SKBF TEKNISK
KBS RAPPORT

80-03

GETOUT – a one-dimensional model for groundwater transport of radionuclide decay chains

Bertil Grindfelt
Mark Elert

Kemakta konsult AB, January 1980

SVENSK KÄRNBRÄNSLEFÖRSÖRJNING AB / PROJEKT KÄRNBRÄNSLESÄKERHET

POSTADRESS: Kärnbränslesäkerhet, Box 5864, 102 48 Stockholm, Telefon 08-67 95 40

GETOUT - A ONE-DIMENSIONAL MODEL FOR GROUNDWATER TRANSPORT OF RADIONUCLIDE DECAY CHAINS

Bertil Grundfelt
Mark Elert

Kemakta konsult AB, January 1980

A list of other reports published in this series is attached at the end of this report. Information on KBS technical reports from 1977-1978 (TR 121) and 1979 (TR 79-28) is available through SKBF/KBS.

FÖRTECKNING ÖVER KBS TEKNISKA RAPPORTER

1977-78

TR 121 KBS Technical Reports 1 - 120.
Summaries. Stockholm, May 1979.

1979

TR 79-28 The KBS Annual Report 1979.
KBS Technical Reports 79-01--79-27.
Summaries. Stockholm, March 1980.

1980

TR 80-01 Kompletterande geohydrologiska undersökningar inom karlshamnsområdet
Lennart Ekman
Bengt Gentzschein
Sveriges geologiska undersökning, januari 1980

TR 80-02 Modelling of rock mass deformation for radioactive waste repositories in hard rock
Ove Stephansson
Per Jonasson
Department of Rock Mechanics
University of Luleå, Luleå

Tommy Groth
Department of Soil and Rock Mechanics
Royal Institute of Technology, Stockholm
1980-01-29

TR 80-03 GETOUT - a one-dimensional model for groundwater transport of radionuclide decay chains
Bertil Grundfelt
Mark Elert
Kemakta konsult AB, januari 1980

KEMAKTA konsult AB
Bertil Grundfelt
Mark Elert

Summary

The GETOUT-code, originally developed at Batelle Pacific Northwest Laboratories (PNL), was used in the KBS-project to calculate the radionuclide discharges from the repository. The version used in KBS was a translation of the PNL BASIC-language version as by december 1976. In this report a new version , mathematically compatible whith the PNL FORTRAN version as by 1979-08-15, is documented. Details are given on the differences between this new version and the version used in the KBS project up to now.

Table of contents

1. History and product description

In the safety analysis in the KBS project /1, 2/, a one-dimensional dispersion model with chain decay and geo-chemical retardation called GETOUT has been used to calculate the nuclide inflows to the biosphere from the final repository. The model was originally developed at Batelle Pacific Northwest Laboratory (PNL) /3/. When KBS was started in December of 1976, a BASIC version of the computer program then used by PNL for inflow calculations was obtained. Owing to numerical problems, the dispersion was omitted in this version for those nuclides that are formed by means of chain decay during transport through the bedrock /4/. The BASIC program was translated within KBS to FORTRAN, whereby the logic was revised to simplify parameter studies and reduce computer run costs.

In this project, the GETOUT version used in the KBS safety analysis has been updated to coincide with a later version from PNL /5/. In connection with this updating, the program has been "trimmed" for more efficient use and running:

- Options for calculating triple-nuclide chains with the same retention factors have been put in.
- A routine package for calculating the source strengths at the time of canister breakthrough have been included.
- The calculations of the time scales that are used have been revised.

- The part of the program that calculates nuclide contributions where special approximations must be resorted to, has been made more flexible.
- The printouts have been made more user-oriented.
- A separate program for plotting results has been developed.

This report describes the above-mentioned measures. The GETOUT logic is described in Appendix 1. Appendix 2 is a listing of GETOUT and Appendix 3 a description of required input data. Examples of the printouts obtained from the program are shown in Appendix 4.

The report describes the measures related to programming that have been adopted within the project. Readers who are not familiar with GETOUT and the manner in which GETOUT works are advised to read Appendix 1 first.

2. Updating of GETOUT with reference to PNL's new version

GETOUT is based on the analytical solution of a system of partial differential equations of the form:

$$K_i \frac{\partial Ni}{\partial t} = D \frac{\partial^2 Ni}{\partial x^2} - u \frac{\partial Ni}{\partial x} - K_i \lambda_i N_i + K_{i-1} \lambda_{i-1} N_{i-1} \qquad (1)$$

where: N_i = the flow of nuclide (mole/s) through a cross-section at point x (m) at time t (s),

K_i = the retention factor for nuclide i (groundwater velocity/nuclide velocity),

D = the dispersion coefficient (m^2/s)

u = the groundwater velocity (m/s),

λ_i = the decay constant for nuclide i (s^{-1}).

The GETOUT version used in the KBS safety analysis /1, 2, 6, 7/ suffered from two weaknesses:

a) Dispersion was not included for the nuclides that are formed through chain decay during the course of the transport.

b) "Band-Release", i.e. leaching at a constant rate over a given period of time, was simulated by a superimposition of an incremental increase of the leach rate at the time of canister breakthrough and an incremental decrease when the entire waste quantity is dissolved.

This approach is relevant for single- and double-nuclide chains, but gives too long-lasting releases to the recipient for some triple-nuclide chains.

In the current version, both of these weaknesses have been eliminated. The solution for the triple-nuclide chains has been revised so that "Band-Release" is simulated by a square wave instead of the former superimposition.
The numerical problems that were responsible for the omission of dispersion for the daughter nuclides in the former version (mainly cancellation and products between extremely large and extremely small numbers) have been solved by means of greater precision and by the separate handling of number parts and exponent parts. Thus, many of the products
8 that occur in the program are calculated as the sum of the natural logarithms of the factors.

Mathematically, the current version coincides with a listing of PNL's version dated August 15th, 1979. One addition, calculation of triple-nuclide chains with identical retention factors, has been made, however. The logic of the new version is described in Appendix 1. Appendix 2 is a listing of the program.

Another change in the current version is that in calculation cases where problems arise with negative square root arguments in the equations for dispersion release, the entire chain is not omitted, as in the PNL version, but rather only those terms concerned. If terms are omitted, a printout is obtained on the line printer where the omitted terms are marked so that the results can be checked.

3. Calculation and storage of source strengths at time of canister breakthrough

In the former version, the source strengths at the time of canister breakthrough were read directly from a formatted file with one column of source strengths for each of several fixed times. This system, which had been translated directly from the BASIC program, quickly proved to be unwieldy, since as soon as a time for canister breakthrough was chosen that was not in the available file, it was necessary to prepare a new source strength file and reprogram the tests by means of which the right column in the file was chosen.

In the current version, a routine package for calculating the source strengths has been included. This routine package also permits a reprocessing, i.e. a separation of a certain fraction of some of the nuclides, in which case the source strength calculation takes place in two steps.

In connection with the inclusion of the decay routine, internal storage of source strengths, half-lives and retention factors has also been introduced, making it possible to omit the temporary file that was previously used for storing these data (the former logic in this respect was directly transferred from the BASIC program). The advantage of this is a considerable reduction of both the execution time and the i/o-time.

4. Calculation of time scales

Different types of time scales are calculated in the program for:

- Single-nuclide chains
- Double- and triple-nuclide chains
- The sum of individual contributions to a nuclide's inflow and for the plot of the individual dose as a function of time that is obtained.

In the former version, various problems arose in the routines that calculate these time scales, entailing a risk that parts of the nuclide peaks would be missed. This in turn meant that some experience of the program was required to evaluate the runs.

The problem with the time scale for the single-nuclide chains was that there was a risk that the beginning and end of the peak in the dispersion case would be missed.

In order to avoid this problem, a routine that seeks out the point where the inflow is to be, 10^{-15} x max. inflow, has been put in. The routine works with the Newton Raphson method. In order to eliminate the risk of missing the no-dispersion peak, a point corresponding to the nuclide's travel time with plug-flow has been included in the otherwise logarithmic time scale.

The time scale for double- and triple-nuclide chains is linear. In some cases, there was formerly a risk that the maximum value would be missed because it fell in between two time steps. This risk has been considerably reduced by introducing a point at each nuclide's travel time with plug-flow and at the nuclide's travel time plus leach duration. At the same time, the number of time steps has been increased. Two temporary files, previously used to store the times used as limits in the time scale calculation for double- and triple-nuclide chains, have been replaced by internal fields in the program, improving efficiency.

The time scale for the sum of the individual nuclide contributions is logarithmic. In the former version, the maximum value sometimes fell in between two time steps. In the present version, one point is automatically placed at the time for the highest of the maximum values of the individual contributions and one point at the next-highest maximum.

5. Printouts for evaluation

In checking the calculation results to be used for dose calculations, it was formerly necessary to go through the entire run and check manually that the individual contributions were correctly added. This has now been considerably facilitated by tabulation of both the maximum values of individual contributions and the maximum value of the sum (see Appendix 4).

6. Plot program

A plot program, DRAW /8/, developed at VTT in Finland, has been adapted to the available hardware. Appendix 4 shows an example of the images that can be generated by DRAW.

REFERENCES

1. Handling of Spent Nuclear Fuel and Final Storage of Vitrified High-Level Reprocessing Waste. KBS (1977)

2. Handling and Final Storage of Unreprocessed Spent Nuclear Fuel. KBS (1978)

3. Lester P H, Jansen G, Burkholder H C: Migration of Radionuclide Chains through an Adsorbing Medium. AIChE Symposium Series No 152, 71 (1975), 202

4. Burkholder H C, Cloninger M O, Baker D A, Jansen G: Incentives for Partitioning High-Level Waste. BNWL-1972 (Nov 1975)

5. DeMier W.V., Cloninger M.O., Burkholder H.C., Liddell P.J.: GETOUT - A Computer Program for predicting Radionuclide Decay Chain Transport Through Geologic Media. PNL-2970 (Aug. 1979)

6. Grundfelt B: Transport of Radioactive Elements in Groundwater from a Rock Repository KBS Technical Report 43 (KEMAKTA Konsult AB 77-12-13).

7. Grundfelt B: Nuclide Migration from a Rock Repository for Spent Fuel KBS Technical Report 77 (KEMAKTA Konsult AB 78-08-31).

8. Raiko R: Plotting Code Draw. VTT Technical Report REP-8/79 (79-06-28).

Description of GETOUT and its input data requirements

GETOUT calculates the transport of radioactive nuclides from a final repository to the biosphere. The model is one-dimensional and includes dispersion, geochemical retardation and chain decay. The transport equation has been solved analytically for a constant leach rate /5/. The result is obtained in the form of the activity inflow to a recipient (Ci/year) as a function of time. The transport equation has been solved for decay chains of single-, double- and triple-nuclide length. Longer decay chains are simulated by means of one of two approximations.

1) APPROX 1: Radioactive equilibrium, the daughter nuclide is assumed to be very short-lived in relation to the parent nuclide.

2) APPROX 2: Short-lived parent nuclide that is assumed to decay completely before the start of leaching.

The total inflow of a given nuclide is then modelled as the sum of the contributions from the different parent nuclides. Figures 1-4 show examples of how the decay chains of the actinides can be "pieced together".

The designation PSEUDO also occurs in the figures, which means that the uranium-238 inventory in the waste is divided into two pseudo-inventories of uranium-234 by means of the decay equation for double-nuclide chains. Radium-226 from uranium-238 can then be calculated as the difference between two triple-nuclide chains with the two pseudo-inventories as parent nuclides. Thorium-230 from uranium-238 can be calculated in the same manner as the difference between two double-nuclide chains.

Single-, double- and triple nuclides chains are designated in the figures as SINGLE, DOUBLE and TRIPLE after the subroutines in the program that perform the calculations.

The program logic for GETOUT is described in Figure 5. The figure is not a flow scheme, but rather an illustration of the principle of the information flow.

The single-, double- and triple-nuclide chains are calculated by the subroutines SINGLE, DOUBLE and TRIPLE with associated auxiliary routines. The above-mentioned approximations APPROX 1 and APPROX 2 are calculated in the routine SPEC. The individual contributions to a nuclide's total inflow are stored for single-nuclide chains on file 13, for double-nuclide chains on file 14, for triple-nuclide chains on file 15 and for the approximations on file 16. If requested, a listing of the individual contributions is obtained on file 9. The total inflows are obtained as a line printer plot on file 20 and in unformatted form on file 17 for possible further use as input to a special plot program. If desired, the total inflows can also be taken out on file 7 in formatted form for further use in e.g. a dose calculation program. GETOUT also calculates an approximate dose by means of multiplication by a dose factor which can be, for example, a maximum individual dose per unit inflow to the recipient area. This "dose factor" may, for example, be based on previous runs with a biosphere model. The resulting dose is obtained as a line printer plot on file 20.

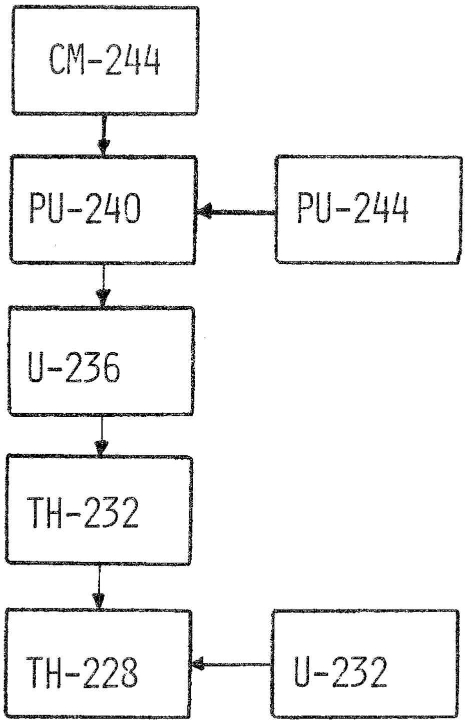
Two files are needed as input data to a run:

- File 1 with the nuclides' half-lives (years) and source strength (Ci/t U) in the irradiated fuel upon discharge from the reactor. One record for each nuclide with nuclide name, half-life and source strength, FORMAT (A6, 2E10.2). The file is concluded with a record with -1 in columns 1-2 and two records with file identification in clear text FORMAT (20A4). Figure 6 shows a listing of a source strength file.

- File 5 with information on the time of canister breakthrough, leach duration, groundwater travel time, nuclides to be calculated, retention factors (coefficients) for these nuclides etc. The records in file 5 are described in Appendix 3.

The 4N chain

Figure 1

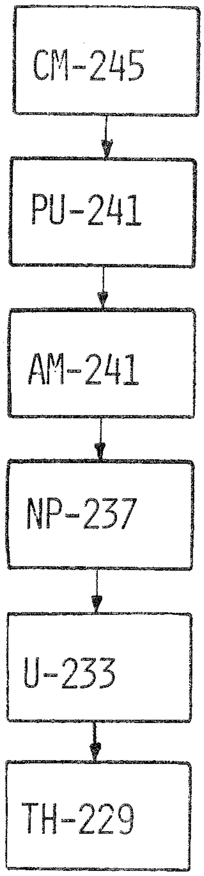


Calculation logic for the 4N chain

(chain)

Cm-244	from Cm-244:	SINGLE
Pu-244	from Pu-244:	SINGLE
Pu-240	from Cm-244:	DOUBLE
	from Pu-244:	neglected (the inventory of Pu-244 is very small)
U-236	from Cm-244:	TRIPLE
	from Pu-244:	neglected (see above)
	from Pu-240:	DOUBLE
	from U-236:	SINGLE
Th-232	from Cm-244:	TRIPLE (Th-232 from Pu-240 with APPROX 2)
	from Pu-244:	neglected (see above)
	from Pu-240:	TRIPLE
	from U-236:	DOUBLE
	from Th-232	SINGLE
U-232	from U-232:	SINGLE
Th-228		= Th-232 (the half-life of Th-228 is much smaller than that of Th-232 and the inventory of U-232 is small)

The 4N+1 chain Figure 2

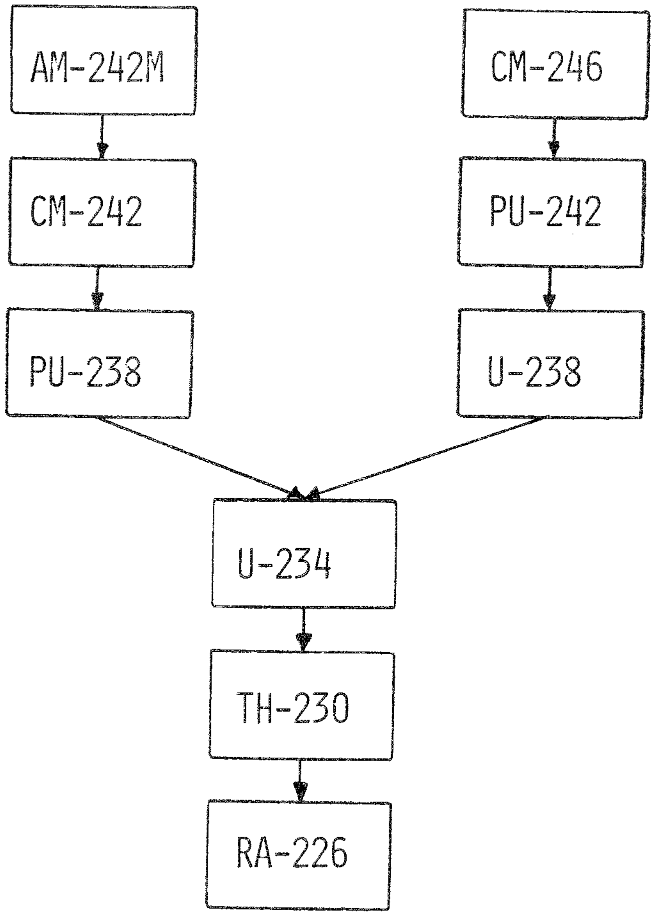


Calculation logic for the 4N + 1 chain

(chain)		
Cm-245	from Cm-245:	SINGLE
Pu-241	from Cm-245:	DOUBLE
	from Pu-241:	SINGLE
Am-241	from Cm-245:	DOUBLE (half-life of Pu-241 <<half-life of Cm-245)
	from Pu-241:	DOUBLE
	from Am-241:	SINGLE
Np-237	from Cm-245:	TRIPLE (half-life of Pu-241 <<half-life of Am-241)
	from Pu-241:	TRIPLE
	from Am-241:	DOUBLE
	from Np-237:	SINGLE
U-233	from Cm-245:	TRIPLE (half-life of Pu-241 and Am-241 <<half-life of Cm-245)
	from Pu-241:	TRIPLE (U-233 from Am-241 with APPROX 2)
	from Am-241:	TRIPLE
	from Np-237:	DOUBLE
	from U-233:	SINGLE
Th-229	from Cm-245:	neglected (inventory of Cm-245 << inventories of Np-237 and Am-241)
	from Pu-241 and Am-241:	TRIPLE (Th-229 from Np-237 with APPROX 2)
	from Np-237:	TRIPLE
	from U-233:	DOUBLE
	from Th-229:	SINGLE

