE8 THE FOLLOWING INFORMATION...
C                         ATTRIBUTE TYPE NAMES
C                         PROBLEM TITLE
C                         CODE TITLES
C                         DATE
C                         QUANTITY TITLES
C
```

```
*CALL PTCOM
C
*CALL MISC
C
      DIMENSION ANAMES(12)
      DIMENSION SDATE(2)
C
      DATA ANAMES /
     *'TITLE/CODE/DATE/PROBLEM/VECTOR/COORD1/COORD2/ISOTOPE $'
     * /
      DATA INIT /1/
C
C         DO THIS SUBROUTINE ONLY ONCE PER PLOT TAPE
      IF(INIT.NE.1) GO TO 900
      INIT = 0
C
C         WRITE ATTRIBUTE TYPE NAMES
      CALL TITLE(1,1,ANAMES)
C
C         WRITE DATE TO PLOT TAPE
      CALL DATE(SDATE)
      SDATE(2) = 1H$
      CALL TITLE(3,1,SDATE)
C
C         WRITE CODE TITLES TO PLOT TAPE
      CALL TITLE(2,1,'NWFT/DVM$')
C
C         WRITE COORDINATE TITLES TO PLOT TAPE
      CALL TITLE(6,1,'TIME (YR)$')
      CALL TITLE(6,2,'EPA RELEASE RATIO$')
      CALL TITLE(6,3,'EPA HEALTH EFFECTS$')
      CALL TITLE(6,4,'LEACH TIME (YR)$')
      CALL TITLE(7,1,'INTEGRATED DISCHARGE (CI)$')
      CALL TITLE(7,2,'DISCHARGE RATE (CI/DAY)$')
      CALL TITLE(7,3,'SOURCE RATE (CI/DAY)$')
      CALL TITLE(7,4,'EPA RELEASE RATIO$')
```

```
469)          CALL TITLE(7,5,'EPA HEALTH EFFECTS$')
470)          CALL TITLE(7,6,'PROBABILITY$')
471)          CALL TITLE(7,7,'TRAVEL TIME (YR)$')
472)          CALL TITLE(7,8,'CANISTER LIFE (YR)$')
473)     C
474)     C        WRITE PROBLEM TITLE TO PLOT TAPE
475)          CALL TITLE(4,1,TTL)
476)     C
477)       900 RETURN
478)          END
479)     *DECK TITLE
480)          SUBROUTINE TITLE(ATYP,VAL,STRING)
481)     C
482)     C        PURPOSE -- WRITE ATTRIBUTE TITLES TO PLOT TAPE
483)     C
484)     C        ATYP   = ATTRIBUTE TYPE
485)     C        VAL    = ATTRIBUTE VALUE WHOSE TITLE IS TO BE WRITTEN
486)     C        STRING = CHARACTER ARRAY CONTAINING TITLE
487)     C
488)          IMPLICIT INTEGER (A-Z)
489)     C
490)     *CALL PTCOM
491)     C
492)          DIMENSION STRING(8)
493)          DIMENSION CHARS(120)
494)     C
495)          DATA MAXATYP /8/
496)          DATA MAXAVAL /10*0/
497)     C
498)     C        FIND NUMBER OF CHARACTERS
499)        40 DECODE(120,1040,STRING) (CHARS(I),I=1,120)
500)      1040 FORMAT(120A1)
501)          DO 50 ILAST=1,120
502)          IF(CHARS(ILAST).EQ.1H$) GO TO 51
503)        50 CONTINUE
504)        51 CONTINUE
```

```
C
C          WRITE ATTRIBUTE VALUE TITLE TO PLOT TAPE
      AVAL(1) = ATYP
      AVAL(ATYP) = VAL
   60 WRITE(8,1060) (AVAL(I),I=1,MAXATYP)
 1060 FORMAT(10(I3,1X))
   70 WRITE(8,1070) (CHARS(I),I=1,ILAST)
 1070 FORMAT(120A1)
      AVAL(1) = 0
C
C          SET CURRENT ATTRIBUTE TO VALUE WHOSE TITLE WAS SET
C
      RETURN
      END
*DECK PTREC
      SUBROUTINE PTREC(X,Y)
C
C          PURPOSE -- WRITE A RECORD CONTAINING ATTRIBUTES AND NUMERICAL
C                     DATA TO PLOT TAPE.
C
C          X        = FIRST COORDINATE
C          Y        = SECOND COORDINATE
C
C          AVAL     = CURRENT ATTRIBUTE VALUES
C
*CALL PTCOM
C
C          WRITE ATTRIBUTES TO PLOT TAPE
      WRITE(8,1000) (AVAL(N),N=1,MAXATYP)
 1000 FORMAT(10(I3,1X))
C
C          WRITE NUMERICAL DATA TO PLOT TAPE
      WRITE(8,1010) X, Y
 1010 FORMAT(4(1PE20.10))
C
      RETURN
```

```
      END
*COMDECK PTCOM
      COMMON /PTCOM/
C                 -----
     *   AVAL(8), MAXATYP
C
      INTEGER AVAL
C
C -AVAL-    CURRENT ATTRIBUTE VALUES
C -MAXATYP- MAXIMUM ATTRIBUTE TYPES USED IN THE CODE
*COMDECK TIME
      COMMON /TIME/ TIME
C                 ----
C
C -TIME-    CURRENT PROBLEM TIME IN YEARS
C
*COMDECK EPA
      COMMON /EPA/
     *   EPALIM(120), EPAINT(120), EPASUM,
     *   EPATOX(120), EPAHEF(120), EPARSK,
     *   BMASS(120), REFMAS,
     *   NPT60, TRTIME,
     *EPAE
      LEVEL 2, SAMSUM
      COMMON /EPAL/
     *   SAMSUM(1023), SAMRSK(1023), VECNUM(1023),
     *   INPT60(1023),
     *EPALE
C
C-EPALIM(IS)- EPA RELEASE LIMIT OVER 1E4 YR PER 1000MTHM OF WASTE
C-EPAINT(IS)- RATIO OF INTEGRATED DISCHARGE TO EPALIM(IS)
C-EPASUM-     SUM OF EPA RATIOS
C-EPATOX(IS)- HEALTH EFFECTS PER CURIE OF IS RELEASED TO RIVER
C-EPAHEF(IS)- HEALTH EFFECTS FROM ISOTOPE IS
C-EPARSK-     HEALTH EFFECTS FROM ALL ISOTOPES
C-BMASS(IS)-  WASTE MASS FOR ASSIGNING INVENTORIES (MTHM)
```

```
C-REFMAS-      WASTE MASS FOR COMPUTING EPA FRACTIONS (MTHM)
C-SAMSUM(NORV)- EPA RELEASE RATIO FOR VECTOR NORV
C-SAMRSK(NORV)- EPA HEALTH EFFECTS FOR VECTOR NORV
C-VECNUM(NORV)- CONTAINS NORV (USED IN SORTING)
C-INPT60(NORV)- FLAG FOR WHETHER OR NOT A VECTOR IS WITHIN 10CFR60
C                    0 FOR NOT WITHIN 10FR60
C                    1 FOR WITHIN 10CFR60
C-NPT60-    TOTAL NUMBER OF VECTORS WITHIN PART60
C-TRTIME- TRAVEL TIME FROM EDGE OF REPOSITORY
*DECK EPAINI
      SUBROUTINE EPAINI
C
C PURPOSE -- INITIALIZE ARRAYS IN COMMON BLOCK /EPA/
C
*CALL EPA
*CALL ISOTOPE
*CALL SOLIMIT
C
C          INITIALIZE EPA LIMITS AND HEALTH EFFECTS
      DO 100 IR=1,NOISO
C
C          DEFAULT LIMIT IS 10 CI FOR ALPHA EMITTERS
C          500 CI FOR NON-ALPHA EMITTERS
      IF(AM(IR).LT.200.) EPALIM(IR) = 500.
      IF(AM(IR).GE.200.) EPALIM(IR) = 10.
C
C          THERE ARE SOME HEAVY NON-ALPHA EMITTERS
      NAME = ISONAME(IR)
      IF(NAME.EQ.'AC228 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'PB212 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'TL208 ') EPALIM(IR) = 500.
      IF(NAME.EQ.' U237 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'PA233 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'RA225 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'TL209 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'PB209 ') EPALIM(IR) = 500.
```

```
      IF(NAME.EQ.'AM242M') EPALIM(IR) = 10.
      IF(NAME.EQ.'AM242 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'TH234 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'PA234M') EPALIM(IR) = 500.
      IF(NAME.EQ.'PA234 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'PB214 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'BI214 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'PB210 ') EPALIM(IR) = 10.
      IF(NAME.EQ.'BI210 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'NP239 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'TH231 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'FR223 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'PB211 ') EPALIM(IR) = 500.
      IF(NAME.EQ.'TL207 ') EPALIM(IR) = 500.
C
C         OVERRIDE DEFAULT FOR CERTAIN NUCLIDES AS STATED
C         IN EPA STANDARD
      IF(NAME.EQ.'AM241 ') EPALIM(IR) = 10.
      IF(NAME.EQ.'AM243 ') EPALIM(IR) = 4.
      IF(NAME.EQ.' C 14 ') EPALIM(IR) = 200.
      IF(NAME.EQ.'CS135 ') EPALIM(IR) = 2000.
      IF(NAME.EQ.'CS137 ') EPALIM(IR) = 500.
      IF(NAME.EQ.' I129 ') EPALIM(IR) = 900.
      IF(NAME.EQ.'NP237 ') EPALIM(IR) = 20.
      IF(NAME.EQ.'PU238 ') EPALIM(IR) = 400.
      IF(NAME.EQ.'PU239 ') EPALIM(IR) = 100.
      IF(NAME.EQ.'PU240 ') EPALIM(IR) = 100.
      IF(NAME.EQ.'PU242 ') EPALIM(IR) = 100.
      IF(NAME.EQ.'RA226 ') EPALIM(IR) = 3.
      IF(NAME.EQ.'SR 90 ') EPALIM(IR) = 80.
      IF(NAME.EQ.'TC 99 ') EPALIM(IR) = 2000.
      IF(NAME.EQ.'SN126 ') EPALIM(IR) = 80.
C
C         DEFAULT TOXICITY FOR ALPHA AND NON-ALPHA EMITTERS
      IF(AM(IR).LT.200.) EPATOX(IR) = 0.0198
      IF(AM(IR).GE.200.) EPATOX(IR) = 0.719
```

```
C
C          THERE ARE SOME HEAVY NON-ALPHA EMITTERS
      IF(NAME.EQ.'AC228 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PB212 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'TL208 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.' U237 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PA233 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'RA225 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'TL209 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PB209 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'AM242M') EPATOX(IR) = 0.719
      IF(NAME.EQ.'AM242 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'TH234 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PA234M') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PA234 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PB214 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'BI214 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PB210 ') EPATOX(IR) = 0.719
      IF(NAME.EQ.'BI210 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'NP239 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'TH231 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'FR223 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'PB211 ') EPATOX(IR) = 0.0198
      IF(NAME.EQ.'TL207 ') EPATOX(IR) = 0.0198
C
C          OVERRIDE DEFAULTS FOR SPECIFIC     NUCLIDES LISTED IN TABLE D-1
C          OF EPA 520/5-80/002 (SMITH, FOWLER, AND GOLDIN).
C          RA226 IS TREATED LIKE AM241 VALUE DIVIDED BY 30% (RATIO OF
C          EPA RELEASE LIMITS).  DEFAULTS SET ABOVE COME FROM AM241
C          AND CS137, WHOSE EPA RELEASE LIMITS ARE EQUAL TO THE DEFAULT.
      IF(NAME.EQ.' C 14 ') EPATOX(IR) = 4.58E-2
      IF(NAME.EQ.'SR 90 ') EPATOX(IR) = 1.21E-1
      IF(NAME.EQ.'TC 99 ') EPATOX(IR) = 2.85E-4
      IF(NAME.EQ.'SN126 ') EPATOX(IR) = 1.20E-1
      IF(NAME.EQ.' I129 ') EPATOX(IR) = 1.08E-2
      IF(NAME.EQ.'CS135 ') EPATOX(IR) = 3.81E-3
```