The 4N+2 chain

Figure 3



Calculation logic for the 4N + 2 chain

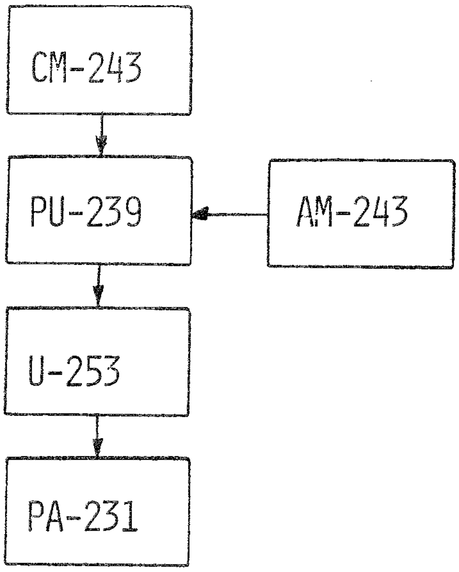
(chain)

Am-242M	from Am-242M:	SINGLE
Cm-242	from Am-242M:	DOUBLE
	from Cm-242:	SINGLE
Pu-238	from Am-242M:	DOUBLE (half-life of Cm-242 << inventories of Pu-238 and Am-242M)
	from Cm-242:	neglected (inventory of Cm-242 <<inventories of Pu-238 and Am-242M)
	from Pu-238:	SINGLE
Cm-246	from Cm-246:	SINGLE
Pu-242	from Cm-246:	DOUBLE
	from Pu-242:	SINGLE
U-238	from Cm-246:	TRIPLE
	from Pu-242:	DOUBLE
	from U-238:	SINGLE
U-234	from Am-242M:	TRIPLE (half-lives of Cm-242 << half-life of Am-242M)
	from Cm-242:	DOUBLE (U-234 from Pu-238 with APPROX 2)
	from Pu-238:	DOUBLE
	from Cm-246:	TRIPLE (U-238 from Cm-246 with APPROX 1)
	from Pu-242:	DOUBLE (U-238 from Pu-242 with APPROX 1)
	from U-238:	DOUBLE
	from U-234:	SINGLE

Th-230	from Th-230:	SINGLE
	from U-234:	DOUBLE
	from U-238:	DOUBLE (with PSEUDO)
	from Pu-242:	DOUBLE (U-238 from Pu-242 with APPROX 1)
	from Cm-246:	neglected (inventory of Cm-246 <<inventory of U-238)
	from Pu-238:	TRIPLE
	from Cm-242:	TRIPLE (Th-230 from Pu-238 with APPROX 2)
	from Am-242M:	TRIPLE (half-lives of Cm-242 and Pu-238 <<half-life of Am-242M)
Ra-226	from Ra-226:	SINGLE
	from Th-230:	DOUBLE
	from U-234:	TRIPLE
	from U-238:	TRIPLE (with PSEUDO)
	from Pu-242:	DOUBLE (U-238 from Pu-242 with APPROX 1)
	from Cm-246:	neglected (see Th-230 from Cm-246)
	from Pu-238, Cm-242 and Am-242M:	TRIPLE (Ra-226 from U-234 with APPROX 2)

The 4N+3 chain

Figure 4



Calculation logic for the 4N + 3 chain

(chain)		
Am-243	from Am-243:	SINGLE
Cm-243	from Cm-243:	SINGLE
Pu-239	from Am-243:	DOUBLE
	from Cm-243:	DOUBLE
	from Pu-239:	SINGLE
U-235	from Am-243:	TRIPLE
	from Cm-243:	neglected (the inventory of Cm-243 is small in relation to the inventory of Am-243)
	from Pu-239:	DOUBLE
	from U-235:	SINGLE
Pa-231	from Am-243: and Cm-243:	TRIPLE (Pa-231 from Pu-239 with APPROX 2)
	from Pu-239:	TRIPLE
	from U-235:	DOUBLE
	from Pa-231:	SINGLE

Figure 5

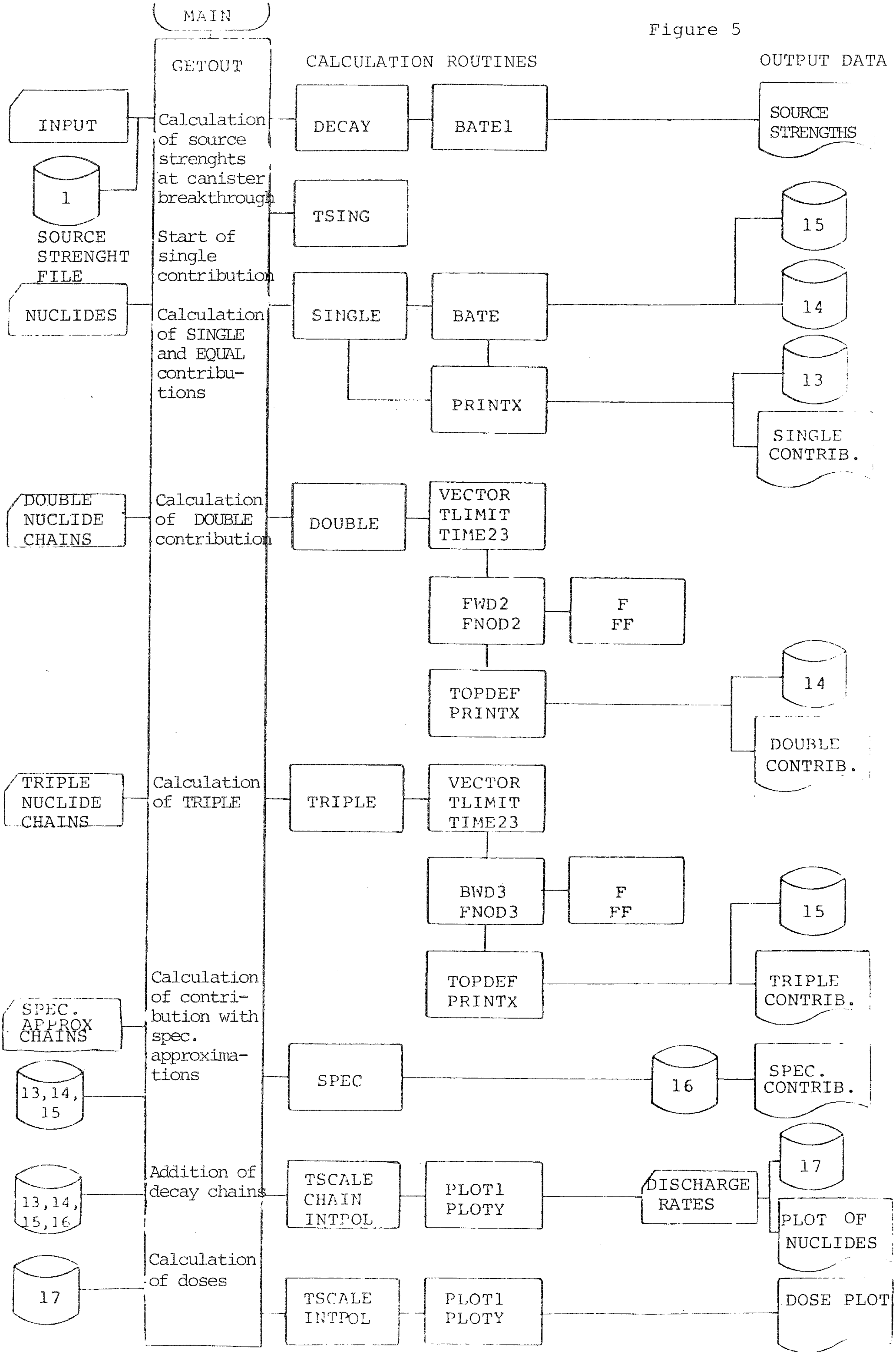

MAIN
GETOUT
CALCULATION ROUTINES
OUTPUT DATA
INPUT
1
SOURCE STRENGHT FILE
Calculation of source strenghts at canister breakthrough
DECAY
BATE1
SOURCE STRENGTHS
TSING
Start of single contribution
15
14
NUCLIDES
Calculation of SINGLE and EQUAL contributions
SINGLE
BATE
PRINTX
13
SINGLE CONTRIB.
DOUBLE NUCLIDE CHAINS
Calculation of DOUBLE contribution
DOUBLE
VECTOR TLIMIT TIME23
FWD2 FNOD2
F FF
TOPDEF PRINTX
14
DOUBLE CONTRIB.
TRIPLE NUCLIDE CHAINS
Calculation of TRIPLE
TRIPLE
VECTOR TLIMIT TIME23
BWD3 FNOD3
F FF
15
TOPDEF PRINTX
TRIPLE CONTRIB.
SPEC. APPROX CHAINS
Calculation of contribution with spec. approximations
13,14, 15
SPEC
16
SPEC. CONTRIB.
13,14, 15,16
Addition of decay chains
TSCALE CHAIN INTPOL
PLOT1 PLOTY
DISCHARGE RATES
17
PLOT OF NUCLIDES
17
Calculation of doses
TSCALE INTPOL
PLOT1 PLOTY
DOSE PLOT

Figure 6

Listing of source strength file

475-SOURCE	PAGE	4
H 3	1.23+01	7.30+02
C 14	5.74+03	1.45+00
SE79	6.50+04	3.95-01
KR85	1.08+01	1.12+04
RB87	5.00+10	1.92-05
SR89	1.43-01	7.99+05
SR90	2.81+01	7.62+04
Y 91	1.62-01	1.05+06
ZR93	1.50+06	1.86+00
ZR95	1.79-01	1.57+06
NB95	9.59-02	1.58+06
TC99	2.11+05	1.43+01
RU103	1.08-01	1.42+06
RU106	1.00+00	6.18+05
PD107	7.00+06	1.17-01
CD109	1.24+00	4.52-06
AG110M	6.93-01	4.33+03
CD113M	1.40+01	1.17+01
SB125	2.70+00	9.45+03
TE125M	1.59-01	3.33+03
TE127M	2.98-01	1.80+04
SN126	9.98+04	5.69-01
I 129	1.70+07	3.75-02
CS134	2.05+00	2.65+05
CS135	3.00+06	2.51-01
CS137	3.00+01	1.09+05
CE144	7.79-01	1.23+06
PM147	2.62+00	1.01+05
SM151	8.72+01	1.26+03
EU152	1.20+01	1.06+01
EU154	1.60+01	7.34+03
EU155	1.82+00	7.86+03
HO166M	1.20+03	8.20-04
RA226	1.60+03	1.12-08
TH228	1.91+00	1.21-03
TH229	7.30+03	2.80-08
TH230	8.00+04	1.38-05
TH232	1.41+10	2.13-11
PA231	3.25+04	1.97-05
U 232	7.20+01	5.41-03
U 233	1.62+05	4.16-05
U 234	2.47+05	6.72-01
U 235	7.10+08	1.48-02
U 236	2.39+07	2.71-01
U 238	4.51+09	3.15-01
NP237	2.13+06	3.28-01
PU236	2.85+00	3.74-01
PU238	8.90+01	2.78+03
PU239	2.44+04	3.18+02
PU240	6.76+03	4.90+02
PU241	1.46+01	1.10+05
PU242	3.79+05	1.53+00
PU244	8.00+07	2.21-15
AM241	4.33+02	7.78+01
AM242M	1.52+02	8.36+00
AM243	7.65+03	2.10+01
CM242	4.46-01	3.45+04
CM243	32.00+00	3.96+00
CM244	1.82+01	3.00+03
CM245	8.26+03	4.32-01
CM246	4.71+03	9.20-02
CM248	3.52+05	1.18-06

-1
SOURCE STRENGTHS AND HALFLIFES FOR SPENT PWR-FUEL, HALFLIFES ARE ACCORDING TO
ORIGEN'S DATA LIBRARY. SOURCE STRENGTHS ARE TAKEN FROM KBS TR 01 (1977-04-05)