```
685)          IF(NAME.EQ.'CS137 ') EPATOX(IR) = 1.98E-2
686)          IF(NAME.EQ.'NP237 ') EPATOX(IR) = 5.96E-1
687)          IF(NAME.EQ.'PU238 ') EPATOX(IR) = 2.29E-2
688)          IF(NAME.EQ.'PU239 ') EPATOX(IR) = 6.92E-2
689)          IF(NAME.EQ.'PU240 ') EPATOX(IR) = 6.53E-2
690)          IF(NAME.EQ.'AM241 ') EPATOX(IR) = 7.19E-1
691)          IF(NAME.EQ.'PU242 ') EPATOX(IR) = 6.76E-2
692)          IF(NAME.EQ.'AM243 ') EPATOX(IR) = 2.68E+0
693)      100 CONTINUE
694)    C
695)    C         PRINT EPA LIMIT ARRAY
696)          WRITE(6,1000) (ISONAME(J), EPALIM(J), J=1,NOISO)
697)     1000 FORMAT('1EPA RELEASE LIMITS OVER 1E4 YR PER 1000MTHM WASTE (CI)'/
698)         *   5(1X, A6, 1PE14.6, 4X) )
699)    C
700)    C         PRINT EPA TOXICITY ARRAY
701)          WRITE(6,1010) (ISONAME(J), EPATOX(J), J=1,NOISO)
702)     1010 FORMAT(/// ' EPA HEALTH EFFECTS PER CURIE RELEASED' /
703)         *   5(1X, A6, 1PE14.6, 4X) )
704)    C
705)          RETURN
706)          END
```

Appendix F. RATRAP Code Listing

```
SILLING,STMFZ,TP0,T77,P2.
ACCOUNT,SILLING,1349.
COMMENT.
COMMENT. ****************************************
COMMENT.
COMMENT. GENERATE NEW CYCLE OF RATRAP, BOTH CALCOMP
COMMENT. AND TEKTRONIX.
COMMENT.
COMMENT. ****************************************
COMMENT.
COMMENT. UPDATE AND COMPILE
COMMENT.
UPDATE,F,N,W.
RFL,137000.
FTN,I,L=0.
REDUCE.
RETURN,OLDPL.
COMMENT.
COMMENT. GENERATE ABSOLUTE FILE FOR CALCOMPS
COMMENT.
ATTACH,FILELIB,MR=1.
ATTACH,DISSPLA,MR=1.
ATTACH,BNLU,MR=1.
ATTACH,NRCPLTR,ID=ZZGNRC,MR=1.
ATTACH,NRCCLCP,ID=ZZGNRC,MR=1.
LIBRARY,NRCCLCP,NRCPLTR,FILELIB,DISSPLA,BNLU.
FILE,TAPE99,RT=W,BT=I.
LIBLOAD,DISSPLA,RESET.
LOAD,LGO.
NOGO,ABS.
COMMENT.
COMMENT. TRY RUNNING CALCOMP FILE.
COMMENT.
FILE,TAPE99,RT=W,BT=I.
ATTACH,TAPE8,SILLINGTAPER,ID=ZZRNRC,MR=1.
ABS.
```

```
RETURN,TAPE99.
COMMENT.
COMMENT. GENERATE ABSOLUTE FILE FOR TEKTRONIX
COMMENT.
LIBRARY,BNLU,DISSPLA,FILFLIB.
LOAD,LGO.
NOGO,ABSTX.
COMMENT.
COMMENT. TRY RUNNING TEKTRONIX FILE.
COMMENT.
REQUEST,TAPE99,*PF.
FILE,TAPE99,RT=W,BT=I.
ABSTX.
CATALOG,TAPE99,PLOTFILE,ID=SAS,ST=MFA.
REQUEST,A,*PF.
REWIND,NEWPL,LGO,ABS,ABSTX.
COPYP,NEWPL,A.
COPYP,LGO,A.
COPYP,ABS,A.
COPYP,ABSTX,A.
COMMENT.
COMMENT. SAVE THE PERMFILE
COMMENT.
EXIT,U.
PURGE,S,SILLINGRATRAP,CY=8,ID=ZZRNRC,PW=[redacted].
EXIT,U.
CATALOG,A,SILLINGRATRAP,CY=8,ID=ZZRNRC,RP=999,PW=[redacted].
*EOR
*COMDECK CC
      COMMON /CC/ ANAM(10,10), MAXAVAL(10)
     *   ,MAXATYP, LIMPNT, LIMAVAL, LIMATYP, LIMCOO
     *   ,ACOO(4), COOMIN(510,4), COOMAX(510,4), MAXPNT
     *   ,MAXCOO, LSCX, LSCY, LSCAT
      LEVEL 2, ATITLE
      COMMON /CCL/ ATITLE(8,510,10)
      INTEGER ANAM, ATITLE, ACOO
```

```
*DECK PLOTNUC
      PROGRAM PLOTNUC(INPUT,OUTPUT,TAPE8,TAPE99,TAPE98,
     *   TAPE5=INPUT,TAPE6=OUTPUT,TAPE9)
C
C     MAKE DISSPLA PLOTS FROM PLOT TAPE (TAPE8) GENERATED
C     BY RADIONUCLIDE TRANSPORT CODES
C
      IMPLICIT INTEGER (A-Z)
C
*CALL CC
C
      DIMENSION AREQ(10,12)
C
C         READ PLOT TAPE HEADER
      CALL ATINIT
C
C         FIND STRUCTURE OF PLOT FILE
      CALL SURVEY
C
C         FIND LIMITS OF DATA ON PLOT FILE
      CALL SCAN
C
C         REMOVE OUT-OF-RANGE DATA POINTS
      CALL EDIT
C
C         OPEN PLOT DEVICE
      CALL OPENDEV
C
C         LOOP OVER PLOTS
      DO 100 IPLT=1,100
C
C         GET PLOT REQUEST FROM INPUT FILE
      CALL GETREQ(AREQ,MAXOVR)
      IF(MAXOVR.EQ.0) GO TO 800
C
C         REOPEN PLOT FILE EVERY FOURTH PLOT
```

```
      IF(MOD(IPLT,4).EQ.0) CALL REOPDEV
C
C          DRAW OVERLAY SEMILOG PLOT
      CALL OVERLA(AREQ,MAXOVR,IPLT)
  100 CONTINUE
C
C          PLOTS FINISHED -- CLOSE DEVICE
  800 CALL CLOSDEV
      STOP
      END
*DECK OPENDEV
      SUBROUTINE OPENDEV
C
C          PURPOSE -- DOES OPENING AND CLOSING OF PLOT DEVICES
C
      IMPLICIT INTEGER(A-Z)
C
      DIMENSION PERSON(2)
C
C          READ DEVICE TYPE FROM INPUT FILES
      READ(5,1000) DEVICE
 1000 FORMAT(A10)
C
C          CHECK FOR VALID DEVICE TYPE
      IF(DEVICE.EQ.'CALCOMP'    .OR.
     *   DEVICE.EQ.'TEKTRONIX' ) GO TO 10
      WRITE(6,1005) DEVICE
 1005 FORMAT('1INVALID PLOT DEVICE ', A10)
      STOP
C
C          READ PROGRAMMER NAME FOR CALCOMP LABEL
   10 READ(5,1010) (PERSON(I),I=1,2)
 1010 FORMAT(2A10)
      WRITE(6,1015) DEVICE, (PERSON(I),I=1,2)
 1015 FORMAT('1DEVICE=       ', A10 /
     *       ' PROGRAMMER=  ', 2A10 /// )
```