Program listing

Routine	Card No	Routine	Card No
MAIN	1	FWD2	2 714
BLOCK DATA	17	FNOD2	2 909
GETOUT	59	FWD3	3 009
DECAY	557	BWD3	3 066
BATE1	717	FNOD3	3 613
SINGLE	766	BB16	3 815
DOUBLE	909		
TRIPLE	1 025		
TSING	1 145		
TIME23	1 180		
TLIMIT	1 225		
VECTOR	1 263		
GTEST	1 301		
F	1 339		
FF	1 389		
SUM2	1 605		
FERRNT	1 662		
DIFERF	1 691		
BATE	1 735		
TOPDEF	1 765		
PRINTX	1 816		
SPEC	1 876		
CHAIN	1 971		
INTPOL	2 020		
TSCALE	2 041		
PLOT1	2 132		
PLOTY	2 189		
SCALEY	2 602		
DECODH	2 639		
TEXP	2 676		
TEXP1	2 684		
TDERFC	2 692		
INTNUK	2 700		

```
      LOGICAL MIGOUT,NOPUN,NOLINK,LINK,PUNCH,WRITEX
      COMMON/PRINT/ MIGOUT
      CALL ERRSET(208,0,-1,1,1)
      CALL ERRSET(207,50,-1,1,1)
      READ(5,1)LINK,PUNCH,WRITEX,MIGOUT
    1 FORMAT(5L5)
      IF(.NOT.(LINK.OR.MIGOUT))GO TO 100
      NOLINK=.NOT.LINK
      NOPUN=.NOT.PUNCH
      CALL GETOUT(NOLINK,NOPUN,WRITEX)
      STOP
  100 WRITE(6,2)
    2 FORMAT('1***  NO OUTPUT WAS SPECIFIED AND AS IT IS LATE THE COMPUT
     *ER CHOOSED TO TERMINATE EXECUTION ***')
      STOP
      END
      BLOCK DATA
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
C
C     BLOCKDATA INITIATES NUCLIDE NAMES AND RETENTION COEFFICIENTS TO
C     DUMMY VALUES
C
     *               GAP(6),BREAK,DUR,PE,ETA,TD,V,PATH
      COMMON/DEY1/    FISS(40),CHAIN(6,3,4),IFI,ICH
      COMMON/DEY2/   ETEXP(75),AN(75),C(75),ANAME(76),SF(75),NUK(75),
     *               IN,ISL
      COMMON/PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
     *               CMAX(75,10),DOSE2(75)
      COMMON/HALVA/ THALV(75),RET(75)
      REAL*16 THALV
      REAL*16 ETEXP,AN,C
      REAL*8 ENAME,ANAME,FISS,CHAIN
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,BREAK,DUR,PE,ETA,TD,V,PATH
      DATA EQ/3*0.0/
      DATA ETA /1.D0/
      DATA ICONTR /75*0/
      DATA RET/75*0.0/
      DATA SF/75*1./
      DATA IFI/24/,
     1     FISS/'H 3    ','C 14   ','KR85   ','SR89   ','SR90   ','Y 91   ',
     *'ZR93   ',
     2  'ZR95   ','NB95   ','TC99   ','RU103  ','RU106  ','SB125  ','TE125M',
     3        'TE127M','I 129  ','CS134  ','CS135  ','CS137  ','CE144  ',
     4        'PM147  ','SM151  ','EU154  ','EU155  ',16*'XX000  '/
      DATA ICH/4/,
     1     CHAIN/'CM246  ','PU242  ','U 238  ','U 234  ','TH230  ','RA226  ',
     2        'AM242M','CM242  ','PU238  ','U 234  ','TH230  ','RA226  ',
     3          6*'      ',
     4          '      ','CM244  ','PU240  ','U 236  ','TH232  ','TH228  ',
     5        'CM248  ','PU244  ','PU240  ','U 236  ','TH232  ','TH228  ',
     6          3*'      ',              'PU236  ','U 232  ','TH228  ',
     7          2*'      ',      'CM243  ','PU239  ','U 235  ','PA231  ',
     8          2*'      ',      'AM243  ','PU239  ','U 235  ','PA231  ',
     9          6*'      ',
```

```
     *             'CM245 ','PU241 ','AM241 ','NP237 ','U 233 ','TH229 ',
     1       12*'      '/
      DATA ENAME /3*'XX000'/
      END
      SUBROUTINE GETOUT(NLINK,NPUNCH,WRITEX)
C
C     THIS ROUTINE ADMINISTRATES THE CALCULATIONS OF THE NUCLIDE
C     RELEASE RATES
C
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
     *               GAP(6),BREAK,DUR,PE,ETA,TD,WATVEL,PATH
      COMMON/C2/     NSTEP,U(304),Y(304),DOSE1(304),TIME(304),
     *               TBEGIN,TFIN,TTOPP,BRPEAK,DINO
      COMMON/HALVA/  THALV(75),RET(75)
      COMMON/DEY2/  ETEXP(75),AN(75),C(75),ANAME(76),SF(75),NUK(75),
     *              IN,ISL
      COMMON /PEAKS/ICONTR(75),TSTART(75,10),TPEAK (75,10),TEND(75,10),
     *              CMAX(75,10),DOSE2(75)
      COMMON /LGTEST/ LG(6)
      REAL*16 T,TUPP,THALV,ETEXP,AN,C
      REAL*8 RNAME(75),ENAME,SNAME,SNAME1,SNAME2,
     *SNAME3,COMP,IC1(20),IBL,IDOS,BNAME,
     *        ANAME,TEXT1(2),TEXT2(6),TEXT(2),GNAME(3),NNN
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,BREAK,DUR,PE,ETA,TD,WATVEL,PATH
      LOGICAL AKT,CLAY,INS,SUB,NLINK,NPUNCH,WRITEX
      LOGICAL LG
      DIMENSION CA(75),YMAX(10),TIMY(10),DOSMAX(75),TPE(75),
     *          TSLASK(90),TMI(75),TMA(75),UMAX(75)
      DATA IUPP/0/
      DATA TEXT /'GLASS','FUEL'/
      DATA TEXT1 /'POROSITY','SPACING '/
      DATA TEXT2 /'CU.METER','METERS  ','S/CU.MET','        ','ER      '
     *,'        '/
      DATA IDOS/'DOSES'/
      DATA COMP/'-1'/,IBL/'        '/
      DATA ALN2 /0.6931472/
  805 FORMAT(I2)
  810 FORMAT(8F10.0)
  820 FORMAT(A6,4X,F10.0)
  920 FORMAT(1X,A6,1PE10.2,10E10.2)
  930 FORMAT('1TIME   ',10X,1PE10.2,10E10.2)
      READ(5,1)NS,J3
C
    1 FORMAT(6I5)
      READ(5,2)BREAK,DUR,DIFF,TONNE
    2 FORMAT(5E10.2)
      READ(5,3)PERM,GRAD,SPACE,PATH,CLAY
    3 FORMAT(4E15.4,L5)
C
      IF(CLAY)GO TO 80
      WATVEL=2.842E9*GRAD*(SPACE*PERM)**(2.0/3.0)
      DISP=DIFF+5.04E-3*(SPACE*PERM*GRAD)**2/DIFF
      DISP=DISP*3.1536E7
      GO TO 90
   80 WATVEL=3.1536E7*PERM*GRAD/SPACE
      DISP=DIFF*3.1536E7*SPACE/2.0
```

```
   90 CONTINUE
C
C     INITIATE THE DECAY ROUTINE
C
      CALL DECAY(1)
C
      T=BREAK
C
      READ(5,810) TUPP
      IF (TUPP.EQ.0.) GO TO 20
C
C     IF REPROCESSING READ THE SPLITFACTORS DEVIATING FROM 1.
C
      READ(5,805)IANT
      DO 11 I=1,IANT
      READ(5,820) BNAME,VARDE
      SF(INTNUK(BNAME))=VARDE
11    CONTINUE
C
C     CALCULATION OF SOURCE STRENGTHS AT THE TIME OF LEACH INCIDENT
C
20    IF (TUPP.EQ.0.) TUPP=1.E20
      IF (T.LT.TUPP.OR.IUPP.EQ.2) GO TO 25
      CALL DEC(TUPP,TUPP,1,J,1)
      IUPP=2
25    IF (IUPP.EQ.0) CALL DEC(T,T,IUPP,J,0)
      IF (IUPP.EQ.2) CALL DEC(T-TUPP,T,IUPP,J,0)
      DO 29 L=1,ISL
      CA(L)=C(L)*TONNE
29    CONTINUE
      WRITE(6,930)T
      DO 32 I=1,ISL
      WRITE(6,920)ANAME(I),THALV(I),CA(I)
   32 CONTINUE
      REWIND 1
      JK=2
      IF(CLAY)JK=1
      JJ=2
      IF(TUPP.NE.1.E20) JJ=1
      WATIM=PATH/WATVEL
      WRITE(6,41)TEXT(JJ),BREAK,DUR,PATH,WATVEL,WATIM,DISP,TONNE
     *,PERM,GRAD,TEXT1(JK),SPACE,TEXT2(JK),TEXT2(JK+2),TEXT2(JK+4)
   41 FORMAT('1CALCULATION OF RADIONUCLIDE CHAIN MIGRATION FROM AN UNDER
     *GROUND REPOSITORY',/1X,74('*')/'0WASTE FORM: ',A8/'0TIME OF LEACH
     *INCIDENT ',1PE10.1,' YEARS AFTER REACTOR DISCHARGE'/' LEACH DURATI
     *ON ',8X,E10.1,' YEARS'/' MIGRATION PATH LENGTH ',1X,E10.1,' METERS
     *'/' GROUNDWATER VELOCITY ',2X,E10.1,' METERS/YEAR'/' GROUNDWATER T
     *RAVEL TIME',E10.1,' YEARS'/
     *                                                      ' DISPERSION CO
     *EFFICIENT ',E10.1,' SQ.METERS/YEAR'/' INVOLVED WASTE AMOUNT ',1X,E
     *10.1,' TONNES'/' PERMEABILITY',11X,E10.1,' METERS/SECOND'/' GRADIE
     *NT ',14X,E10.1,' METERS/METER'/1X,A8,15X,E10.1,1X,A8,A8,A8/)
      WRITE(6,61)
   61 FORMAT(1X//' NUCL.',5X,'H.LIFE (Y)',2X,'SOURCE (CI)',1X,
     *       'RET.',9X,'DOSE FACT'/)
       III=1
```

```
      C
      C     STARTING POINT OF TIMESCALE IN SINGLE
      C
            PE=PATH*WATVEL/DISP
            CALL TSING(PE,TH1)
      C
      C     START OF SINGLE LOOP
      C
        100 CONTINUE
            READ(5,44)SNAME,AKT,AREAK,VOLK,DOSE
         44 FORMAT(A6    ,L2/3E10.2)
            IF(SNAME.EQ.COMP)GO TO 200
            IJ=INTNUK(SNAME)
            IF(CLAY)RET(IJ)=1+VOLK*(1-SPACE)/SPACE
            IF(.NOT.CLAY)RET(IJ)=1+AREAK*2./.0105/
           *  (PERM*SPACE)**.333333
            THALF(1)=THALV(IJ)
            AQ(1)=CA(IJ)
            ENAME(1)=SNAME
            EQ(1)=RET(IJ)
            ALAMB(1)=ALN2/THALF(1)
            DOSE2(IJ)=DOSE
            WRITE(6,77)SNAME,THALF(1),AQ(1),EQ(1),DOSE
         77 FORMAT(1H ,A6    ,1PE12.2,9E12.2)
            CALL SINGLE(TH1,III,AKT,NS)
            GO TO 100
      C
      C     START OF DOUBLE LOOP
      C
        200 CONTINUE
            WRITE(6,205)
        205 FORMAT('0')
            II=0
            III=2
        210 CONTINUE
            READ(5,5)SNAME1,SNAME2,INS,SUB
          5 FORMAT(A6    ,4X,A6    ,2L2)
            IF(SNAME1.EQ.COMP)GO TO 300
            IJ1=INTNUK(SNAME1)
            IJ2=INTNUK(SNAME2)
            IF(RET(IJ1).EQ.0..OR.RET(IJ2).EQ.0.)GO TO 250
            THALF(1)=THALV(IJ1)
            THALF(2)=THALV(IJ2)
            ALAMB(1)=ALN2/THALF(1)
            ALAMB(2)=ALN2/THALF(2)
            AQ(1)=CA(IJ1)
            AQ(2)=CA(IJ2)
            ENAME(1)=SNAME1
            ENAME(2)=SNAME2
            EQ(1)=RET(IJ1)
            EQ(2)=RET(IJ2)
            IF(II)260,260,320
        250 CONTINUE
            WRITE(6,211)SNAME1,SNAME2
        211 FORMAT('XDOUBLE: ',2A8,'NO INPUT PROVIDED IN SINGLE-LOOP')
            GO TO 210
```

```
  260 CONTINUE
C
C      TWO MEMBER CHAINS WITH EQUAL RETENTICN COEFFICIENTS CALCULATED BY
C      SINGLE.
C
       IF(DABS(EQ(1)-EQ(2))-0.01*EQ(1))261,261,265
  261 CONTINUE
       CALL SINGLE(THI,III,AKT,NS)
       WRITE(6,262)ENAME(1),ENAME(2)
  262 FORMAT(' EQUAL:',2X,2A8)
       GO TO 210
  265 CONTINUE
       WRITE(6,8)((ENAME(I),EQ(I)),I=1,2)
    8 FORMAT(1H0,42HATTEMPT TO TREAT UNEQUAL COEFF. AS EQUAL: ,A6    ,4H
     *K= ,1PE10.2/1H ,42X,A6    ,4H K= ,E10.2/1H ,34HCHAIN DELETED,EXECUT
     *ION CONTINUING)
       GO TO 210
  300 CONTINUE
       IF(II-1)310,350,350
  310 CONTINUE
       WRITE(6,205)
       II=1
       GO TO 210
C
C      CALL FOR DOUBLE
C
  320 CONTINUE
       CALL DOUBLE(INS,SUB,J3)
       WRITE(6,330)ENAME(1),ENAME(2)
       IF(.NOT.INS)WRITE(6,340)LG(1),LG(2)
  340 FORMAT('+',T28,'TERMS OMITTED IN BWD2:  2(',L1,'), 4(',L1,')')
  330 FORMAT(' DOUBLE:',1X,2A8)
       GO TO 210
  350 CONTINUE
C
C      END OF DOUBLE LOOP, START OF TRIPLE LOOP
C
       WRITE(6,205)
       III=3
       II=0
  400 CONTINUE
       READ(5,16)SNAME1,SNAME2,SNAME3,INS
   16 FORMAT(A6    ,4X,A6    ,4X,A6    ,L2)
       IF(SNAME1.EQ.COMP)GO TO 500
       IJ1=INTNUK(SNAME1)
       IJ2=INTNUK(SNAME2)
       IJ3=INTNUK(SNAME3)
       IF(RET(IJ1).EQ.0..OR.RET(IJ2).EQ.0..OR.RET(IJ3).EQ.0.)GO TO 452
       THALF(1)=THALV(IJ1)
       THALF(2)=THALV(IJ2)
       THALF(3)=THALV(IJ3)
       ALAMB(1)=ALN2/THALF(1)
       ALAMB(2)=ALN2/THALF(2)
       ALAMB(3)=ALN2/THALF(3)
       AQ(1)=CA(IJ1)
       AQ(2)=CA(IJ2)
```

```
      AQ(3)=CA(IJ3)
      ENAME(1)=SNAME1
      ENAME(2)=SNAME2
      ENAME(3)=SNAME3
      EQ(1)=RET(IJ1)
      EQ(2)=RET(IJ2)
      EQ(3)=RET(IJ3)
      IF(II.EQ.0)GO TO 455
C
C        CALL FOR TRIPLE
C
      CALL TRIPLE(INS,J3)
      WRITE(6,450) ENAME
      IF(.NOT.INS)WRITE(6,440)LG(6),LG(5),LG(5),LG(4),LG(6),LG(3),LG(4),
     * LG(5),LG(1),LG(3),LG(5),LG(1),LG(2),LG(3)
  440 FORMAT('+',T34,'TERMS OMITTED IN BWD3: 5(',2L1,'), 6(',2L1,'),','
     * 7(',L1,'), 8(',L1,') , 9(',2L1,'), 10(',L1,'), 11(',L1,'), 12(',L
     *1,'), 13(',L1,'), 14(',L1,'), 15(',L1,')')
  450 FORMAT(' TRIPLE:',1X,3A8)
      GO TO 400
  452 CONTINUE
      WRITE(6,411)SNAME1,SNAME2,SNAME3
  411 FORMAT('XTRIPLE: ',3A8,'NO INPUT PROVIDED IN SINGLE LOOP')
      GO TO 400
C
C     THREE MEMBER CHAINS WITH EQUAL RETENTION COEFFICIENTS CALCULATED
C     BY SINGLE
C
  455 CONTINUE
      IF(EQ(1).NE.EQ(2).OR.EQ(1).NE.EQ(3))GO TO 460
      CALL SINGLE(TH1,III,AKT,NS)
      WRITE(6,451)ENAME
  451 FORMAT(' EQUAL: ',3A8)
      GO TO 400
  460 CONTINUE
      WRITE(6,461)ENAME,EQ
  461 FORMAT('0ATTEMPT TO TREAT UNEQUAL COEFFICIENTS AS EQUAL IN THE CHA
     *IN: ',3(4X,A6)/T58,'EQ= ',3F10.0)
      GO TO 400
C
  500 CONTINUE