```
145)      C
146)      C          BRANCH ACCORDING TO DEVICE TYPE
147)            IF(DEVICE.EQ.'CALCOMP') GO TO 100
148)            IF(DEVICE.EQ.'TEKTRONIX') GO TO 110
149)      C
150)      C          OPEN CALCOMP DEVICE
151)        100 CALL NAME(PERSON)
152)            CALL ORIGIN(0., -0.5, 0)
153)            CALL PLOTDVC('DEFAULT')
154)            CALL DISINIT
155)            CALL BGNPL(1)
156)            GO TO 190
157)      C
158)      C          OPEN TEKTRONIX DEVICE
159)        110 CALL PLOTDVC('DEFAULT')
160)            CALL DISINIT
161)            CALL BGNPL(1)
162)            CALL PAGE(9.5, 7.25)
163)            GO TO 190
164)      C
165)      C          END OF DEVICE OPENING ENTRY
166)        190 RETURN
167)      C
168)      C*****************************************************************
169)      C
170)      C          REOPEN PLOT DEVICE
171)            ENTRY REOPDEV
172)      C
173)      C          BRANCH ACCORDING TO DEVICE TYPE
174)            IF(DEVICE.EQ.'CALCOMP') GO TO 200
175)            IF(DEVICE.EQ.'TEKTRONIX') GO TO 210
176)      C
177)      C          REOPEN CALCOMP DEVICE
178)        200 CALL ENPLT(0., 0.)
179)            CALL NAME(PERSON)
180)            GO TO 290
```

```
C
C          REOPEN TEKTRONIX DEVICE
  210 GO TO 290
C
C          END OF DEVICE REOPENING ENTRY
  290 RETURN
C
C*********************************************************************
C
C          CLOSE PLOT DEVICE
      ENTRY CLOSDEV
C
C          BRANCH ACCORDING TO DEVICE TYPE
      IF(DEVICE.EQ.'CALCOMP') GO TO 300
      IF(DEVICE.EQ.'TEKTRONIX') GO TO 310
C
C          CLOSE CALCOMP DEVICE
  300 CALL DONEPL
      CALL ENPLT(3., 0.)
      GO TO 390
C
C          CLOSE TEKTRONIX DEVICE
  310 CALL DONEPL
      GO TO 390
C
C          END OF DEVICE CLOSING ENTRY
  390 RETURN
C
C*********************************************************************
C
      END
*DECK ATINIT
      SUBROUTINE ATINIT
C
C     PURPOSE -- READ ATTRIBUTE TYPE NAMES FROM PLOT TAPE HEADER.
C
```

```
C     HEADER HAS THE FOLLOWING FORM...
C          TITLE/(NAME2)/(NAME3)/(NAME4)/.../(NAMEN)
C     WHERE NAMEI IS THE NAME OF ATTRIBUTE TYPE I.
C     NAMES ARE USED TO IDENTIFY THE FORMAT OF PLOT RECORDS TO
C     FOLLOW, WHICH MAY VARY BETWEEN CODE VERSIONS AND RUNS.
C     FOR EXAMPLE, NAME2 MIGHT BE 'PRESSURE' ON ONE PLOT TAPE
C     AND 'DISCHARGE' ON ANOTHER.
C
C     THE FIRST ATTRIBUTE TYPE MUST HAVE THE NAME 'TITLE'.
C     THIS ATTRIBUTE WILL BE NONZERO IN SUBSEQUENT RECORDS WHEN
C     THE TITLE OF SOME ATTRIBUTE VALUE IS TO BE READ.
C
C     ATTRIBUTE TYPE NAMES MUST HAVE NO MORE THAN 10 CHARACTERS
C     AND NOT CONTAIN ANY BLANKS.
C
      IMPLICIT INTEGER (A-Z)
C
*CALL CC
C
      DIMENSION IN(120)
C
      DATA ANAM /100*1H /
      DATA ATITLE /40800*1H /
      DATA MAXATYP /10/
      DATA MAXAVAL /10*0/
      DATA LIMPNT, LIMAVAL, LIMATYP, LIMCOO /3100, 510, 10, 4/
      DATA INIT /1/
C
C         PERFORM THIS ROUTINE ONLY ONCE PER PLOT TAPE
      IF(INIT.NE.1) GO TO 900
      INIT = 0
C
C         READ PLOT TAPE HEADER
      REWIND 8
      READ(8,1000) JUNK
      READ(8,1000) (IN(COL),COL=1,120)
```

```
 1000 FORMAT(120A1)
C
C          CHECK FOR EMPTY FILE
      IF(EOF(8).EQ.0) GO TO 10
      WRITE(6,1005)
 1005 FORMAT('1PLOT TAPE EMPTY -- ATINIT')
      GO TO 900
C
C          ASSIGN NAMES TO ATTRIBUTE TYPES
   10 COLNAM = 1
      ATYP = 1
      DO 20 COLIN=1,120
      IF(IN(COLIN).EQ.1H ) GO TO 21
      IF(IN(COLIN).EQ.1H/) GO TO 15
C
C          STORE A CHARACTER IN ATTRIBUTE TYPE NAME ARRAY
      ANAM(COLNAM,ATYP) = IN(COLIN)
      COLNAM = COLNAM + 1
      GO TO 20
C
C          SLASH MEANS BEGIN NEW ATTRIBUTE TYPE
   15 ATYP = ATYP + 1
      COLNAM = 1
   20 CONTINUE
   21 MAXATYP = ATYP
      REWIND 8
C
  900 RETURN
      END
*DECK SURVEY
      SUBROUTINE SURVEY
C
C          PURPOSE -- LOOK AT PLOT TAPE AND GET STATISTICS AND LIMITS
C
C          THIS ROUTINE ASSUMES THAT THE PLOT TAPE HEADER HAS BEEN
C          READ BY ATINIT.
```

```
C
      IMPLICIT INTEGER (A-Z)
C
*CALL CC
C
      DIMENSION TAGLIST(10,101), NTAG(101), A(10)
C
      DATA LIMTAG /101/
C
C
C         BEGIN LOOP OVER PLOT RECORDS
      REC = 0
   10 READ(8,1010) (A(ATYP),ATYP=1,MAXATYP)
 1010 FORMAT(10(I3,1X))
      IF(EOF(8).NE.0) GO TO 100
      REC = REC + 1
C
C         IF TITLE ATTRIBUTE IS NONZERO A TITLE RECORD FOLLOWS
      IF(A(1).EQ.0) GO TO 30
C
C         READ TITLE FOR VALUE NUMBER A(A(1)) OF ATTRIBUTE TYPE A(1)
      ATYP = A(1)
      VAL = A(ATYP)
   20 READ(8,1020) (ATITLE(N,VAL,ATYP),N=1,8)
 1020 FORMAT(8A10)
      GO TO 10
C
C         IF TITLE ATTRIBUTE IS 0 NUMERICAL DATA FOLLOWS.  INCREMENT
C          ATTRIBUTE COUNTERS, THEN SKIP NEXT RECORD.
   30 DO 50 ATYP=1,MAXATYP
      MAXAVAL(ATYP) = MAX0(MAXAVAL(ATYP),A(ATYP))
   50 CONTINUE
  160 READ(8,1160) JUNK
 1160 FORMAT(A10)
      GO TO 10
C
```

```
C     END OF FILE REACHED -- PRINT NUMBER OF RECORDS
  100 WRITE(6,1100) REC
 1100 FORMAT('1SURVEY COMPLETE   ', I6, ' RECORDS READ' // )
C
C         PRINT ATTRIBUTE TITLES
      DO 200 ATYP=1,MAXATYP
      IF(MAXAVAL(ATYP).EQ.0) GO TO 200
  110 WRITE(6,1110) ATYP, (ANAM(COL,ATYP),COL=1,10)
 1110 FORMAT('0ATTRIBUTE TYPE ', I3, ' HAS NAME ', 10A1)
      M = MAXAVAL(ATYP)
      DO 150 VAL=1,M
  140 WRITE(6,1140) VAL, (ATITLE(N,VAL,ATYP),N=1,8)
 1140 FORMAT(5X, 'VALUE ', I3, '  HAS TITLE ', 8A10)
  150 CONTINUE
  200 CONTINUE
C
C         FIND NUMBER OF OCCURRENCES OF EACH SET OF ATTRIBUTES
C
C
C         PRINT HEADER
      WRITE(6,1150)
 1150 FORMAT('1NAME TAGS FOUND FOR NUMERICAL DATA ON PLOT TAPE' ///
     *   '  RECORDS', T20, 'ATTRIBUTE LIST' // )
C
C         INITIALIZE TAGLIST AND NTAG
      DO 312 ITAG=1,LIMTAG
      NTAG(ITAG) = 0
      DO 311 ATYP=1,MAXATYP
      TAGLIST(ATYP,ITAG) = 0
  311 CONTINUE
  312 CONTINUE
C
C         LOOP OVER RECORDS ON PLOT TAPE
      REWIND 8
      MAXTAG = 0
      DO 350 IREC=1,50000
```

```
361)      C
362)      C          READ NEXT TAG
363)            READ(8,1010) (A(ATYP),ATYP=1,MAXATYP)
364)            IF(EOF(8).NE.0) GO TO 360
365)            IF(A(1).NE.0) GO TO 340
366)      C
367)      C          COMPARE TAG WITH THOSE ON LIST
368)            IF(MAXTAG.EQ.0) GO TO 330
369)            DO 325 ITAG=1,MAXTAG
370)            DO 320 ATYP=1,MAXATYP
371)            IF(A(ATYP).NE.TAGLIST(ATYP,ITAG)) GO TO 325
372)        320 CONTINUE
373)      C
374)      C          TAG AGREES WITH ITAG'TH ON LIST
375)            NTAG(ITAG) = NTAG(ITAG) + 1
376)            GO TO 340
377)      C
378)      C          END OF LOOP OVER TAG LIST
379)        325 CONTINUE
380)      C
381)      C          TAG DOES NOT AGREE WITH ANY ON LIST.
382)      C          CHECK NUMBER OF TAGS AGAINST LIMIT.
383)        330 IF(MAXTAG.LT.LIMTAG) GO TO 332
384)            WRITE(6,1170) LIMTAG
385)       1170 FORMAT(/// ' TOO MANY DIFFERENT TAGS TO COUNT THEM ALL.' /
386)           *            ' LIMIT IS ', I7 /
387)           *            ' TO INCREASE, CHANGE LIMTAG AND DIMENSIONS OF' /
388)           *            ' NTAG() AND TAGLIST(,) IN SUBROUTINE SURVEY.' /
389)           *            ' SUMMARY OF TAGS READ SO FAR FOLLOWS...' // )
390)            GO TO 365
391)      C
392)      C          NUMBER OF TAGS IS WITHIN LIMIT
393)        332 MAXTAG = MAXTAG + 1
394)            DO 335 ATYP=1,MAXATYP
395)            TAGLIST(ATYP,MAXTAG) = A(ATYP)
396)        335 CONTINUE
```

```
      NTAG(MAXTAG) = 1
C
C          SKIP NUMERICAL DATA OR TITLE
  340 READ(8,1160) JUNK
C
C          END OF LOOP OVER TAPE RECORDS
  350 CONTINUE
C
C          PRINT SUMMARY
  360 IF(MAXTAG.GT.0) GO TO 365
      WRITE(6,1180)
 1180 FORMAT(// ' NO NUMERICAL RECORDS FOUND' //)
      GO TO 399
  365 DO 370 ITAG=1,MAXTAG
      WRITE(6,1190) NTAG(ITAG), (TAGLIST(ATYP,ITAG),ATYP=1,MAXATYP)
 1190 FORMAT(' ', I7, T20, 30(I3.3, 1X))
  370 CONTINUE
  399 CONTINUE
C
      RETURN
      END
*DECK SCAN
      SUBROUTINE SCAN
C
C          PURPOSE -- GET COORDINATE LIMITS FROM PLOT TAPE
C
*CALL CC
C
C
      DIMENSION SCRATCH(4), AREC(10)
C
      INTEGER AREC
C
C          FIND WHICH ATTRIBUTE TYPES REFER TO COORDINATES
      MAXCOO = 0
   30 WRITE(6,1030)
```