      IF(II.EQ.1)GO TO 508
      II=1
      WRITE(6,205)
      GO TO 400
  508 CONTINUE
C
C     END OF TRIPLE LOOP
C
      DO 513 JJ=13,15
      ENDFILE JJ
      BACKSPACE JJ
      REWIND JJ
  513 CONTINUE
C
C     THE "SPECIAL APPROXIMATION LOOP".
```

```
C
      REWIND 16
      WRITE(6,205)
11000 CONTINUE
      DO 11001 K=1,3
      GNAME(K)=IBL
11001 ENAME(K)=IBL
      READ(5,11002)GNAME,IAPPR,ENAME
11002 FORMAT(3A8,I6,3A8)
      IF(GNAME(1).EQ.COMP)GO TO 12000
      IF(IAPPR.EQ.1.AND.ENAME(1).NE.GNAME(2))GO TO 11300
      IFILE=3
      DO 11100 K=1,3
      IF(ENAME(K).NE.IBL)GO TO 11100
      IFILE=K-1
      GO TO 11200
11100 CONTINUE
11200 IF (IFILE.EQ.0) GO TO 11300
      IF(IAPPR.EQ.1)GO TO 11250
      IF (ENAME(IFILE).NE.GNAME(1))GO TO 11300
      IJ=INTNUK(ENAME(1))
      Q1=CA(IJ)*THALV(IJ)
      IF(Q1.LE.0.0)GO TO 11400
      IJ=INTNUK(GNAME(2))
      Q2=CA(IJ)*THALV(IJ)
      FACT=Q2/Q1
      IF(GNAME(3).EQ.IBL)GO TO 11250
      IJ=INTNUK(GNAME(3))
      Q2=CA(IJ)*THALV(IJ)
      FACT=FACT+Q2/Q1
      IF(FACT.LE.0)GO TO 11500
11250 CONTINUE
      CALL SPEC(GNAME,ENAME,IFILE,IAPPR,FACT,&11000)
      WRITE(6,11251)GNAME,ENAME,IAPPR
11251 FORMAT(' SPEC:',3X,3A8,5X,3A8,'IAPPR= ',I3)
      GO TO 11000
11300 CONTINUE
      WRITE(6,11301)ENAME,GNAME,IAPPR
11301 FORMAT('0SORRY I CANNOT DO THE IMPOSSIBLE,LOOK FOR YOURSELF: ENAME
     *= ',3A8/55X,'GNAME= ',3A8/55X,'IAPPR= ',I3/)
      GO TO 11000
11400 CONTINUE
      WRITE(6,11401)(GNAME(I),I=1,2)
      IF(GNAME(3).NE.IBL)WRITE(6,11402)GNAME(3)
      WRITE(6,11403)Q1,ENAME(1)
11401 FORMAT('0THE CHAIN ',A6,' FROM ',A6)
11402 FORMAT('+',29X,'AND ',A6)
11403 FORMAT(' WAS NOT CALCULATED BY SPEC BECAUSE OF ',1PE10.2,'-INVENTO
     *RY FOR ',A6)
      GO TO 11000
11500 CONTINUE
      WRITE(6,11501)GNAME,ENAME,FACT
11501 FORMAT('0THE CHAIN ',3A8,'WAS NOT CALCULATED BY SPEC BECAUSE '/' T
     *HE CHAIN ',3A8,' GAVE FACT= ',1PE10.2/)
      GO TO 11000
12000 CONTINUE
```

```
      ENDFILE 16
      BACKSPACE 16
      REWIND 16
C
C     END OF "SPECIAL APPROXIMATION LOOP".
C
      IF(NLINK)RETURN
C
C     START OF PEAK COMBINATION LOOP
C
      DO 600 I=1,ISL
      IJ=I
      TMA(IJ)=0.0
      TMI(IJ)=0.0
      TPE(IJ)=0.0
      UMAX(IJ)=0.0
      TMAX=0.0
      TMIN=0.0
      NN=ICONTR(IJ)
      DO 505 J=1,20
      IC1(J)=IBL
  505 CONTINUE
      IF(NN.EQ.0)GO TO 600
      IF(NN.EQ.1.AND.CMAX(IJ,1).EQ.0)GO TO 590
      IF(NN.EQ.1)GO TO 570
C
      DO 510 J=1,NN
      YMAX(J)=CMAX(IJ,J)
      TIMY(J)=TPEAK(IJ,J)
      TSLASK(3*J-2)=TSTART(IJ,J)
      TSLASK(3*J-1)=TPEAK(IJ,J)
      TSLASK(3*J)=TEND(IJ,J)
  510 CONTINUE
C
      TMAX=0.0
      II=3*NN
      DO 530 J=3,II,3
      IF(YMAX(J/3).LT.1.E-20)GO TO 530
      IF(TSLASK(J).GT.TMAX)TMAX=TSLASK(J)
  530 CONTINUE
      TMIN=TMAX
      DO 540 J=1,II,3
      IF(YMAX((J+2)/3).LT.1.E-20)GO TO 540
      IF(TSLASK(J).LT.TMIN.AND.TSLASK(J).GE.10.)TMIN=TSLASK(J)
  540 CONTINUE
      IF(TMIN.EQ.TMAX)GO TO 590
      IF(TMIN.LE.0.0.OR.TMAX.LE.0.0)GO TO 590
      TMA(IJ)=TMAX
      TMI(IJ)=TMIN
      CALL TSCALE(YMAX,TIMY,NN,TMAX,TMIN,U,NT,&590)
      CALL CHAIN(ANAME(I),NT,IC1,I2)
      RMAX=0.0
      DO 564 J=1,NT
      IF(Y(J).LE.RMAX)GO TO 564
      RMAX=Y(J)
      TPE(IJ)=U(J)
```

```
  564 CONTINUE
      UMAX(IJ)=RMAX
      IF(NPUNCH)GO TO 8000
      WRITE(7,561)ANAME(I),NT,U(1),U(NT),RMAX
  561 FORMAT(A6    ,4X,I4,1X,1PE15.5,2E15.5)
      WRITE(7,1000)((U(J),Y(J)),J=1,NT)
 1000 FORMAT(1PE9.2,7E9.2)
 8000 CONTINUE
      CALL  PLOT1(ANAME(I),IC1,I2,NT,RMAX,WRITEX)
      WRITE(17)ANAME(I),NT
      WRITE(17)(U(J),Y(J),J=1,NT)
      I2=I2-1
      GO TO 600
  570 READ(13,END=590)ENAME,ISTEP
      IF(ENAME(1).EQ.ANAME(I))GO TO 575
      IF(ISTEP.EQ.0)GO TO 570
      READ(13)(U(J),Y(J),J=1,ISTEP)
      GO TO 570
  575 CONTINUE
      IF(ISTEP.EQ.0)GO TO 590
      READ(13)(U(J),Y(J),J=1,ISTEP)
      TMI(IJ)=U(1)
      TMA(IJ)=U(ISTEP)
      RMAX=CMAX(IJ,1)
      TPE(IJ)=TPEAK(IJ,1)
      UMAX(IJ)=RMAX
      IF(NPUNCH)GO TO 585
      WRITE(7,561)ANAME(I),ISTEP,U(1),U(ISTEP),RMAX
      WRITE(7,1000)((U(J),Y(J)),J=1,ISTEP)
  585 CONTINUE
      I2=1
      CALL  PLOT1(ANAME(I),IC1,I2,ISTEP,RMAX,WRITEX)
      WRITE(17)ENAME(1),ISTEP
      WRITE(17)((U(J),Y(J)),J=1,ISTEP)
      GO TO 599
  590 CONTINUE
      WRITE(6,591)ANAME(I),TMIN,TMAX
  591 FORMAT('0WHILE EXECUTING PEAK COMBINATION LOOP OF MAIN PROGRAM NO
     *PEAKS WHERE FOUND FOR ',A6    /
     *  '    TMIN= ',1PE12.4,'  TMAX= ',E12.4)
  599 CONTINUE
      REWIND13
  600 CONTINUE
C
C     END OF PEAK COMBINATION LOOP
C
      ENDFILE 17
      BACKSPACE 17
      REWIND 17
C
C     SUMMARY PRINTOUT LOOP
C
      WRITE(6,13000)
13000 FORMAT('1SUMMARY OF NUCLIDE CONTRIBUTIONS'/1X,34('*')//
     *       ' NUCLIDE',6(4X,'TPEAK',5X,'CMAX',2X)/)
      DO 14000 J=1,ISL
```

```
      IF(ICONTR(J).EQ.0) GO TO 14000
      NN=ICONTR(J)
      WRITE(6,13100)ANAME(J),(TPEAK(J,K),CMAX(J,K),K=1,NN)
13100 FORMAT((1X,A7,1PE9.2,E9.2,2X,5(2E9.2,2X))/8X,4(2E9.2,2X))
14000 CONTINUE
      WRITE(6,15000)
15000 FORMAT('1SUMMARY OF NUCLIDE INFLOWS AND APPROXIMATIVE DOSES'/1X,
     *50('*')//' NUCLIDE',4X,'TPEAK',7X,'CMAX',8X,'DOSEMAX'//)
      DO 16000 J=1,ISL
      NN=ICONTR(J)
      DOSMAX(J)=UMAX(J)*DOSE2(J)
      IF(NN.EQ.0)GO TO 16000
      WRITE(6,15100)ANAME(J),TPE(J),UMAX(J),DOSMAX(J)
15100 FORMAT(1X,A7,1P3E12.2)
16000 CONTINUE
C
C     END OF SUMMARY PRINTOUT LOOP
C
      TMATOT=0.0
C
C     CALCULATION OF DOSES
C
      DO 711 I=1,ISL
      IF(TMA(I).GT.TMATOT.AND.TMA(I).GT.10.0)TMATOT=TMA(I)
  711 CONTINUE
      TMITOT=TMATOT
      DO 712 I=1,ISL
      IF(TMI(I).LT.TMITOT.AND.TMI(I).GT.10.0)TMITOT=TMI(I)
  712 CONTINUE
      CALL TSCALE(DCSMAX,TPE,ISL,TMATOT,TMITOT,U,IS,&850)
      DO 701 I=1,IS
      Y(I)=0.0
  701 CONTINUE
  710 READ(17,END=800)NNN,ISTEP
      READ(17)(TIME(I),DOSE1(I),I=1,ISTEP)
      IJ=INTNUK(NNN)
      DO 750 I=1,ISTEP
      DOSE1(I)=DOSE1(I)*DOSE2(IJ)
  750 CONTINUE
      CALL INTPOL(IS,ISTEP)
      GO TO 710
  800 CONTINUE
      RMAX=0.0
      DO 801 I=1,IS
      IF(Y(I).GT.RMAX)RMAX=Y(I)
  801 CONTINUE
      CALL PLOT1(IDOS,IC1,0,IS,RMAX,WRITEX)
      RETURN
  850 CONTINUE
      WRITE(6,860)
  860 FORMAT('0WHILE EXECUTING PEAK COMBINATION LOOP OF MAIN PROGRAM NO
     *DOSES WHERE FOUND')
      RETURN
      END
```

```
      SUBROUTINE DECAY(IPRINT)
C
```

```
C     THIS ROUTINE IS FOR CALCULATING THE NUCLIDE INVENTORY AT THE
C     TIME OF LEACH INCIDENT, WITH OR WITHOUT REPROCESSING
C
      REAL*16 TEXP1,QLOG
      REAL*16 ALN2,T,THALV(75),CNOLL(75),ALAMB(75),ANNOLL(75)
      COMMON/HALVA/ THALV,RET(75)
      REAL*8 SLUT,BLANK
      DIMENSION IDENT(40),IFISS(40),ICHAIN(6,3,4),JCHAIN(6,3,4)
      REAL*8 FISS(40),CHAIN(6,3,4)
      COMMON/DEY1/ FISS,CHAIN,IFI,ICH
      REAL*16 ETEXP(75),AN(75),C(75)
      REAL*8 ANAME(76)
      COMMON/DEY2/ ETEXP,AN,C,ANAME,SF(75),NUK(75),IN,ISL
      REAL*16 AL(6),ET(6),ANN(6)
      COMMON/DEY3/ AL,ET,ANN
      DATA SLUT/'-1     '/, BLANK/'      '/, JCHAIN/72*1/
      DATA ALAMB/75*0./,ANNOLL/75*0./
      DATA ALN2/.6931471806Q00/
  800 FORMAT(A6,2E10.2)
  810 FORMAT(20A4)
  900 FORMAT('1INITIATION OF DECAY WITH'//1X,20A4/1X,20A4//
     1     15X,'NUCL. ',3X,'THALF',7X,'C0    ',7X,'ALAMB',7X,'ANZERO'//
     2     75(I11,4X,A6,4(2X,1PE10.3)/))
  910 FORMAT(/' FISSION PRODUCTS:'//30(I11,4X,A6/))
  920 FORMAT(/' DECAY CHAINS:'//4(I11,4X,3(A6,3X)/
     1        5(15X,A6,3X,A6,3X,A6/)//))
  930 FORMAT('1CHANGED ACTIVITIES AT THE TIME OF REPROCESSING=',
     1 F14.3 ,' YR'///' NR  FNR   NUKLID    SF',8X,'CZERO',8X,'ANZERO'/)
  940 FORMAT(1X,I2,3X,I2,3X,A6,3X,F6.4,2(3X,1PE10.3))
  950 FORMAT('1ACTIVITIES AT TIME=',F14.3,' YEARS'///
     1        ' NR  FNR   NUCLIDE   CURIE'/)
  960 FORMAT(1X,I2,3X,I2,3X,A6,2X,1PE10.3)
  996 FORMAT('+',T36,A6,' NOT IN  DATASET, ICHAIN=',3(I2,1X),'***'//
     1        75(10X,A6/))
  997 FORMAT('+',T36,A6,' NOT IN  DATASET, IFISS=',I2,' ***'//
     1        75(10X,A6/))
  998 FORMAT('+',T36,' NO ENDMARK IN DATASET ***'//75(10X,A6/))
  999 FORMAT('1'/////10X,'*** INITIATION OF DECAY:')
      DO 10 I=1,76
      READ(1,800) ANAME(I),THALV(I),CNOLL(I)
      IF (ANAME(I).EQ.SLUT) GO TO 20
   10 CONTINUE
      WRITE(6,999)
      WRITE(6,998) ANAME
      STOP
   20 READ(1,810) IDENT
      IBL=I
      ANAME(IBL)=BLANK
      ISL=IBL-1
      DO 50 I=1,IFI
      DO 40 IX=1,ISL
      IF (FISS(I).NE.ANAME(IX)) GO TO 40
      IFISS(I)=IX
      NUK(I)=IX
      GO TO 50
   40 CONTINUE
```

```
      WRITE(6,999)
      WRITE(6,997) FISS(I),I,(ANAME(IX),IX=1,ISL)
      STOP
 50   CONTINUE
      IN=IFI
      DO 100 I=1,ICH
      DO 100 K=1,6
      DO 70 J=1,3
      DO 60 IX=1,IBL
      IXX=IBL-IX+1
      IF (CHAIN(K,J,I).NE.ANAME(IXX)) GO TO 60
      ICHAIN(K,J,I)=IXX
      IF (IXX.EQ.IBL) GO TO 70
      DO 55 IY=IFI,IN
      IF (NUK(IY).EQ.IXX) GO TO 70
 55   CONTINUE
      IN=IN+1
      NUK(IN)=IXX
      GO TO 70
 60   CONTINUE
      WRITE(6,999)
      WRITE(6,996) CHAIN(K,J,I),K,J,I,(ANAME(IX),IX=1,ISL)
      STOP
 70   CONTINUE
      DO 90 J=1,2
      DO 80 JX=J,3
      IF (J.EQ.JX) GO TO 80
      IF (ICHAIN(K,J,I).NE.ICHAIN(K,JX,I)) GO TO 80
      JCHAIN(K,J,I)=JCHAIN(K,J,I)+1
      JCHAIN(K,JX,I)=JCHAIN(K,JX,I)+1
 80   CONTINUE
 90   CONTINUE
 100  CONTINUE
      DO 105 I=1,IN
      IND=NUK(I)
      ALAMB(IND)=ALN2/THALV(IND)
      ANNOLL(IND)=CNOLL(IND)/ALAMB(IND)
 105  CONTINUE
      IF (IPRINT.NE.1) RETURN
      WRITE( 6,900) IDENT,(I,ANAME(I),THALV(I),CNOLL(I),ALAMB(I),
     1              ANNOLL(I),I=1,ISL)
      WRITE( 6,910) (I,FISS(I),I=1,IFI)
      WRITE( 6,920) (I,((CHAIN(K,J,I),J=1,3),K=1,6),I=1,ICH)
      RETURN
      ENTRY DEC(T,TX,IUPP,ISTEG,IPRINT)
      DO 110 I=1,ISL
      C(I)=0.
      AN(I)=0.
      ETEXP(I)=TEXP1(-T*ALAMB(I))
 110  CONTINUE
      DO 120 I=1,IFI
      INDEX=IFISS(I)
      AN(INDEX)=ANNOLL(INDEX)*ETEXP(INDEX)
 120  CONTINUE
      DO 170 I=1,ICH
      DO 160 J=1,3
```

```
      DO 125 K=1,6
      AL(K)=0.
      ET(K)=0.
      ANN(K)=0.
125   CONTINUE
      DO 130 K=1,6
      IF (ICHAIN(K,J,I).EQ.IBL) GO TO 130
      KST=K
      GO TO 140
130   CONTINUE
      GO TO 160
140   DO 150 K=KST,6
      KX=K-KST+1
      INDEX=ICHAIN(K,J,I)
      AL(KX)=ALAMB(INDEX)
      ET(KX)=ETEXP(INDEX)
      ANN(KX)=ANNOLL(INDEX)/FLOAT(JCHAIN(K,J,I))
      AN(INDEX)=AN(INDEX)+BATE1(KX,ANAME(INDEX),T)
150   CONTINUE
160   CONTINUE
170   CONTINUE
      DO 180 I=1,IN
      IND=NUK(I)
      C(IND)=AN(IND)*ALAMB(IND)
180   CONTINUE
      IF (IUPP.EQ.1) GO TO 186
      IF (IPRINT.EQ.0) RETURN
      WRITE( 6,950) TX
      DO 185 I=1,IN
      IND=NUK(I)
      WRITE( 6,960) I,IND,ANAME(IND),C(IND)
185   CONTINUE
      RETURN
186   DO 190 I=1,IN
      IND=NUK(I)
      CNOLL(IND)=SF(IND)*C(IND)
      ANNOLL(IND)=CNOLL(IND)/ALAMB(IND)
190   CONTINUE
      IF (IPRINT.EQ.0) RETURN
      WRITE( 6,930) TX
      DO 200 I=1,IN
      IND=NUK(I)
      WRITE( 6,940) I,IND,ANAME(IND),SF(IND),CNOLL(IND),ANNOLL(IND)
200   CONTINUE
      RETURN
      END
      FUNCTION BATE1(INR,ANAME,T)
C
C     THIS ROUTINE CALCULATES THE DECAY OF THE INITIAL NUCLIDE INVENTORY
C
      REAL*16 TERM1,BATSUM,T
      REAL*8 ANAME
      REAL*16 AL(6),ET(6),ANN(6)
      COMMON/DEY3/ AL,ET,ANN
      REAL*16 FAKT(6),TERM(6,6),DUMMY(42)