```
 1030 FORMAT('1COORDINATE ATTRIBUTE FIELDS' /)
      DO 50 IA=1,MAXATYP
      IF(ANAM(1,IA).NE.1HC.OR.
     *   ANAM(2,IA).NE.1HO.OR.
     *   ANAM(3,IA).NE.1HO.OR.
     *   ANAM(4,IA).NE.1HR.OR.
     *   ANAM(5,IA).NE.1HD) GO TO 50
C
C          ATTRIBUTE TYPE IA REFERS TO A COORDINATE
C          FIND THE COORDINATE NUMBER
      DECODE(1,1000,ANAM(6,IA)) JCOO
 1000 FORMAT(I1)
      ACOO(JCOO) = IA
  140 WRITE(6,1040) IA, JCOO
 1040 FORMAT(' ATTRIBUTE TYPE', I3,
     *   ' REFERS TO COORDINATE FIELD', I3)
C
C          COMPARE AGAINST CURRENT COORDINATE COUNT
      MAXCOO = MAX0(MAXCOO,JCOO)
   50 CONTINUE
C
C          INITIALIZE COORDINATE LIMITS
      DO 90 ICOO=1,LIMCOO
      DO 80 IVAL=1,LIMAVAL
      COOMAX(IVAL,ICOO) = -1.E90
      COOMIN(IVAL,ICOO) = 1.E90
   80 CONTINUE
   90 CONTINUE
C
C          LOOP OVER PLOT RECORDS
      REWIND 8
      DO 200 IREC=1,70000
C
C          READ ATTRIBUTE LIST
  100 READ(8,1100) (AREC(I),I=1,MAXATYP)
 1100 FORMAT(10(I3, 1X))
```

```
      IF(EOF(8).NE.0) GO TO 210
C
C          IF TITLE ATTRIBUTE IS NONZERO SKIP NEXT RECORD
      IF(AREC(1).EQ.0) GO TO 110
  105 READ(8,1105) JUNK
 1105 FORMAT(A1)
      GO TO 200
C
C          READ NUMERICAL DATA
  110 READ(8,1110) (SCRATCH(I),I=1,MAXCOO)
 1110 FORMAT(4(1PE20.10))
C
C          LOOP OVER COORDINATE FIELDS
      DO 120 ICOO=1,MAXCOO
      IVAL = AREC(ACOO(ICOO))
      COOMAX(IVAL,ICOO) = AMAX1(COOMAX(IVAL,ICOO),SCRATCH(ICOO))
      COOMIN(IVAL,ICOO) = AMIN1(COOMIN(IVAL,ICOO),SCRATCH(ICOO))
  120 CONTINUE
  200 CONTINUE
C
C          SCAN COMPLETE -- PRINT LIMITS
  210 WRITE(6,1210)
 1210 FORMAT(///'1SCAN COMPLETE' /// ' FIELD', T10, 'QUANTITY', T40,
     *   'MIN', T60, 'MAX' / )
      DO 330 ICOO=1,MAXCOO
      M = MAXAVAL(ACOO(ICOO))
      DO 320 IVAL=1,M
  310 WRITE(6,1310) ICOO, IVAL, COOMIN(IVAL,ICOO),
     *   COOMAX(IVAL,ICOO)
 1310 FORMAT(1H , I3, T14, I3, T34, 1PE15.6, T54, 1PE15.6)
  320 CONTINUE
  330 CONTINUE
C
      RETURN
      END
*DECK EDIT
```

```
      SUBROUTINE EDIT
C
C PURPOSE -- TRANSFER NUMERICAL DATA TO UNFORMATTED FILE (TAPE9)
C
*CALL CC
C
      DIMENSION AREC(10), BUF(10)
C
C         LOOP OVER PLOT RECORDS
      REWIND 8
      REWIND 9
      LINES9 = 0
      DO 100 IREC=1,50000
C
C         READ ATTRIBUTE LIST
      READ(8,1000) (AREC(I),I=1,MAXATYP)
 1000 FORMAT(10(I3, 1X))
      IF(EOF(8).NE.0) GO TO 110
C
C         DROP TITLE RECORDS
      IF(AREC(1).EQ.0) GO TO 30
      READ(8,1010) JUNK
 1010 FORMAT(A1)
      GO TO 100
C
C         READ NUMERICAL DATA INTO BUFFER
   30 READ(8,1020) (BUF(I),I=1,MAXCOO)
 1020 FORMAT(4(1PE20.10))
C
C         COPY THE RECORD ONTO UNFORMATTED PLOT TAPE
      WRITE(9) (AREC(I),I=1,MAXATYP)
      WRITE(9) (BUF(I),I=1,MAXCOO)
C
C         INCREMENT COUNTER
      LINES9 = LINES9 + 1
  100 CONTINUE
```

```
  110 CONTINUE
C
C          PRINT NUMBER OF RECORDS
      WRITE(6,1100) IREC-1, LINES9
 1100 FORMAT('1', 'EDIT COMPLETE' //
     *        ' ', I7, ' RECORDS ON OLD FILE' /
     *        ' ', I7, ' RECORDS ON NEW FILE' )
C
      RETURN
      END
*DECK GETSEQ
      SUBROUTINE GETSEQ(AVAL,DLOAD)
C
C          PURPOSE -- READ NUMERICAL DATA RECORDS WHOSE ATTRIBUTES
C                     ARE EQUAL TO THE CURRENT ATTRIBUTES
C
C          DLOAD  = ARRAY INTO WHICH THE COORDINATE FIELDS WILL
C                   BE LOADED
C
C          AVAL   = CURRENT ATTRIBUTES INDEXED BY ATTRIBUTE TYPE
C
*CALL CC
C
      DIMENSION DLOAD(3100,4)
      DIMENSION AREC(10), AVAL(10)
C
      INTEGER AREC, AVAL
C
C          CLEAR DLOAD ARRAY
      DO 50 IPNT=1,LIMPNT
      DO 40 ICOO=1,LIMCOO
      DLOAD(IPNT,ICOO) = 0.
   40 CONTINUE
   50 CONTINUE
C
C          BEGIN SEARCH FOR PLOT RECORDS WITH CURRENT ATTRIBUTES
```

```
      REWIND 9
      IPNT = 0
      DO 90 IREC=1,70000
C
C          READ AN ATTRIBUTE LIST AND COMPARE
   60 READ(9) (AREC(I),I=1,MAXATYP)
      IF(EOF(9).NE.0) GO TO 100
      DO 70 I=1,MAXATYP
      IF(AREC(I).NE.AVAL(I)) GO TO 80
   70 CONTINUE
C
C          ATTRIBUTES AGREE -- STORE DATA ON NEXT CARD
      IPNT = IPNT + 1
   75 READ(9) (DLOAD(IPNT,I),I=1,MAXCOO)
      GO TO 90
C
C          ATTRIBUTES DISAGREE -- SKIP NEXT CARD
   80 READ(9) JUNK
 1080 FORMAT(A1)
   90 CONTINUE
C
C          LOAD COMPLETE
  100 MAXPNT = IPNT
  110 WRITE(6,1110) (AVAL(I),I=1,MAXATYP)
 1110 FORMAT('0SEQUENCE LOADED WITH ATTRIBUTES  ', 10(I3, 1X))
  120 WRITE(6,1120) MAXPNT
 1120 FORMAT(1H , I7, ' POINTS LOADED')
C
      RETURN
      END
*DECK GETREQ
      SUBROUTINE GETREQ(AREQ,MAXOVR)
C
C          PURPOSE -- READ PLOT REQUEST FROM INPUT FILE
C
C          AREQ(ITYP,IOVR) = ARRAY OF REQUESTED ATTRIBUTES FOR PLOT
```

```
C          MAXOVR = NUMBER OF OVERLAYS FOR THIS PLOT. 0 MEANS STOP.
C
*CALL CC
C
      DIMENSION AREQ(10,12), CARD(9)
C
      INTEGER AREQ
C
C          CLEAR AREQ ARRAY
      DO 40 IVAL=1,12
      DO 30 ITYP=1,LIMATYP
   30 AREQ(ITYP,IVAL) = 0
   40 CONTINUE
C
C          DEFAULT PLOT TYPE IS LINEAR X, LOG Y
      LSCX = 0
      LSCY = 1
C
C          DEFAULT CURVE TYPE IS CONNECTED
      LSCAT = 0
C
C          READ PLOT REQUESTS
      MAXOVR = 0
      M = 12 + 1
      DO 100 IOVR=1,M
      READ(5,1000) (CARD(I),I=1,9)
 1000 FORMAT(9A10)
C
C          CHECK FOR OPTIONS
      IF(INLINE(CARD,'LOGX$').EQ.1) LSCX = 1
      IF(INLINE(CARD,'LINY$').EQ.1) LSCY = 0
      IF(INLINE(CARD,'SCAT$').EQ.1) LSCAT = 1
      DECODE(90,1050,CARD) (AREQ(ITYP,IOVR),ITYP=1,MAXATYP)
C     WRITE(6,1050) (AREQ(ITYP,IOVR),ITYP=1,MAXATYP)
 1050 FORMAT(10(I3, 1X))
      IF(EOF(5).NE.0) GO TO 900
```

```
      IF(AREQ(1,IOVR).LT.0) GO TO 900
C
C          CHECK FOR END-OF-PLOT CARD (BLANK)
      DO 70 ITYP=1,MAXATYP
      IF(AREQ(ITYP,IOVR).NE.0) GO TO 100
   70 CONTINUE
      MAXOVR = IOVR - 1
      GO TO 200
  100 CONTINUE
  200 CONTINUE
  900 RETURN
      END
*DECK INLINE
      FUNCTION INLINE(CARD, WORD)
C
C PURPOSE -- SEE IF WORD IS CONTAINED IN CARD
C
C          WORD MUST BE TERMINATED BY A $
C          INLINE = 0 IF NOT IN LINE
C                   1 IF IN LINE
C
      IMPLICIT INTEGER (A-Z)
C
      DIMENSION CARD(9)
      DIMENSION C(90), W(10)
C
C          C IS THE EXPANSION OF CARD
C          W IS THE EXPANSION OF WORD
      DECODE(90,1000,CARD) (C(I),I=1,90)
 1000 FORMAT(90A1)
      DECODE(10,1000,WORD) (W(I),I=1,10)
C
C
C          SEARCH FOR WORD IN CARD
      INLINE = 0
      DO 100 I=1,90
```

```
      DO 70 J=1,10
      IF(W(J).EQ.1H$) GO TO 110
      IF(I+J-1.GT.90) GO TO 900
      IF(W(J).NE.C(I+J-1)) GO TO 100
   70 CONTINUE
C
C          STRING FOUND
      GO TO 110
  100 CONTINUE
C
C          STRING NOT FOUND
      GO TO 900
C
C          SEARCH SUCCESSFUL
  110 INLINE = 1
C
  900 RETURN
      END
      END
*DECK OVERLA
      SUBROUTINE OVERLA(AREQ,MAXOVR,IPLT)
C
C          PURPOSE -- PRODUCE PLOTS
C
C          AREQ(ITYP,IOVR) = ARRAY OF PLOT REQUESTS
C          MAXOVR = NUMBER OF CURVES TO OVERLAY ON SAME PLOT
C          IPLT = PLOT NUMBER
C
*CALL CC
C
      DIMENSION AREQ(10,12), SCR1(400), SCR2(400), X(3100,4)
      DIMENSION LTEMP(8)
      DIMENSION SCR4(8), SCR5(8)
C
      INTEGER AREQ, SCR1, SCR2, SCR3, AOVR
      INTEGER SCR4, SCR5
```

```
C
C          FIND WHICH ATTRIBUTE IS THE OVERLAY ATTRIBUTE (IF ANY)
      AOVR = 0
      NAOVR = 0
      IF(MAXOVR.EQ.1) GO TO 40
      DO 20 ITYP=1,MAXATYP
      DO 10 IOVR=1,MAXOVR
      IF(AREQ(ITYP,IOVR).EQ.AREQ(ITYP,1)) GO TO 10
      NAOVR = NAOVR + 1
      AOVR = ITYP
      GO TO 20
   10 CONTINUE
   20 CONTINUE
C
C          THERE MUST NOT BE MORE THAN 1 OVERLAY ATTRIBUTE
      IF(NAOVR.LE.1) GO TO 40
   30 WRITE(6,1030) NAOVR
 1030 FORMAT(1H0, I7, ' OVERLAY ATTRIBUTES REQUESTED.',
     *   '  LIMIT IS 1.  PLOT ABANDONED -- OVERLA')
      GO TO 900
C
C          CANNOT PLOT DIFFERENT PHYSICAL QUANTITIES ON SAME AXIS
   40 DO 50 ICOO=1,MAXCOO
      IF(ACOO(ICOO).EQ.AOVR) GO TO 60
   50 CONTINUE
      GO TO 70
   60 WRITE(6,1060) AOVR
 1060 FORMAT('0OVERLAY ATTRIBUTE', I3,
     *   ' IS A COORDINATE.  PLOT ABANDONED -- OVERLA')
      GO TO 900
C
C          DETERMINE COORDINATE TYPES
   70 ACOOX = ACOO(1)
      ACOOY = ACOO(2)
      JCOOX = AREQ(ACOOX,1)
      JCOOY = AREQ(ACOOY,1)
```