      EQUIVALENCE (FAKT(1),DUMMY(1)),(TERM(1),DUMMY(7))
```

```
  900 FORMAT(///1X,A6,10X,'BATE=',1PE14.7//' AL=   ',6(3X,E14.7)/' ET=   '
     1      ,6(3X,E14.7)/' ANN= ',6(3X,E14.7)//' FAKT=',6(3X,E14.7)//
     2      ' TERM=',6(3X,E14.7)/5(9X,E14.7,5(3X,E14.7)/))
  910 FORMAT(1X,I5,2X,3(Q37.30))
  920 FORMAT(//' *** TERM1=',1PE14.7,'  I=',I1,'  T=',0PF14.3,3X,A6//)
  930 FORMAT('1 ')
  999 FORMAT(/////10X,'*** BATE HAR ANROPATS MED',A6,' OCH INR=',
     1      I2,' ***')
      IF (INR.GT.0.AND.INR.LT.7) GC TC 10
      WRITE(6,999) ANAME,INR
      STOP
   10 BATSUM=0.
         DO 20 I=1,42
         DUMMY(I)=0.
   20    CONTINUE
         DO 70 I=1,INR
         FAKT(I)=ANN(I)
         IXSL=I-1
         IF (IXSL.EQ.0) GO TO 40
            DO 30 IX=1,IXSL
            FAKT(IX)=FAKT(IX)*AL(IXSL)
   30       CONTINUE
   40       DO 60 J=I,INR
            TERM(I,J)=ET(J)
               DC 50 JX=I,INR
               IF (J.NE.JX) TERM(I,J)=TERM(I,J)/(AL(JX)-AL(J))
   50          CONTINUE
   60       CONTINUE
   70    CONTINUE
         DO 90 I=1,INR
         TERM1=0.
            DO 80 J=I,INR
            TERM1=TERM1+TERM(I,J)
   80       CONTINUE
         BATSUM=BATSUM+FAKT(I)*TERM1
   90    CONTINUE
         BATE1=BATSUM
      RETURN
      END
      SUBROUTINE SINGLE(TH1,III,AKT,NS)
C
C     THIS ROUTINE CALCULATES THE RELEASE RATES FOR  SINGLE-NUCLIDES
C
      COMMON/PRINT/  MIGOUT
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
     *              GAP(6),TO,DUR,PE,ETA,TD,V0,PATH
      COMMON/C2/    NSTEP,BRD(304),BRND(304),TIME(304),THETA(304),
     *              TBEGIN,TFIN,TTOPP,BRPEAK,DINO
      COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
     *               CMAX(75,10),DOSE2(75)
      COMMON /TMAXND/ TND1(75),TND2(75)
      REAL*8 ENAME,B1,B5,B2,B6,AM1,G1,G2,G3,B9,TDERFC
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,TO,DUR,PE,ETA,TD,V0,PATH
      LOGICAL AKT,MIGOUT
      J1=0
      J2=0
```

```
      R=1.0/EQ(1)
      V1=V0*R
      IPSEUD=0
      DNR(1)=ALAMB(1)*PATH/V0
      AN1=AQ(1)
      BRD(1)=0.0
      BRD(2)=0.0
      AL1=ALAMB(1)
      G3=V1*DUR/PATH
      GAP(1)=AQ(1)/AL1
      GAP(4)=AQ(1)*TEXP(-AL1*DUR)/AL1
      IF(III.LT.2)GO TO 20
      CALL VECTOR(III,IPSEUD)
   20 CONTINUE
      TFIN=T0+PATH/V1*(2-TH1)+DUR
      T1=TH1*PATH/V1
      TBEGIN=T0+T1
      TTOPP=T0+T1
      TTOPND=PATH/V1+T0
      NSTEP=0.0
      BRPEAK=0.0
      PEAKND=0.0
      DINO=0.0
      IF(AQ(1).LT.1.E-15)GO TO 3740
      IF((TBEGIN-T0)*ALAMB(1).GT.40.0)GO TO 3740
      IF(AKT)GO TO 370
      S3=(TFIN/TBEGIN)**(1.0/FLOAT(NS-1))
      TIME(1)=TBEGIN
      THETA(1)=T1*V0/PATH
      I=2
  350 CONTINUE
      TIME(I)=S3*TIME(I-1)
      IF(TIME(I).LT.TTOPND.OR.TIME(I-1).GE.TTOPND)GO TO 360
      TIME(I)=TTOPND
  360 CONTINUE
      THETA(I)=(TIME(I)-T0)*V0/PATH
      I=I+1
      IF(I.GT.NS) GO TO 390
      GO TO 350
  370 S1=(TFIN-TBEGIN)/FLOAT(NS)
      DO 380 I=2,NS
      TIME(I)=TIME(I-1)+S1
  380 THETA(I)=(TIME(I)-T0)*V0/PATH
  390 CONTINUE
      DO 750 J=1,NS
      G2=THETA(J)*R
      G1=G2-1.0D0
      BL1=AL1*(TIME(J)-T0)
      B1=0.5*DSQRT(PE/G2)*G1
      B9=G2-G3
      IF(B9)540,540,550
  540 B9=1.D-20
  550 B2=0.5*DSQRT(PE/B9)*(G1-G3)
      B5=.5*DSQRT(PE/G2)*(G1+2.0D0)
      B6=.5*DSQRT(PE/B9)*(G1+2.0D0-G3)
      AM1=TDERFC(-B1)+TEXP(PE)*TDERFC(B5)
```

```
      IF(G2.GT.G3)AM1=AM1-(TDERFC(-B2)+TEXP(PE)*TDERFC(B6))
      F1=AM1/2.0
      IF(III-1)621,621,622
  621 CONTINUE
      IF(1.E50*F1/DUR*AN1.LE.1.E-03)GO TO 619
      BRD(J)=TEXP(DLOG(F1*AN1/DUR)-BL1)
      GO TO 625
  622 BRD(J)=BATE(TIME(J),T0,III)/DUR*F1
      GO TO 625
  619 CONTINUE
      BRD(J)=0.0
  625 CONTINUE
      IF(G1.LT.0.0.OR.G1-G3.GT.0.0)GO TO 700
      IF(III-1)680,680,685
  680 CONTINUE
      BRND(J)=AN1/DUR*TEXP(-BL1)
      GO TO 710
  685 BRND(J)=BATE(TIME(J),T0,III)/DUR
      GO TO 710
  700 BRND(J)=0.0
  710 IF(BRD(J-1).LT.1.E-15.AND.BRD(J).GE.1.E-15.AND.J1.EQ.0)J1=J
      IF(BRD(J-1).GE.1.E-15.AND.BRD(J).LT.1.E-15)J2=J-1
      IF(BRD(J).LE.BRPEAK)GO TO 720
      BRPEAK=BRD(J)
      TTOPP=TIME(J)
  720 CONTINUE
      IF(BRND(J).LE.PEAKND)GO TO 730
      PEAKND=BRND(J)
  730 CONTINUE
      IF(J2.GT.0)GO TO 760
  750 CONTINUE
  760 CONTINUE
 3282 IF(PEAKND)3285,3285,3283
 3283 DINO =BRPEAK/PEAKND
      GO TO 3286
 3285 DINO =0.0
 3286 CONTINUE
      IF(J1.EQ.0.AND.J2.EQ.0)GO TO 3740
      IF(J1.GT.0.AND.J2.EQ.0)J2=NS
      IF(J1.EQ.0)J1=1
      TBEGIN=TIME(J1)
      TFIN=TIME(J2)
      NSTEP=J2-J1+1
      IF(J1.EQ.1)GO TO 3740
      DO 3710 I=1,NSTEP
      II=J1+I-1
      TIME(I)=TIME(II)
      THETA(I)=THETA(II)
      BRD(I)=BRD(II)
      BRND(I)=BRND(II)
 3710 CONTINUE
 3740 CONTINUE
      ID=III+12
      WRITE(ID)ENAME,NSTEP
      IF(NSTEP.LE.0)GO TO 3750
      WRITE(ID)(TIME(I),BRD(I),I=1,NSTEP)
```

```
 3750 IJ=INTNUK(ENAME(III))
      ICONTR(IJ)=ICONTR(IJ)+1
      NN=ICONTR(IJ)
      TSTART(IJ,NN)=TBEGIN
      TEND(IJ,NN)=TFIN
      TPEAK(IJ,NN)=TTOPP
      CMAX(IJ,NN)=BRPEAK
      IF(.NOT.MIGOUT)RETURN
      TND1(IJ)=TTOPND
      TND2(IJ)=TTOPND+DUR
      NCHAIN=III
      CALL PRINTX(NCHAIN)
      RETURN
      END
      SUBROUTINE DOUBLE(INS,SUB,J3)
C
C     THIS ROUTINE CALCULATES THE RELEASE RTTES FOR TWO-NUCLIDE CHAINS
C
      COMMON/PRINT/ MIGOUT
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
     *              GAP(6),T0,DUR,PE,ETA,TD,V0,PATH
      COMMON/C2/    IS,BAND(304),BANDND(304),TIME(304),THETA(304),
     *              TBEGIN,TFIN,TTOPP,BMAX,DINO
      COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
     *               CMAX(75,10),DOSE2(75)
      COMMON/HALVA/  THALV(75),RET(75)
      COMMON /LGTEST/ LG(6)
      COMMON /XXX/ XX5,XX6,XX7,XX19,XX29,XX39
      REAL*16 THALV
      REAL*8 XX5,XX6,XX7,XX19,XX29,XX39
      REAL*8 SWD1,SWD2,SWD3,SNOD1,SNOD2,SNOD3
      REAL*8 ENAME,PS1,PS2,PS3
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,T0,DUR,PE,ETA,TD,V0,PATH
      LOGICAL MIGOUT,INSCON,INS,SUB
      LOGICAL LDISP,LNDISP
      LOGICAL LG
      DIMENSION Q(15)
      DATA PS1/'U 238'/,PS2/'TH230'/,PS3/'U 234'/
      NCHAIN=2
      INSCON=INS
      D9=0.0
      DINO=0.0
      IPSEUD=0
      AK=DMAX1(EQ(1),EQ(2))
      AL=DMIN1(EQ(1),EQ(2))
      BMAX=0.0
      IF(ENAME(1).NE.PS1.OR.ENAME(2).NE.PS2)GO TO 100
      IPSEUD=1
  100 CONTINUE
      CALL VECTOR(NCHAIN,IPSEUD)
      CALL TLIMIT(ENAME,NCHAIN,Q,ICOND)
      IF(ICOND)641,641,9999
  641 CONTINUE
      CALL TIME23(NCHAIN,Q,J3)
  200 CONTINUE
      INS=.TRUE.
```

```
      LG(1)=.TRUE.
      LG(2)=.TRUE.
      LDISP=.TRUE.
      LNDISP=.TRUE.
      XX5=DUR*(V0/PATH*(EQ(1)*DNR(1)-EQ(2)*DNR(2))/(EQ(1)-EQ(2)))
      XX6=0.
      XX7=0.
      XX19=DUR*(DNR(1)*V0/PATH)
      XX29=DUR*(DNR(2)*V0/PATH-ALAMB(1))
      XX39=0.
      G1=4*EQ(2)*(DNR(2)-DNR(1))/PE
      G2=4.D0*EQ(1)*EQ(2)*(DNR(2)-DNR(1))/(PE*(EQ(1)-EQ(2)))
      IF(G1.LT.-1.)LG(1)=.FALSE.
      IF(G2.LT.-1.)LG(2)=.FALSE.
      IF(.NOT.LG(1).OR..NOT.LG(2))INS=.FALSE.
      CALL FNCD2
      IF(LDISP)CALL FWD2
      DO 1060 J=1,IS
      TD=THETA(J)
      DSM=AK*ETA+DUR*V0/PATH
      IF(1.-DSM/TD.GT.1.E-7)LNDISP=.FALSE.
      IF(.NOT.LNDISP)SNOD2=0.0
      IF(LDISP)CALL BWD2(SWD1,SWD2,SWD3,2,J)
      IF(LNDISP)CALL BNOD2(SNOD1,SNOD2,SNOD3,2,J)
      IF(.NOT.LDISP)SWD2=SNOD2
      SWD2=SWD2*ALAMB(2)
      SNOD2=SNOD2*ALAMB(2)
      IF(IPSEUD-2)1040,1050,1050
 1040 CONTINUE
      BAND(J)=SWD2
      BANDND(J)=SNOD2
      GO TO 1055
 1050 CONTINUE
      BAND(J)=BAND(J)-SWD2
      BANDND(J)=BANDND(J)-SNOD2
 1055 CONTINUE
      IF(TD.LT.AL.OR.TD.GT.AK+DUR*V0/PATH)BANDND(J)=0.0
 1060 CONTINUE
      IF(IPSEUD-1)1100,1070,1090
 1070 CONTINUE
      IJ=INTNUK(PS3)
      EQ(3)=EQ(1)
      THALF(3)=THALF(1)
      DNR(3)=DNR(1)
      SWAP1=GAP(1)
      SWAP2=GAP(4)
      EQ(1)=RET(IJ)
      THALF(1)=THALV(IJ)
      ALAMB(1)=0.69314718/THALF(1)
      IPSEUD=2
      CALL VECTOR(NCHAIN,IPSEUD)
      GO TO 200
 1090 CONTINUE
      EQ(1)=EQ(3)
      THALF(1)=THALF(3)
      DNR(1)=DNR(3)
```

```
      GAP(1)=SWAP1
      GAP(4)=SWAP2
 1100 CONTINUE
      CALL TOPDEF
      IF(MIGOUT) CALL PRINTX(NCHAIN)
      WRITE(14)(ENAME(I),I=1,3),IS
      IF(IS.EQ.0)GO TO 9999
      WRITE(14)(TIME(I),BAND(I),I=1,IS)
      IJ=INTNUK(ENAME(2))
      ICONTR(IJ)=ICONTR(IJ)+1
      NN=ICONTR(IJ)
      TSTART(IJ,NN)=TBEGIN
      TEND(IJ,NN)=TFIN
      TPEAK(IJ,NN)=TTOPP
      CMAX(IJ,NN)=BMAX
 9999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRIPLE(INS,J3)
C
C     THIS ROUTINE CALCULATES THE RELEASE RATES FOR THREE-NUCLIDE CHAINS
C
      COMMON/PRINT/ MIGOUT
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
     *              GAP(6),T0,DUR,PE,ETA,TD,V0,PATH
      COMMON/C2/    IS,BAND(304),BANDND(304),TIME(304),THETA(304),
     *              TBEGIN,TFIN,TTOPP,BMAX,DINO
      COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
     *               CMAX(75,10),DOSE2(75)
      COMMON/HALVA/  THALV(75),RET(75)
      COMMON /XXX/ XX5,XX6,XX7,XX19,XX29,XX39
      COMMON /DIV/ DSW,DSN
      COMMON /LGTEST/ LG(6)
      REAL*16 THALV
      REAL*8 SWD1,SWD2,SWD3,SNOD1,SNOD2,SNOD3
      REAL*8 XX5,XX6,XX7,XX19,XX29,XX39
      REAL*8 DSW,DSN
      REAL*8 ENAME,PS1,PS2,PS3,PS4,SWAP1,Y
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,T0,DUR,PE,ETA,TD,V0,PATH
      LOGICAL MIGOUT,INS,INSCON
      LOGICAL LDISP,LNDISP
      LOGICAL LG
      DATA PS1/'U 238'/,PS2/'TH230'/,PS3/'RA226'/,PS4/'U 234'/
      DIMENSION Q(15)
      NCHAIN=3
      BMAX=0.0
      DINO=0.0
      IPSEUD=0
      AK=DMAX1(EQ(1),EQ(2),EQ(3))
      AL=DMIN1(EQ(1),EQ(2),EQ(3))
      IF(ENAME(1).NE.PS1.OR.ENAME(2).NE.PS2.OR.ENAME(3).NE.PS3)GO TO 100
      IPSEUD=1
  100 CONTINUE
      CALL VECTOR(NCHAIN,IPSEUD)
      CALL TLIMIT(ENAME,NCHAIN,Q,ICOND)
      IF(ICOND)641,641,9999
```

```
  641 CONTINUE
      CALL TIME23(NCHAIN,Q,J3)
  200 CONTINUE
      INS=.TRUE.
      LDISP=.TRUE.
      LNDISP=.TRUE.
      XX5=DUR*(V0/PATH*(EQ(1)*DNR(1)-EQ(2)*DNR(2))/(EQ(1)-EQ(2)))
      XX6=DUR*(V0/PATH*(EQ(2)*DNR(2)-EQ(3)*DNR(3))/(EQ(2)-EQ(3)))
      XX7=DUR*(V0/PATH*(EQ(1)*DNR(1)-EQ(3)*DNR(3))/(EQ(1)-EQ(3)))
      XX19=DUR*(DNR(1)*V0/PATH)
      XX29=DUR*(DNR(2)*V0/PATH)
      XX39=DUR*(DNR(3)*V0/PATH)
      DO 250 I=1,6
      LG(I)=.TRUE.
  250 CONTINUE
      CALL G1(1,2,3,LG(1),&300)
      CALL G3(1,2,3,LG(1),&300)
      CALL G2(1,2,LG(1))
  300 CALL G2(1,3,LG(2))
      CALL G2(2,3,LG(3))
      CALL G5(1,2,LG(4))
      CALL G5(1,3,LG(5))
      CALL G5(2,3,LG(6))
      DO 350 I=1,6
      IF(LG(I))GO TO 350
      INS=.FALSE.
  350 CONTINUE
      CALL FNOD3
      DO 1060 J=1,IS
      TD=THETA(J)
      DSM=AK*ETA+DUR*V0/PATH
      IF(1.-DSM/TD.GT.1.E-7)LNDISP=.FALSE.
      IF(.NOT.LNDISP)SNOD3=0.0
      IF(LDISP)CALL BWD3(SWD1,SWD2,SWD3,3,J)
      IF(LNDISP)CALL BNOD3(SNOD1,SNOD2,SNOD3,3,J)
      IF(.NOT.LDISP)SWD3=SNOD3
      BRIID=SWD3*ALAMB(3)
      BRND=SNOD3*ALAMB(3)
      IF(IPSEUD-2)1040,1050,1050
 1040 CONTINUE
      BAND(J)=BRIID
      BANDND(J)=BRND
      GO TO 1060
 1050 CONTINUE
      BAND(J)=BAND(J)-BRIID
      BANDND(J)=BANDND(J)-BRND
 1060 CONTINUE
      IF(IPSEUD-1)1100,1070,1090
 1070 CONTINUE
      IJ=INTNUK(PS4)
      SWAP2=DNR(1)
      SWAP5=GAP(1)
      SWAP6=GAP(4)