```
C
C          FIND X-AXIS SCALE
      GO TO (82,85), LSCX + 1
C
C          X-AXIS IS LINEAR
   82 CALL AXSPLT(COOMIN(JCOOX,1), COOMAX(JCOOX,1), 5.0,
     *   XLO, XSTEP, XAXIS)
      GO TO 90
C
C          X-AXIS IS LOGARITHMIC
   85 XMAX = AMAX1(COOMAX(JCOOX,1), 1.E-90)
      XHI = 10. ** IFIX(ALOG10(XMAX))
      IF(XHI.LT.XMAX) XHI = XHI * 10.
      XLO = XHI / 1.E5
      XAXIS = 5.0
C
C          FIND Y-AXIS SCALE
   90 GO TO (92,95), LSCY + 1
C
C          Y-AXIS IS LINEAR
   92 CALL AXSPLT(COOMIN(JCOOY,2), COOMAX(JCOOY,2), 5.0,
     *   YLO, YSTEP, YAXIS)
      GO TO 100
C
C          Y-AXIS IS LOGARITHMIC
   95 YMAX = AMAX1(COOMAX(JCOOY,2), 1.E-90)
      YHI = 10. ** IFIX(ALOG10(YMAX))
      IF(YHI.LT.YMAX) YHI = YHI * 10.
      YLO = YHI / 1.E5
      YAXIS = 5.0
  100 CONTINUE
C
C          SET COORDINATE TITLES
      DO 105 IW=1,8
      SCR4(IW) = ATITLE(IW,JCOOX,ACOOX)
      SCR5(IW) = ATITLE(IW,JCOOY,ACOOY)
```

```
  105 CONTINUE
      CALL TITLE(1H ,1,SCR4,100,
     *   SCR5,100,XAXIS,YAXIS)
C
C          SET TITLES FOR NON-OVERLAY ATTRIBUTES
      LINE = 0
      DO 150 ITYP=2,MAXATYP
      IF(ITYP.EQ.AOVR.OR.ITYP.EQ.ACOOX.OR.ITYP.EQ.ACOOY)
     *   GO TO 150
      LINE = LINE + 1
      DO 145 LW=1,8
      LTEMP(LW) = ATITLE(LW,AREQ(ITYP,1),ITYP)
  145 CONTINUE
      CALL LINES(LTEMP, SCR1, LINE)
  150 CONTINUE
      CALL LSTORY(SCR1,LINE,0.,5.25)
C
C          LOAD THE DATA
  160 WRITE(6,1160) LSCX, LSCY, XLO, XSTEP, XHI, YLO, YSTEP, YHI
 1160 FORMAT('0LSCX=      ', I13,      5X, 'LSCY=      ', I13 /
     *        ' XLO=      ', 1PE13.6, 5X, 'XSTEP=     ', 1PE13.6 /
     *        ' XHI=      ', 1PE13.6, 5X, 'YLO=       ', 1PE13.6 /
     *        ' YSTEP=    ', 1PE13.6, 5X, 'YHI=       ', 1PE13.6)
C
C          SET PLOT TYPE
      IF(LSCX.EQ.0.AND.LSCY.EQ.0) CALL GRAPH(XLO, XSTEP, YLO, YSTEP)
      IF(LSCX.EQ.1.AND.LSCY.EQ.0) CALL XLOG(XLO, 1.0, YLO, YSTEP)
      IF(LSCX.EQ.0.AND.LSCY.EQ.1) CALL YLOG(XLO, XSTEP, YLO, 1.0)
      IF(LSCX.EQ.1.AND.LSCY.EQ.1) CALL LOGLOG(XLO, 1.0, YLO, 1.0)
C     CALL NOCHEK
      DO 170 IOVR=1,MAXOVR
      CALL GETSEQ(AREQ(1,IOVR),X)
C
C          IF NO POINTS WERE LOADED, CREATE A DUMMY POINT
      IF(MAXPNT.GT.0) GO TO 164
      MAXPNT = 1
```

```
      X(1,1) = 0.
      X(1,2) = 0.
C
C          APPLY LOWER LIMIT TO Y VALUES TO AVOID ERROR MESSAGES
  164 DO 165 IPNT=1,MAXPNT
      X(IPNT,2) = AMAX1(X(IPNT,2), 1.E-90)
  165 CONTINUE
C
C          DRAW THE GRAPH
      IMARK = MAX0(1, MAXPNT/3)
      IF(LSCAT.EQ.1) IMARK = -1
      CALL CURVE(X(1,1),X(1,2),MAXPNT,IMARK)
C
C          SKIP THE CIRCLE SYMBOL BECAUSE IT LOOKS TOO MUCH LIKE A SQUARE
      DUM1 = 1.E-90
      DUM2 = 1.E-90
      IF(IOVR.EQ.1) CALL CURVE(DUM1, DUM2, 1, 0)
  170 CONTINUE
C
C          SET LEGEND FOR OVERLAY ATTRIBUTES
      IF(NAOVR.EQ.0) GO TO 200
      IF(MAXOVR.GT.5) CALL HEIGHT(0.07)
  180 ENCODE(10,1180,SCR2) (ANAM(I,AOVR),I=1,9)
 1180 FORMAT(9A1, 1H$)
      CALL MYLEGN(SCR2,100)
      DO 190 IOVR=1,MAXOVR
      DO 185 LW=1,8
      LTEMP(LW) = ATITLE(LW,AREQ(AOVR,IOVR),AOVR)
  185 CONTINUE
      CALL LINES(LTEMP, SCR2, IOVR)
  190 CONTINUE
      CALL RESET('HEIGHT')
      CALL LEGEND(SCR2,IOVR,3.,5.25)
C
C          DRAW GRID
  200 IF(LSCX.EQ.0.AND.LSCY.EQ.0) CALL GRID(1,1)
```

```
      IF(LSCX.EQ.1.AND.LSCY.EQ.0) CALL GRID(-9,1)
      IF(LSCX.EQ.0.AND.LSCY.EQ.1) CALL GRID(1,-9)
      IF(LSCX.EQ.1.AND.LSCY.EQ.1) CALL GRID(-9,-9)
C
C          END OF PLOT
      CALL ENDPL(-1)
C
  900 RETURN
      END
*EOR
CALCOMP
S.SILLING X74173

-1
*EOR
TEKTRONIX
S.SILLING X74173
000 001 001 001 001 001 009 001 LOGX
000 001 001 001 001 001 009 009

-1
```