      SWAP3=THALF(1)
      SWAP4=EQ(1)
      THALF(1)=THALV(IJ)
```

```
      ALAMB(1)=0.69314718/THALF(1)
      EQ(1)=RET(IJ)
      IPSEUD=2
      CALL VECTOR(NCHAIN,IPSEUD)
      GO TO 200
 1090 CONTINUE
      DNR(1)=SWAP2
      GAP(1)=SWAP5
      GAP(4)=SWAP6
      THALF(1)=SWAP3
      EQ(1)=SWAP4
 1100 CONTINUE
      CALL TOPDEF
      IF(MIGOUT) CALL PRINTX(NCHAIN)
      WRITE(15)(ENAME(I),I=1,3),IS
      IF(IS.EQ.0)GO TO 9999
      WRITE(15)(TIME(I),BAND(I),I=1,IS)
      IJ=INTNUK(ENAME(3))
      ICONTR(IJ)=ICONTR(IJ)+1
      NN=ICONTR(IJ)
      TSTART(IJ,NN)=TBEGIN
      TEND(IJ,NN)=TFIN
      TPEAK(IJ,NN)=TTOPP
      CMAX(IJ,NN)=BMAX
 9999 RETURN
      END
      SUBROUTINE TSING(PE,TH1)
C
C     THIS ROUTINE  CALCULATES THE START POINT FOR SINGLE-NUCLIDE
C     RELEASE WITH DISPERSION
C
      REAL*8 PE
      DATA PI/3.14159/,EP/1.E-12/,EPS/1.E-15/
      FUNK(TH,PE)=0.5*TDERFC(0.5*(DSQRT(PE/TH)-DSQRT(PE*TH))+TEXP(PE)*
     *TDERFC(0.5*DSQRT(PE/TH)+DSQRT(PE*TH)))
      DER(TH,PE)=0.5*DSQRT(PE/PI/TH**3)*TEXP(-PE/4./TH*(1-TH)**2)
      AK=1
      BK=1.E-5
   10 CONTINUE
      AM=0.5*(AK+BK)
      F=FUNK(AM,PE)
      IF(F.LT.EPS)GO TO 20
      IF(F.GT.EP)GO TO 30
      THE=AM
      GO TO 40
   20 CONTINUE
      BK=AM
      GO TO 10
   30 CONTINUE
      AK=AM
      GO TO 10
   40 CONTINUE
      A=FUNK(THE,PE)
      IF(A.LT.EPS)GO TO 50
      B=DER(THE,PE)
      THE=THE-A/B
```

```
      GO TO 40
   50 CONTINUE
      TH1=THE
      RETURN
      END
      SUBROUTINE TIME23(NCHAIN,Q,J3)
C
C     THIS ROUTINE CALCULATES A TIME VECTOR FOR DOUBLE AND TRIPLE.
C     WITHIN THE LIMITS GIVEN BY TLIMIT
C
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
     *              GAP(6),T0,T1,PE,ETA,TD,V,AL
      COMMON/C2/    IB,BAND(304),BANDND(304),T(304),TDL(304),
     *              TBEGIN,TFIN,TTOPP,BRPEAK,DINO
      REAL*8 ENAME
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,T0,T1,PE,ETA,TD,V,AL
      DIMENSION Q(15)
      II=5*NCHAIN
      N=II-1
      DO 400 JC=1,N
      NQ=JC
      DO 300 J=JC,II
      IF(Q(J).LT.Q(NQ))NQ=J
  300 CONTINUE
      SWAP=Q(NQ)
      Q(NQ)=Q(JC)
      Q(JC)=SWAP
  400 CONTINUE
      J1=0
      IF(Q(1).GT.0.0)J1=1
      DO 500 N=2,II
      IF(Q(N).EQ.0.0.OR.Q(N).EQ.Q(N-1))GO TO 500
      J1=J1+1
      Q(J1)=Q(N)
  500 CONTINUE
      IB=1
      DO 700 JC=2,J1
      STEP=(Q(JC)-Q(JC-1))/J3
      T(IB)=Q(JC-1)
      TDL(IB)=(T(IB)-T0)*V/AL
      IB=IB+1
      DO 600 J=2,J3
      T(IB)=T(IB-1)+STEP
      TDL(IB)=(T(IB)-T0)*V/AL
      IB=IB+1
  600 CONTINUE
  700 CONTINUE
      IB=IB-1
      RETURN
      END
      SUBROUTINE TLIMIT(ENAME,NCHAIN,Q,ICOND)
C
C     THIS ROUTINE DEFINES THE LIMITS BETWEEN WHICH THE TIMESCALE FOR
C     THE CALCULATION OF DOUBLE AND TRIPLE CHAINS ARE TO BE LAID OUT.
C
      DIMENSION Q(15),IJ(3)
```

```
      REAL*8 ENAME(3)
      COMMON/PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
     *                  CMAX(75,10),DOSE2(75)
      COMMON /TMAXND/TND1(75),TND2(75)
C
      ICOND=0
C
      DO 10 I=1,NCHAIN
      IJ(I)=INTNUK(ENAME(I))
      IF(ICONTR(IJ(I)).GT.0)GO TO 10
      WRITE(6,7)(ENAME(J),J=1,2)
      IF(NCHAIN.EQ.3)WRITE(6,8)ENAME(3)
      WRITE(6,9)ENAME(I)
      ICOND=1
   10 CONTINUE
C
    7 FORMAT('0WHILE EXECUTING TLIMIT FOR ',2A8)
    8 FORMAT('+',43X,A8)
    9 FORMAT(' NO PEAK WAS FOUND FOR ',A6)
C
      IF(ICOND.EQ.1)RETURN
C
      DO 20 I=1,NCHAIN
      Q(5*I-4)=TSTART(IJ(I),1)
      Q(5*I-3)=TND1(IJ(I))
      Q(5*I-2)=TPEAK(IJ(I),1)
      Q(5*I-1)=TND2(IJ(I))
      Q(5*I)=TEND(IJ(I),1)
   20 CONTINUE
C
      RETURN
      END
      SUBROUTINE VECTOR(NCHAIN,IPSEUD)
C
C     THIS ROUTINE IS USED BY DOUBLE AND TRIPLE FOR THE
C     CALCULATION OF THE DECAY CONSTANTS
C
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DECAY(3),
     *                 GAP(6),T0,T1,PE,ETA,TD,V,AL
      REAL*8 ENAME
      REAL*8 THALF,AQ,ALAMB,EQ,DECAY,GAP,T0,T1,PE,ETA,TD,V,AL
      SLASK=0.69314718
      SLASK1=SLASK*AL/V
      DO 100 I=1,NCHAIN
      DECAY(I)=SLASK1/THALF(I)
  100 CONTINUE
      X1=TEXP(-ALAMB(1)*T1)
      X2=TEXP(-ALAMB(2)*T1)
      IF(NCHAIN.EQ.3)X3=TEXP(-ALAMB(3)*T1)
      GAP(1)=AQ(1)/ALAMB(1)
      IF(IPSEUD.GE.1)GAP(1)=GAP(1)/2.44E5/(1.0/2.44E5-1.0/4.47E9)
      GAP(2)=AQ(2)/ALAMB(2)
      GAP(3)=0.0
      IF(NCHAIN.EQ.3)GAP(3)=AQ(3)/ALAMB(3)
      GAP(4)=GAP(1)*X1
      GAP(5)=GAP(2)*X2+GAP(1)*ALAMB(1)/(ALAMB(2)-ALAMB(1)
```

```
     *)*(X1-X2)
      IF(NCHAIN.EQ.3)GO TO 200
      GAP(6)=0.0
      GO TO 300
  200 GAP(6)=GAP(3)*X3+GAP(2)*ALAMB(2)/(ALAMB(3)-ALAMB(2)
     *)*(X2-X3)+GAP(1)*ALAMB(1)*ALAMB(2)*
     *X1/(ALAMB(2)-ALAMB(1))/(ALAMB(3)-ALAMB(1))
      GAP(6)=GAP(6)+GAP(1)*ALAMB(1)*ALAMB(2)*X2/(ALAMB(1)
     *-ALAMB(2))/(ALAMB(3)-ALAMB(2))
      GAP(6)=GAP(6)+GAP(1)*ALAMB(1)*ALAMB(2)*X3/(ALAMB(2)
     *-ALAMB(3))/(ALAMB(1)-ALAMB(3))
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GTEST(ARGDUM)
C
C     THIS ROUTINE TEST FOR NEGATIVE SQUARE ROOT ARGUMENTS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /FFFFF/ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
     *              GAP(6),T0,DUR,PE,ETA,TD,V0,PATH
      LOGICAL LG1,LG2,LG3,LG4,LG5
      LG1=.TRUE.
      LG2=.TRUE.
      LG3=.TRUE.
      LG4=.TRUE.
      LG5=.TRUE.
      ENTRY G1(I,J,K,LG1,*)
      G1A=1.D0+4.D0*EQ(K)*DNR(K)/PE-4.*EQ(K)*(EQ(I)*DNR(I)-EQ(J)*
     *DNR(J))/(PE*(EQ(I)-EQ(J)))
      IF(G1A.GE.0.0D0)RETURN
      LG1=.FALSE.
      RETURN1
      ENTRY G2(I,J,LG2)
      G2A=1D0+4D0*EQ(I)*EQ(J)*(DNR(J)-DNR(I))/(PE*(EQ(I)-EQ(J)))
      IF(G2A.LT.0.0D0)LG2=.FALSE.
      RETURN
      ENTRY G3(I,J,K,LG3,*)
      G3A=PE/(4D0*EQ(K))+DNR(K)-(EQ(I)*DNR(I)-EQ(J)*DNR(J))/
     *(EQ(I)-EQ(J))
      IF(G3A.GE.0.0D0)RETURN
      LG3=.FALSE.
      RETURN1
      ENTRY G4(I,J,K,L,LG4)
      G4A=PE/(4D0*EQ(K))+EQ(L)*(DNR(J)-DNR(I))/(EQ(I)-EQ(J))
      IF(G4A.LT.0.0D0)LG4=.FALSE.
      RETURN
      ENTRY G5(I,J,LG5)
      G5A=1D0+4D0*EQ(J)*(DNR(J)-DNR(I))/PE
      IF(G5A.LT.0.0D0)LG5=.FALSE.
      RETURN
      END
```

```
      FUNCTION F(DUMDUM)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 NO
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),AC(3),R(3),NO(6),
```

```
     *T0,T,P,ETA,THETA,V,SCL
      F=0
      ENTRY F4(I,J,K,L)
      F4=(AC(K)*R(K)-AC(L)*R(L))/(AC(K)-AC(L))-(AC(I)*R(I)-AC(J)*R(J)) /
     *(AC(I)-AC(J))
      RETURN
C
      ENTRY F11(I,J)
      F11=NO(I)/(2.D0*T)*R(I)/(R(J)-R(I))
      RETURN
C
      ENTRY F12(I,J)
      F12=NO(I)/(2.D0*T)*R(I)*R(J)
      RETURN
C
      ENTRY F13(I,J,K)
      F13=NO(I)/(2.D0*T)*AC(J)*R(I)*R(J)/(AC(J)-AC(K))
      RETURN
C
      ENTRY F14(I,J,K,L)
      F14=NO(I)/(2.D0*T)*AC(I)*AC(J)*R(I)*R(J)/((AC(I)-AC(J))*(AC(L)-
     *AC(K)))
      RETURN
C
      ENTRY F15(I,J)
      F15=1.D0-AC(I)/AC(J)
      RETURN
C
      ENTRY F16(I,J)
      F16=AC(I)/AC(J)
      RETURN
C
      ENTRY F17(I,J)
      F17=R(J)-R(I)
      RETURN
C
      ENTRY F18(I,J,K)
      F18=(AC(J)*R(J)-AC(K)*R(K))/(AC(J)-AC(K))-R(I)
      RETURN
C
      ENTRY F34(I)
      F34=0.0D0
      IF(AC(I)*ETA/THETA-1..GT.1.D-7)RETURN
      F34=1.D0
      RETURN
      END
      SUBROUTINE FF(DUM)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 NO
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),AC(3),R(3),NO(6),
     *T0,T,P,ETA,THETA,V,SCL
      COMMON/NEGXER/  ERFCM1,EXPCM1
C
      ENTRY FF5(I,J,F5E)
      F5E=-THETA*(AC(I)*R(I)-AC(J)*R(J))/(AC(I)-AC(J))
      RETURN
```

```
C
      ENTRY FF6P(I,J,K,F6PE)
      N=1
      GO TO 50
   20 F6PE=P*ETA/2.D0*(1.D0+F6PMT)
      RETURN
C
      ENTRY FF6M(I,J,K,F6ME)
      N=2
      GO TO 50
   30 CONTINUE
      IF(G.LE.3.D-9)GO TO 35
      TERM=1.-F6PMT
      GO TO 40
   35 G=-G/4.D0
      TERM=2.D0*G*(1.D0+G*(1.D0+G*(2.D0+G*(5.D0+G*
     *(14.D0+G*(42.D0+G*132.D0))))))
   40 F6ME=P*ETA/2.D0*TERM
      RETURN
   50 CONTINUE
      G=4.D0*AC(K)*R(K)/P-4.*AC(K)*(AC(I)*R(I)-AC(J)*R(J))/(P*(AC(I)-
     *AC(J)))
      IF(G.LT.-1.D0)G=-1.D0
      F6PMT=DSQRT(1.D0+G)
      IF(N.EQ.1)GO TO 20
      IF(N.EQ.2)GO TO 30
C
      ENTRY FF7P(I,J,F7PE)
      N=1
      GO TO 100
   60 CONTINUE
      F7PE=P*ETA/2.D0*(1.D0+F7PMT)
      RETURN
C
      ENTRY FF7M(I,J,F7ME)
      N=2
      GO TO 100
   70 CONTINUE
      IF(G.LE.3.D-9) GO TO 80
      TERM=1.D0-F7PMT
      GO TO 90
   80 G=-G/4.D0
      TERM=2.D0*G*(1.D0+G*(1.D0+G*(2.D0+G*(5.D0+G*
     *(14.D0+G*(42.D0+G*132.D0))))))
   90 F7ME=P*ETA/2.D0*TERM
      RETURN
  100 CONTINUE
      G=4.D0*AC(I)*AC(J)*(R(J)-R(I))/(P*(AC(I)-AC(J)))
      IF(G.LT.-1.D0)G=-1.D0
      F7PMT=DSQRT(1.D0+G)
      IF(N.EQ.1)GO TO 60
      IF(N.EQ.2)GO TO 70
C
      ENTRY FF8P(I,J,K,F8PE,F8PA)
      N=1
      GO TO 150
```

```
  110 CONTINUE
      ARG=ETA/2.D0*DSQRT(AC(K)*P/THETA)+F8PMT
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F8PE=ERFCE
      F8PA=ERFCA
      RETURN
C
      ENTRY FF8M(I,J,K,F8ME,F8MA,F8MARG,F8MME,F8MMA)
      N=2
      GO TO 150
  120 CONTINUE
      ARG=ETA/2.D0*DSQRT(AC(K)*P/THETA)-F8PMT
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F8ME=ERFCE
      F8MA=ERFCA
      F8MARG=ARG
      F8MME=EXPCM1
      F8MMA=ERFCM1
      RETURN
C
  150 CONTINUE
      G=((P/(4.D0*AC(K))+R(K)-(AC(I)*R(I)-AC(J)*R(J))/
     *(AC(I)-AC(J)))*THETA)
      IF(G.LT.0.0D0)G=0.D0
      F8PMT=DSQRT(G)
      IF(N.EQ.1)GO TO 110
      IF(N.EQ.2)GO TO 120
C
      ENTRY FF9M(I,J,K,L,F9ME,F9MA,F9MARG,F9MME,F9MMA)
      N=1
      GO TO 200
  160 CONTINUE
      ARG=ETA/2.D0*DSQRT(AC(K)*P/THETA)-F9PMT
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F9ME=ERFCE
      F9MA=ERFCA
      F9MARG=ARG
      F9MME=EXPCM1
      F9MMA=ERFCM1
      RETURN
C
      ENTRY FF9P(I,J,K,L,F9PE,F9PA)
      N=2
      GO TO 200
  170 CONTINUE
      ARG=ETA/2.D0*DSQRT(AC(K)*P/THETA)+F9PMT
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F9PE=ERFCE
      F9PA=ERFCA
      RETURN
  200 CONTINUE
      G=((P/(4.D0*AC(K))+AC(L)*(R(J)-R(I))/(AC(I)-AC(J)))
     **THETA)
      IF(G.LT.0.0D0)G=0.0D0
      F9PMT=DSQRT(G)
      IF(N.EQ.1)GO TO 160
```

```
      IF(N.EQ.2)GO TO 170
C
      ENTRY FF19(I,F19E)
      F19E=-R(I)*THETA
      RETURN
C
      ENTRY FF20P(I,J,F20PE)
      N=1
      GO TO 250
  210 CONTINUE
      F20PE=P*ETA/2.D0*(1.D0+F20PMT)
      RETURN
C
      ENTRY FF20M(I,J,F20ME)
      N=2
      GO TO 250
  220 CONTINUE
      IF(G.LE.3.D-9)GO TO 230
      TERM=1.D0-F20PMT
      GO TO 240
  230 CONTINUE
      G=-G/4.D0
      TERM=2.D0*G*(1.D0+G*(1.D0+G*(2.D0+G*(5.D0+G*
     *(14.D0+G*(42.D0+G*132.D0))))))
  240 CONTINUE
      F20ME=P*ETA/2.*TERM
      RETURN
  250 CONTINUE
      G=4.D0*AC(J)*(R(J)-R(I))/P
      IF(G.LT.-1.D0)G=-1.D0
      SARG=1.D0+G
      F20PMT=DSQRT(DABS(SARG))
      IF(N.EQ.1)GO TO 210
      IF(N.EQ.2)GO TO 220
C
      ENTRY FF21P(I,F21E,F21A)
      ARG=ETA/2.D0*DSQRT(AC(I)*P/THETA)+DSQRT(P*THETA/(4.D0*AC(I)))
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F21E=ERFCE
      F21A=ERFCA
      RETURN
C
      ENTRY FF21(I,F21E,F21A,F21ARG,F21ME,F21MA)
      ARG=ETA/2.D0*DSQRT(AC(I)*P/THETA)-DSQRT(P*THETA/(4.D0*AC(I)))
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F21E=ERFCE
      F21A=ERFCA
      F21ARG=ARG
      F21ME=EXPCM1
      F21MA=ERFCM1
      RETURN
C
      ENTRY FF22(I,F22E,F22A)
      ARG=ETA/2.D0*DSQRT(AC(I)*P/THETA)+DSQRT(P*THETA/(4.D0*AC(I)))
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F22E=ERFCE+P*ETA
```

```
      F22A=ERFCA
      RETURN
C
      ENTRY FF23P(I,J,F23PE,F23PA)
      G=((P/4.D0*AC(J)+R(J)-R(I))*THETA)
      IF(G.LT.0.0D0)G=0.0D0
      PMT=DSQRT(G)
      ARG=ETA/2.D0*DSQRT(AC(J)*P/THETA)+PMT
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F23PE=ERFCE
      F23PA=ERFCA
      RETURN
C
      ENTRY FF23M(I,J,F23ME,F23MA,F23MAR,F23MME,F23MMA)
      G=((P/(4.D0*AC(J))+R(J)-R(I))*THETA)
      IF(G.LT.0.0D0)G=0.0D0
      PMT=DSQRT(G)
      ARG=ETA/2.D0*DSQRT(AC(J)*P/THETA)-PMT
      CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
      F23ME=ERFCE
      F23MA=ERFCA
      F23MAR=ARG
      F23MME=EXPCM1
      F23MMA=ERFCM1
      RETURN
C
      ENTRY FF30(I,J,F30E)
      F30E=-AC(I)*AC(J)*(R(J)-R(I))*ETA/(AC(I)-AC(J))
      RETURN
C
      ENTRY FF31(I,J,K,F31)
      F31=-ETA*(AC(K)*R(K)-AC(K)*(AC(I)*R(I)-AC(J)-R(J))/(AC(I)-AC(J)))
      RETURN
C
      ENTRY FF33(I,J,F33E)
      F33E=-AC(J)*(R(J)-R(I))*ETA
      RETURN
      END
```

```
      SUBROUTINE SUM2(A,B,C,D,E,F)
C
C     THIS ROUTINE COMPUTES THE SUM OF TWO EXPCNETIAL FUNCTIONS :
C      EXP(E)*F=EXP(A)*B+EXP(C)*D
C
      IMPLICIT REAL*16 (G,T-Z)
      REAL*8 A,B,C,D,E,F
      IF(B.EQ.0.D0)GO TO 30
      IF(D.EQ.0.D0)GO TO 50
C     ----B.NE.0..AND.D.NE.0.
      ZB=B
      ZBB=QABS(ZB)
      ZLB=QLOG(ZBB)
      ZILB=QINT(ZLB)
      ZL=ZLB-ZILB
      TL=QSIGN(QEXP(ZL),ZB)
      XL=A+ZILB
      ZD=D
```

```
      ZDD=QABS(ZD)
      ZLD=QLOG(ZDD)
      ZILD=QINT(ZLD)
      ZR=ZLD-ZILD
      TR=QSIGN(QEXP(ZR),ZD)
      XR=C+ZILD
      ZE=XR
      IF(XL.GT.XR)ZE=XL
      G=XR-ZE
      IF(G.GT.-174.D0) GO TO 10
      E=XL
      F=TL
      RETURN
   10 CONTINUE
      IF(G.NE.0.D0)TR=QEXP(G)*TR
      G=XL-ZE
      IF(G.GT.-174.D0)GO TO 20
      E=XR
      F=TR
      RETURN
   20 CONTINUE
      IF(G.NE.0.D0)TL=QEXP(G)*TL
      E=ZE
      F=TL+TR
      RETURN
   30 IF(D.EQ.0.D0)GO TO 40
C     -----B.EQ.0.AND  D.NE.0.
      E=C
      F=D
      RETURN
C     -----B.EQ.0.  AND  D.EQ.0.
   40 E=0.D0
      F=0.D0
      RETURN
C     -----B.NE.0.  AND D.EQ.0.
   50 E=A
      F=B
      RETURN
      END
      SUBROUTINE FERRNT(X,ERF1,EXPC1,ERFC1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NEGXER/ ERFCM1,EXPCM1
      IF(X.EQ.0.0D0)GO TO 10
      IF(X.GT.13.0)GO TO 20
      Y=DABS(X)
      ERFCM1=DERFC(Y)
      EXPCM1=0.0D0
      ERFC1=ERFCM1
      EXPC1=EXPCM1
      IF(X.GT.0.0D0)RETURN
      ERF1=-ERF1
      ERFC1=2.0D0-ERFCM1
      RETURN
   10 CONTINUE
      ERF1=0.0D0
      ERFC1=1.0D0
```

```
      EXPC1=0.0D0
      ERFCM1=ERFC1
      EXPCM1=EXPC1
      RETURN
   20 CONTINUE
      ERF1=0.0D0
      ERFC1=0.0D0
      EXPC1=0.0D0
      ERFCM1=ERFC1
      EXPCM1=EXPC1
      RETURN
      END
```

```
       SUBROUTINE DIFERF(X1,E1,A1,X2,E2,A2,E,A)
C
C     ----- THIS ROUTINE COMPUTES E AND A SUCH THAT
C       EXP(E) * A  = ERFC(X1) - ERFC(X2)
C
C      WHERE: ERFC (ABS(X1)) = EXP(E1) * A1
C             ERFC (ABS(X2)) = EXP(E2) * A2
C
      IMPLICIT REAL*8 (A-H,O-Z)
      J=1
      IF(X1) 3,2,1
 1    J=J+1
 2    J=J+1
 3    IF(X2) 6,5,4
 4    J=J+3
 5    J=J+3
C
C     -------X1    X2    J
C        -     -     1
C        0     -     2
C        +     -     3
C        -     0     4
C        0     0     5
C        +     0     6
C        -     +     7
C        0     +     8
C        +     +     9
C
 6    GO TO (10,20,20,20,30,10,20,10,10),J
   10 A12=A1-A2
 11   GO TO (12,13,13,12,13,13,12,13,13),J
 12   A12=-A12
   13 E=0.D0
      A=A12
      RETURN
C
   20 A12=A1+A2
      A12=A12-2.D0
      GO TO 11
C
 30   E12=0.D0
      A12=A1-A2
      GO TO 13
      END
```

```
      FUNCTION BATE(T,TO,III)
C
C     THIS ROUTINE CALCULATES THE DECAY OF TWO- AND THREE- NUCLIDE
C     CHAINS WITH EQUAL RETENTION COEFFICIENTS
C
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
     *              GAP(6),BREAK,DUR,PE,ETA,TD,WATVEL,PATH
      REAL*8 ENAME
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,BREAK,DUR,PE,ETA,TD,WATVEL,PATH,
     *       ARG1,ARG2,ARG3
      BATE=0.0
      AN1=AQ(1)
      IF(AN1.LE.0.0)RETURN
      AL1=ALAMB(1)
      AL2=ALAMB(2)
      ARG1=-AL1*(T-TO)
      ARG2=-AL2*(T-TO)
      X1=TEXP(ARG1)
      X2=TEXP(ARG2)
      IF(III.EQ.3)GO TO 10
      BATE=AN1*AL2/(AL2-AL1)*(X1-X2)
      RETURN
   10 CONTINUE
      AL3=ALAMB(3)
      ARG3=-AL3*(T-TO)
      X3=TEXP(ARG3)
      BATE=AN1*AL2*AL3*(X1/(AL2-AL1)/(AL3-AL1)+X2/(AL1-AL2)/(AL3-AL2)+
     *X3/(AL1-AL3)/(AL2-AL3))
      RETURN
      END
```

```
      SUBROUTINE TOPDEF
C
C     THIS ROUTINE SORTS OUT ALL ZERO DISCHARGE RATES. IT ALSO
C     IDENTIFYS THE MAXIMUM DISCARGE RATE.
C
      COMMON/C2/    IS,BAND(304),BANDND(304),TIME(304),THETA(304),
     *              TSTART,TFIN,TPEAK,BMAX,DINO
      DATA C /1.E-15/
      J1=0
      J2=0
      TSTART=0.0
      TFIN=0.0
      TPEAK=0.0
      BMAX=0.0
      AMAX=0.0
      IF(BAND(1).LE.1.E-15)GO TO 1120
      J1=1
      TSTART=TIME(1)
 1120 CONTINUE
      DO 1160 J=2,IS
      IF(BAND(J-1).LE.C.AND.BAND(J).GT.C.AND.J1.EQ.0)J1=J
      IF(BAND(J-1).GT.C.AND.BAND(J).LE.C)J2=J-1
      IF(BAND(J).LE.BMAX)GO TO 1140
      TPEAK=TIME(J)
      BMAX=BAND(J)
 1140 CONTINUE
```

```
      IF(BANDND(J).LE.AMAX)GO TO 1160
      AMAX=BANDND(J)
 1160 CONTINUE
      IF(AMAX)1200,1200,1180
 1180 DINO=BMAX/AMAX
 1200 CONTINUE
      IF(J1.EQ.0.AND.J2.EQ.0)GO TO 100
      IF(J1.GT.0.AND.J2.EQ.0)J2=IS
      IF(J1.EQ.0.AND.J2.GT.0)J1=1
      TSTART=TIME(J1)
      TFIN=TIME(J2)
  100 IF(J1.GT.1)GO TO 200
      IS=J2
      GO TO 9999
  200 IS=J2-J1+1
      DO 300 I=1,IS
      II=J1+I-1
      TIME(I)=TIME(II)
      THETA(I)=THETA(II)
      BAND(I)=BAND(II)
      BANDND(I)=BANDND(II)
  300 CONTINUE
 9999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINTX(NCHAIN)
C
C     THIS ROUTINE WRITES THE LINEPRINTER OUTPUT FOR THE CHAIN FRAGMENTS
C
      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DEC(3),
     *              GAP(6),T0,DUR,PE,ETA,TD,V0,PATH
      COMMON/C2/    IS,BAND(304),BANDND(304),TIME(304),THETA(304),
     *              TSTART,TFIN,TPEAK,BMAX,DINO
      REAL*8 ENAME,TEXT1(3)
      REAL*8 THALF,AQ,ALAMB,EQ,DEC,GAP,T0,DUR,PE,ETA,TD,V0,PATH
      DATA TEXT1 /'SINGLE','DOUBLE','TRIPLE'/
      WATTIM=PATH/V0
    1 FORMAT('1',A6,'-CHAIN MIGRATION CALCULATION'/' ',34('*')/'0NUCLIDE
     *: ',A6)
    2 FORMAT('+',T17,'FROM ',A6)
    3 FORMAT('+',T28,'DECAYING VIA ',A6)
      WRITE(9,1) TEXT1(NCHAIN),ENAME(NCHAIN)
      IF(NCHAIN.GT.1)WRITE(9,2) ENAME(1)
      IF(NCHAIN.EQ.3)WRITE(9,3)ENAME(2)
    4 FORMAT('0DESCRIPTION OF LEACH SCENARIO:'/'+',29('_')/'0TIME OF LEA
     *CH INCIDENT:',1PG16.6 ,'  YEARS AFTER REACTOR DISCHARGE'/' LEACH D
     *URATION:',10X,G14.6,'  YEARS'/' MIGRATION PATH LENGTH:',3X,G14.6,
     *'  METERS'/' GROUNDWATER VELOCITY:',4X,G14.6,'  METERS/YEAR'/' GRO
     *UNDWATER TRAVEL TIME:',1X,G14.6,'  YEARS'/' PECLET NUMBER:',11X,
     *G14.6/'0NUCLIDE DATA'/'+',12('_'),40X,A6,6X,A6,6X,A6)
      WRITE(9,4) T0,DUR,PATH,V0,WATTIM,PE,(ENAME(I),I=1,NCHAIN)
    5 FORMAT('0ACTIVITY AT THE TIME OF LEACH INCIDENT (CURIES)',1X,1PE12
     *.2,2E12.2)
      WRITE(9,5)(AQ(I),I=1,NCHAIN)
      DO 15 I=1,NCHAIN
      GAP(I+3)=GAP(I+3)*1.93729E-6
```

```
   15 GAP(I)=GAP(I)*1.93729E-6
    6 FORMAT(' MASS AT THE TIME OF LEACH INCIDENT',5X,'(MOLES)',2X,1PE12
     *.2,2E12.2)
      WRITE(9,6)(GAP(I),I=1,NCHAIN)
    7 FORMAT(' MASS AT THE END OF LEACH INCIDENT',6X,'(MOLES)',2X,1PE12.
     *2,2E12.2)
      NN=NCHAIN+3
      WRITE(9,7)(GAP(I),I=4,NN)
    9 FORMAT(' DECAY NUMBER',36X,1PE12.2,2E12.2)
      WRITE(9,9)(DEC(I),I=1,NCHAIN)
    8 FORMAT(' RETENTION COEFFICIENT',27X,3F12.0)
      WRITE(9,8)(EQ(I),I=1,NCHAIN)
   10 FORMAT('0BRIEF PEAK INFORMATION'/'+',22('_')/'0TIME OF INITIAL DIS
     *CHARGE',1PG19.6 ,'  YEARS AFTER REACTOR DISCHARGE'/' TIME OF PEAK
     *TAIL',13X,G14.6,'  YEARS AFTER REACTOR DISCHARGE'/' TIME OF PEAK M
     *AXIMUM',10X,G14.6,'  YEARS AFTER REACTOR DISCHARGE'/' MAXIMUM DISC
     *HARGE RATE',8X,1PE14.2,'  CURIES/YEAR'/' NUMBER OF TIMESTEPS USED'
     *,15X,I5/' DISP TO NO-DISP PEAKS',9X,0PF14.8)
      WRITE(9,10) TSTART,TFIN,TPEAK,BMAX,IS,DINO
   11 FORMAT(////6X,'TIME',11X,'DIM.LESS TIME',2X,'BAND RELEASE',3X,'D:O
     * NO-DISP'/)
      IF(IS.EQ.0)GO TO 100
      WRITE(9,11)
   12 FORMAT(1X,1PE15.4,3E15.4)
      DO 20 I=1,IS
   20 WRITE(9,12) TIME(I),THETA(I),BAND(I),BANDND(I)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE SPEC(GNAME,ENAME,IFILE,IAPPR,FACT,*)
C
C     THIS ROUTINE CALCULATES SPECIAL NUCLIDE CONTRIBUTIONS BASED ON
C     EITHER OF TWO DECAY APPROXIMATIONS:
C        IAPPR=1: SHORTLIVED DAUGHTER NUCLIDE IN RADIOACTIVE EQUILLIB-
C                 RIUM WHITH ITS PREDECESSOR.
C        IAPPR=2: SHORTLIVED PARENT NUCLIDE WHICH IS ASSUMED TO DECAY
C                 COMPLETLY BEFORE LEACHING.
C     GNAME IS THE CONTRIBUTION TO BE CALCULATED. ENAME IS THE ALREADY
C     CALCULATED CHAIN UPON WICH THE APPROXIMATIVE CALCULATION
C     OF GNAME IS BASED. ENAME IS TO BE FOUND IN DATASET NR 12+IFILE
C     IF IAPPR=2 THE DISCHARGE RATE FOR GNAME IS CALCULATED BY MULTI-
C     PLYING THE DISCHARGE RATE FOR ENAME BY FACT. IF NO DISCHARGE
C     RATE CAN BE CALCULATED THE CONTROL IS TRANSFERRED TO THE STATE-
C     MENT LABELED * IN THE CALLING ROUTINE.
C
      COMMON /PRINT/ MIGOUT
      COMMON /C2/ NSTEP,B(304),DUM1(304),T(304),DUM3(304),
     *   TD1,TD2,TD3,BD,DD
      COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
     *   CMAX(75,10),DOSE2(75)
      REAL*8 GNAME(3),ENAME(3),ANAME(3),TEXT(2),BLANK
      LOGICAL MIGOUT
      DATA BLANK/'        '/,TEXT/'DAUGHTER','PARENT  '/
C
      IF=IFILE
      ID=12+IF
```

```
 10    CONTINUE
       READ(ID,END=900)ANAME,IS
       DO 20 I=1,IF
       IF(ANAME(I).NE.ENAME(I))GO TO 30
 20    CONTINUE
       GO TO 50
 30    IF(IS.EQ.0)GO TO 10
       READ(ID)(T(I),B(I),I=1,IS)
       GO TO 10
 50    IF(IS.EQ.0)GO TO 800
       READ(ID)(T(I),B(I),I=1,IS)
       WRITE(16)GNAME,IS
       IF(IAPPR.EQ.1)GO TO 70
       DO 60 I=1,IS
       B(I)=B(I)*FACT
 60    CONTINUE
 70    CONTINUE
       WRITE(16)(T(I),B(I),I=1,IS)
       BMAX=B(1)
       IMAX=1
       DO 80 I=2,IS
       IF(B(I).LE.BMAX)GO TO 80
       BMAX=B(I)
       IMAX=I
 80    CONTINUE
       IJ=INTNUK(GNAME(1))
       ICONTR(IJ)=ICONTR(IJ)+1
       NN=ICONTR(IJ)
       TSTART(IJ,NN)=T(1)
       TPEAK(IJ,NN)=T(IMAX)
       CMAX(IJ,NN)=BMAX
       TEND(IJ,NN)=T(IS)
       REWIND ID
       IF(.NOT.MIGOUT)RETURN
       WRITE(9,91)GNAME(1),GNAME(2)
       IF(GNAME(3).NE.BLANK)WRITE(9,92)GNAME(3)
 91    FORMAT('1APPROXIMATIVE DISCHARGE RATE FOR ',A6,' FROM ',A6)
 92    FORMAT('+',51X,' AND ',A6)
       WRITE(9,93)TEXT(IAPPR),ENAME
 93    FORMAT(' A SHORTLIVED ',A8,'-NUCLIDE APPROXIMATION WAS MADE ON THE
      * BASIS OF THE CHAIN ',3A8/)
       WRITE(9,94)
 94    FORMAT('0',5X,'TIME',11X,'BAND-RELEASE'/)
       DO 95 I=1,IS
       WRITE(9,96)T(I),B(I)
 95    CONTINUE
 96    FORMAT(1PE16.4,E15.4)
       RETURN
 800   CONTINUE
       WRITE(6,801)GNAME(1),GNAME(2)
       IF(GNAME(3).NE.BLANK)WRITE(6,802)GNAME(3)
 801   FORMAT('0THE CHAIN ',A6,' FROM ',A6)
 802   FORMAT('+',29X,' AND ',A6)
       WRITE(6,803)ID,(ENAME(I),I=1,IF)
 803   FORMAT(' WAS NOT CALCULATED BECAUSE OF ZERO RESULT IN DATASET NR '
      *,I3,' FOR ',3A8/)
```

```
      REWIND ID
      RETURN 1
  900 CONTINUE
      WRITE(6,901)GNAME(1),GNAME(2)
      IF(GNAME(3).NE.BLANK)WRITE(6,902)GNAME(3)
  901 FORMAT('0WHILE EXECUTING SPEC FOR ',A6,' FROM ',A6)
  902 FORMAT('+',44X,' AND ',A6)
      WRITE(6,903)ID,(ENAME(I),I=1,IF)
  903 FORMAT(' DATASET NR ',I3,' DID NOT CONTAIN THE CHAIN ',3A8/)
      REWIND ID
      RETURN 1
      END
```

```
      SUBROUTINE CHAIN(NAME,N,IC1,I2)
C
C     THIS  ROUTINE RULES THE ADDITION OF DIFFERENT DECAY CHAIN
C     FRAGMENTS TO A NUCLIDES TOTAL DISCHARGE RATE
C
      COMMON/C2/    NSTEP,T(304),Y(304),BAND(304),TIME(304),
     *              TBEGIN,TFIN,TTOPP,BRPEAK,DINO
      COMMON/PEAKS/ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
     *             CMAX(75,10),DOSE2(75)
      REAL*8 NAME,IC1(20),IBL,NN(3)
      DATA IBL/'        '/
      DO 15 I=1,N
   15 Y(I)=0.0
      IJ=INTNUK(NAME)
      NJ=ICONTR(IJ)
      DOSE=DOSE2(IJ)
      I2=0
      J2=0
      DO 150 JJJ=13,16
      JJ=JJJ
      II=JJ-12
      IF(JJ.GT.15)II=1
  110 CONTINUE
      IF(J2.GE.NJ)GO TO 155
      READ(JJ,END=150)(NN(I),I=1,3),ISTEP
      IF(ISTEP.EQ.0)GO TO 110
      READ(JJ)(TIME(I),BAND(I),I=1,ISTEP)
      IF(NN(II).NE.NAME)GO TO 110
      CALL INTPOL(N,ISTEP)
      IF(JJ.GT.15)GO TO 120
      J2=J2+1
      I2=I2+1
      IC1(I2)=NN(1)
      GO TO 110
  120 CONTINUE
      J2=J2+1
      I2=I2+1
      IC1(I2)=NN(2)
      IF(NN(3).EQ.IBL)GO TO 110
      I2=I2+1
      IC1(I2)=NN(3)
      GO TO 110
  150 CONTINUE
  155 CONTINUE
```

```
      DO 160 JJ=13,16
      REWIND JJ
  160 CONTINUE
      RETURN
      END
      SUBROUTINE INTPOL(N,ISTEP)
C
C     THIS ROUTINE IS USED BY CHAIN FOR THE ADDITION OF
C     DIFFERENT CHAIN FRAGMENTS
C
      COMMON/C2/    NSTEP,T(304),Y(304),BAND(304),TIME(304),
     *              TBEGIN,TFIN,TTOPP,ERPEAK,DINO
      J1=1
      DO 120 I=1,N
  116 IF(J1.GT.ISTEP-1)GO TO 125
      IF(T(I).GE.TIME(J1).AND.T(I).LT.TIME(J1+1).AND.BAND(J1).GT.0.0
     *.AND.BAND(J1+1).GT.0.0) GO TO 117
      IF(T(I).LT.TIME(J1))GO TO 120
      J1=J1+1
      GO TO 116
  117 Y(I)=Y(I)+BAND(J1)+(T(I)-TIME(J1))/(TIME(J1+1)-TIME(J1))*
     *(BAND(J1+1)-BAND(J1))
  120 CONTINUE
  125 CONTINUE
      RETURN
      END
      SUBROUTINE TSCALE(YMAX,TIMY,NN,TIMAX,TIMIN,U,NT,*)
C
C     THIS SUBROUTINE CALCULATES A LOGARITHMIC TIMESCALE, U, BETWEEN
C     THE POINTS T1 AND T2. T1 AND T2 ARE CHOSEN TO BE SO CLOSE TO
C     TMIN AND TMAX RESPECTIVILY AS POSSIBLE. ONE OR TWO POINTS IN U ARE
C     FORCED TO BE PLACED AT TP1 AND TP2.
      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,T0,T1,PE,ETA,TD,V,AL
C
      DIMENSION U(304)
      DIMENSION  YMAX(1),TIMY(1)
      DATA IS/100/
      REAL*8 BASE,T,TP1,TP2,TMAX,TMIN,T1
      AMAX=0.0
      IMAX=0
      TMAX=TIMAX
      TMIN=TIMIN
      DO 10 I=1,NN
      IF(YMAX(I).LE.AMAX)GO TO 10
      AMAX=YMAX(I)
      IMAX=I
   10 CONTINUE
      IF(IMAX.EQ.0)RETURN 1
      JMAX=0.0
      BMAX=0.0
      DO 20 I=1,NN
      IF(YMAX(I).LE.BMAX)GO TO 20
      IF(I.EQ.IMAX)GO TO 20
      JMAX=I
      BMAX=YMAX(I)
   20 CONTINUE
```

```
      IF(JMAX.EQ.0.OR.BMAX.LT.1.E-20)GO TO 30
      TP2=AMAX1(TIMY(IMAX),TIMY(JMAX))
      TP1=AMIN1(TIMY(IMAX),TIMY(JMAX))
C
C     CALCULATE TIMESTEPS FOR TWO NON-SIMULTANEOUS PEAKS
C
      A1=DLOG(TP2/TP1)/DLOG(TMAX/TMIN)*IS
      IF(A1.LT.0.5)GO TO 30
      N1=IFIX(A1)+1
      BASE=(TP2/TP1)**(1/FLOAT(N1))
      A2=DLOG(TP1/TMIN)/DLOG(BASE)+1
      N2=IFIX(A2)
      T1=TP1/BASE**(N2)
      A3=DLOG(TMAX/TP2)/DLOG(BASE)+1
      N3=IFIX(A3)
      T2=TP2*BASE**N3
      NT=N1+N2+N3
      GO TO 70
C
C     CALCULATE TIMESTEPS FOR SINGLE OR SIMULTANEOUS PEAKS
C
   30 CONTINUE
      TP2=TIMY(IMAX)
      IF(TP2.EQ.TMIN.OR.TP2.EQ.TMAX)GO TO 60
      A1=DLOG(TP2/TMIN)/DLOG(TMAX/TMIN)*IS
      IF(A1.LT.0.5)GO TO 50
      N1=IFIX(A1)+1
      BASE=(TP2/TMIN)**(1/FLOAT(N1))
      A2=DLOG(TMAX/TP2)/DLOG(BASE)+1
      N2=IFIX(A2)
      T2=TP2*BASE**N2
      T1=TMIN
      NT=N1+N2
      GO TO 70
   50 CONTINUE
      A2=DLOG(TMAX/TP2)/DLOG(TMAX/TMIN)*IS
      N2=IFIX(A2)+1
      T2=TMAX
      BASE=(TMAX/TP2)**(1/FLOAT(N2))
      T1=TP2/BASE
      NT=N2+1
      GO TO 70
C
C     CALCULATE TIMESTEPS WHEN PEAKS COINCIDE WITH MAX OR MIN
C
   60 CONTINUE
      BASE=(TMAX/TMIN)**(1/FLOAT(IS)-1)
      T1=TMIN
      T2=TMAX
      NT=IS
C
C     CALCULATE TIMESCALE
C
   70 CONTINUE
      U(1)=T1
      T=T1
```

```
      DO 100 J=2,NT
      T=T*BASE
  100 U(J)=T
      RETURN
      END
      SUBROUTINE PLOT1(NAME,NN,NR,NS,YMAX,WRITEX)
C
C     THIS ROUTINE ARRANGES THE DIFFERENT VARIABLES BEFORE PLOTTING
C
      COMMON/C2/ NSTEP,X(304),Y(304),DOSE1(304),Y1(304),
     *              TBEGIN,TFIN,TTOPP,BMAX,DINO
      DIMENSION YREAL(304)
      REAL*8 YNAME(21),NAME,NN(20)
      LOGICAL WRITEX,AREADY
      DATA NCOL/110/,IU/20/,NPC/1/,IDEC/2/
      YNAME(1)=NAME
      DO 1 I=1,NR
    1 YNAME(I+1)=NN(I)
      NR=NR+1
      AREADY=.FALSE.
      IS=-1
      Y1MIN=-12
      YL=3.
      IF(YMAX.LE.0.0)WRITE(6,3)YNAME(1)
    3 FORMAT('0YMAX.LE.0 IN PLOT1 FOR ',A6)
      IF(YMAX.LE.0.0)RETURN
      Y1MAX=ALOG10(YMAX)
      IND2=NS
      I1=1
      I2=NS
   50 CONTINUE
      IND1=I1
      IF(I1.GE.IND2)RETURN
      DO 300 I=IND1,IND2
      IF(Y(I).LE.0.0.AND.I.EQ.I1)GO TO 200
      IF(Y(I).LE.0.0)GO TO 100
      Y1(I)=ALOG10(Y(I))
      YREAL(I)=Y(I)
      GO TO 300
  100 CONTINUE
      I2=I-1
      CALL PLOTY(I1,I2,NPC,YNAME,Y1MIN,Y1MAX,YL,0.,NCCL,IU,IS,
     *    WRITEX,IDEC,NR,YREAL)
      AREADY=.TRUE.
      I1=I+1
      GO TO 50
  200 CONTINUE
      I1=I1+1
  300 CONTINUE
      IF(AREADY)RETURN
      CALL PLOTY(I1,I2,NPC,YNAME,Y1MIN,Y1MAX,YL,0.,NCCL,IU,IS,
     *    WRITEX,IDEC,NR,YREAL)
      RETURN
      END
      FUNCTION TERFC(X)
      IF(X.GT.12.0)GO TO 1
```

```
      TERFC=ERFC(X)
      GO TO 2
    1 TERFC=0.0
    2 CONTINUE
      RETURN
      END
      SUBROUTINE PLOTY(I1,I2,NPC,NAME,YMI,YMA,YL,FI,NCOL,IU,IS,
     *   WRITEX,IDEC,NR,YREAL)
      COMMON/C2/    NSTEP,X(304),Z(304),TIME(304),Y(304),
     *              TBEGIN,TFIN,TTOPP,BRPEAK,DINO
C
C           LINE PRINTER PLOTTING ROUTINE WITH VERTICAL X-AXIS.
C           THE STEP IN X-DIRECTION IS CONSTANT = ONE LINE.
C           Y      = THE ARRAY CONTAINING THE VALUES TO BE PLOTTED
C           I1     = THE FIRST Y-ELEMENT TO BE PLOTTED AND THE FIRST LINE
C                    NUMBER AS WELL.
C           I2     = THE LAST LINE NUMBER TO BE PLOTTED AND THE LAST
C                    Y-ELEMENT AS WELL, IF NPC=1.
C           NPC    = THE NUMBER OF CURVES TO BE PLOTTED SIMULTANEOUSLY.
C                    NPC MAY NOT EXCEED NPMAX..
C                    IF FOR INSTANCE NPC=3 AND THE VARIABLES TO BE PLOTTED
C                    ARE A, B AND C, THE VALUES SHALL BE STORED IN THE
C                    Y-ARRAY AS FOLLOWS: A1,B1,C1, A2,B2,C2, A3,B3,C3, ...
C           NAME   = THE NAME(S) OF THE PLOTTED VARIABLE(S) (6H-HOLLERITH)
C                    IF NPC>1, THIS PARAMETER IN THE CALLING PROGRAM MUST
C                    BE AN ARRAY, WHERE THE NAMES ARE STORED IN THE SAME
C                    ORDER AS ARE THE VALUES IN THE Y-ARRAY.
C           YMI    = EXTERNALLY SET LOWER Y-LIMIT (FOR THE SCALE-CALC.)
C           YMA    = EXTERNALLY SET UPPER Y-LIMIT (FOR THE SCALE-CALC.)
C           YL     = 0: IF YMI<YMA, THE LIMITS ARE TAKEN FROM YMI AND YMA,
C                       ELSE THE LIMITS ARE COMPUTED FROM Y, INDEPENDENTLY
C                       OF YMI AND YMA.
C                  = 1: TAKING THE LIMITS FROM YMI AND YMA, BUT ADJUSTING
C                       YMIN, IF ANY Y<YMI.
C                  = 2: TAKING THE LIMITS FROM YMI AND YMA, BUT ADJUSTING
C                       YMAX, IF ANY Y>YMA.
C                  = 3: 1. AND 2.
C           FI     > 0: THE SPACE BETWEEN X-AXIS AND THE Y-POSITION IS
C                       FILLED WITH XMARKS (STAPLE DIAGRAM).
C                       ORIGO IS POSITIONED AT Y=0. NO XMARK FOR Y=0.
C                       THE EVERY 100 LINES NUMBER MARKING IS SUPPRESSED.
C                       THE STAPLE DIAGRAM IS POSSIBLE ONLY IF NPC=1.
C           NCOL   = MAXIMAL NUMBER OF COLUMNS AVAILABLE DURING PRINTING
C           IU     = OUTPUT UNIT NUMBER
C           IS     = A SWITCH, THAT SHALL BE SET EITHER <0 (XPLOT THEN
C                    STARTS ITS PRINTING ON NEW PAGE) OR =0 (THE PRINTING
C                    STARTS AFTER A SINGLE LINE-SKIP) BEFORE THE FIRST
C                    CALL. PLOTY RESETS IS ITSELF. AT A REPEATED CALL THE
C                    PLOTTING CONTINUES ON THE ORIGINAL X-AXIS.
C           THE SUBROUTINE SCALE(A,B,C,I) COMPUTES THE SCALE OUT OF THE
C           PREL. VALUE A, AND PUTS THE RESULT IN B. C IS THE PRINTED
C           SCALE AND I IS THE POWER OF 10.
C           THE SUBROUTINE DECODH(N,A,I,J) DECODES THE INTEGER NUMBER N
C           AND PUTS THE J RIGHTMOST FIGURES CONVERTED INTO 1H-HOLLERITH
C           INTO THE ARRAY A FROM A(I) TO A(I+J-1). LEFT ZEROS BLANKED.
C
```

```
      LOGICAL WRITEY,WRITEX
      LOGICAL JUMP,OUT,FILL,SINGLE
C
      DIMENSION IX(16),IY(15),XD(2),YAXIS(2),HNXO(3),FMX(9),FM1(9),
     1          FM2(8),FM3(14),FM4(14),FM5(7),FM6(11),FM7(5),FM8(9),
     2          NAME(1)
      DIMENSION YREAL(1)
      REAL*8 NAME
      REAL*8 FM9(11),FM10(11),RUBR(3),FME(2),FMT(2),FMS(3)
      REAL*8 DUM1(3),DUM2(2),DUM3(2),FM1,FM2,FM3,FM4,FM5,
     1   FM6,FM7,FM8,FMX,YAXIS,BLANK,S,PLURAL
     2   ,YMARK,XMARK,DOT
     3   ,HNXO,XD
      EQUIVALENCE (FM1(7),DUM1(1)),(FM2(7),DUM2(1)),
     1      (FM5(6),DUM3(1))
C
      EQUIVALENCE (YMARK(1),XMARK)
C
C         TO INCREASE THE PERMITTED NUMBER OF SIMULTANEOUSLY PLOTTED
C         CURVES, JUST INCREASE NPMAX AND EXTEND THE YMARK DIMENSION
C         AND DATA ITEM LIST..
      DIMENSION YMARK(4)
      DATA NPMAX/4/, YMARK/1HX,1HO,1HH,1HS/
C
      DATA FM9(1)/5H(1H+,/,FM9(5)/2HX,/,FM9(6)/8H7HX-VALU/,
     1      FM9(7)/2HE,/,FM9(9)/8HX,7HY-VA/,FM9(10)/4HLUE)/
     2      ,FM9(11)/1H /,FME/5H2(1PE,2H))/
     3      ,FMS/3H1PE,1H),2HE)/
      DATA MAXCOL/132/
C
      DATA DOT/1H./, IAST/1H*/, IPLUS/1H+/, BLANK/1H /, PLURAL/1HS/,
     1      YAXIS/5H*....,5H...../, FMX/2HX,,6H(5X,A1,5H,4X),,3H(I6,
     2      4H(A1,,4H9X),,2HA1,4H(4X,,3HI6)/
C
      DATA FM1(1)/5H(1H ,/, DUM1/6H(I6,4X,5H),5X,,4H1HY)/,
     1      FM2(1)/5H(1H ,/,FM2(4)/2HX,/,FM3(1)/5H(1H ,/,DUM2
     2       /6H(2A5),,4H1H>)/, FM3(1)/5H(1H ,/, FM3(9)/3HI5,/, FM3(14)/
     3      3HA1)/, FM4(1)/5H(1H+,/, FM4(9)/3H6X,/, FM4(14)/1H)/, FM5(1)/
     4      5H(1H ,/, DUM3/3HI5,,4H1H.)/, FM6(1)/5H(1H ,/,
     5      FM6(5)/5HX,A1,/, FM6(11)/1H)/,FM7(1)/5H(1H+,/,FM7(5)/5HX,A1)/
     6      ,FM8(1)/5H(1H+,/, FM8(5)/2HX,/, FM8(9)/3HA1)/
C
      INDEX1 = I1
      INDEX2 = I2
      IF(IS.GT.0) GOTO 300
C
      IF(NCOL.GE.19.AND.IU.GT.0.AND.NPC.LE.NPMAX) GOTO 10
    5 PRINT 801, NAME
      RETURN
   10 NP = MAX0(1,NPC)
      IO = IU
      SINGLE = NP.EQ.1
      WRITEX=WRITEX.AND.MAXCOL-NCOL.GT.IDEC+7
      WRITEY=WRITEX.AND.SINGLE.AND.MAXCOL-NCOL.GT.2*IDEC+16
      FMT(1)=FME(1)
      FMT(2)=FME(2)
```