

SKBF
KBS

TEKNISK
RAPPORT

80-03

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

Bertil Grindfelt
Mark Elert

Kemakta konsult AB, January 1980

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

Bertil Grundfelt
Mark Elert

Kemakta konsult AB, January 1980

This report concerns a study which was conducted for the KBS project. The conclusions and viewpoints presented in the report are those of the author(s) and do not necessarily coincide with those of the client.

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

	page
1. History and project description	1
2. Updating of GETOUT with reference to PNL's new version	2
3. Calculation and storage of source strengths at time of canister breakthrough	4
4. Calculation of time scales	5
5. Printouts for evaluation	7
6. Plot program	7
References	8

Appendices

1. Description of GETOUT and its input data requirements
2. Program listing
3. Description of input data for a GETOUT run
4. Examples of output from a GETOUT run

1. History and product description

In the safety analysis in the KBS project /1, 2/, a one-dimensional dispersion model with chain decay and geochemical 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:

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

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 (ground-water 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 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 sub-routines 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.

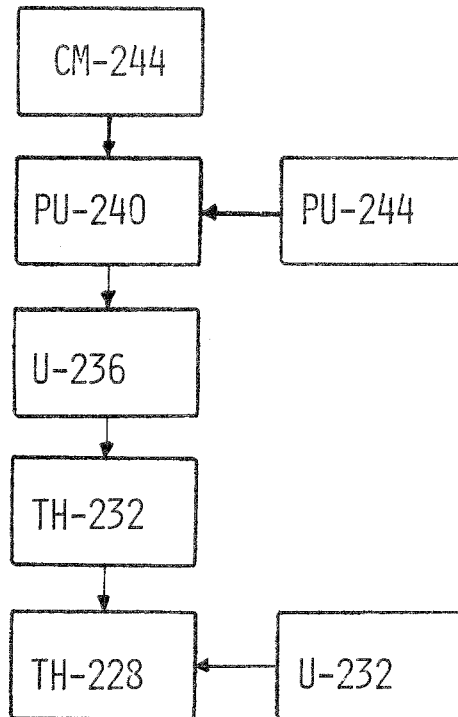
The single-, double- and triple-nuclide chains are calculated by the subroutines SINGLE, DOUBLE and TRIPLE with associated auxiliary routines. The above-mentioned approximations APPROX 1 and APPROX 2 are calculated in the routine SPEC. The individual contributions to a nuclide's total inflow are stored 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:

Appendix 1 (3)

- 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.

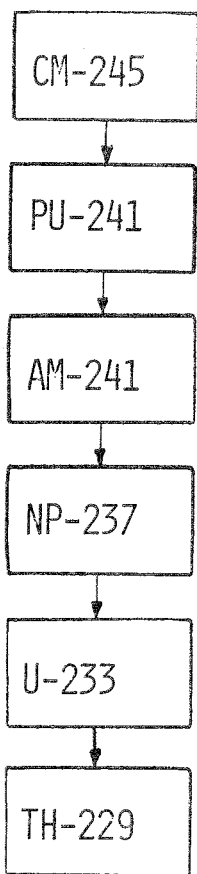
- File 5 with information on the time of canister breakthrough, leach duration, groundwater travel time, nuclides to be calculated, retention factors (coefficients) for these nuclides etc. The records in file 5 are described in Appendix 3.



Calculation logic for the 4N chain

(chain)

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



Calculation logic for the 4N + 1 chain

(chain)

Cm-245 from Cm-245: SINGLE

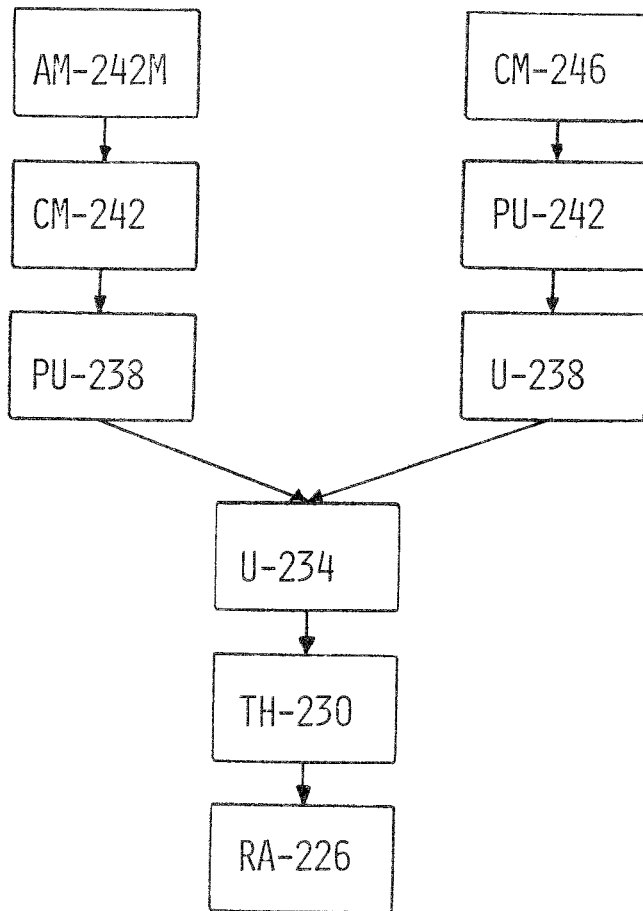
Pu-241 from Cm-245: DOUBLE
 from Pu-241: SINGLE

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



Calculation logic for the 4N + 2 chain

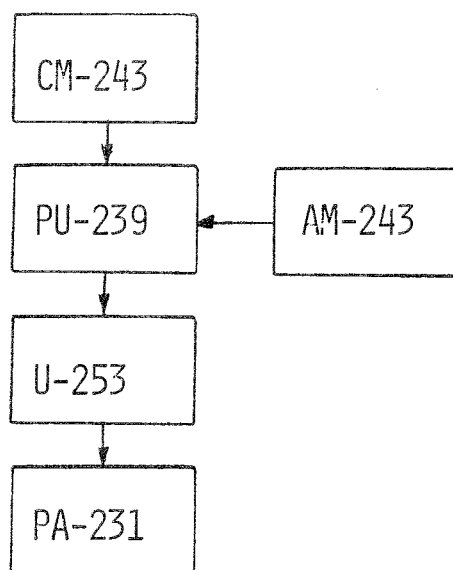
(chain)

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

Figure 3 (continued)

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)



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: TRIPLE (Pa-231 from Pu-239 and Cm-243: with APPROX 2)

from Pu-239: TRIPLE

from U-235: DOUBLE

from Pa-231: SINGLE

Figure 5

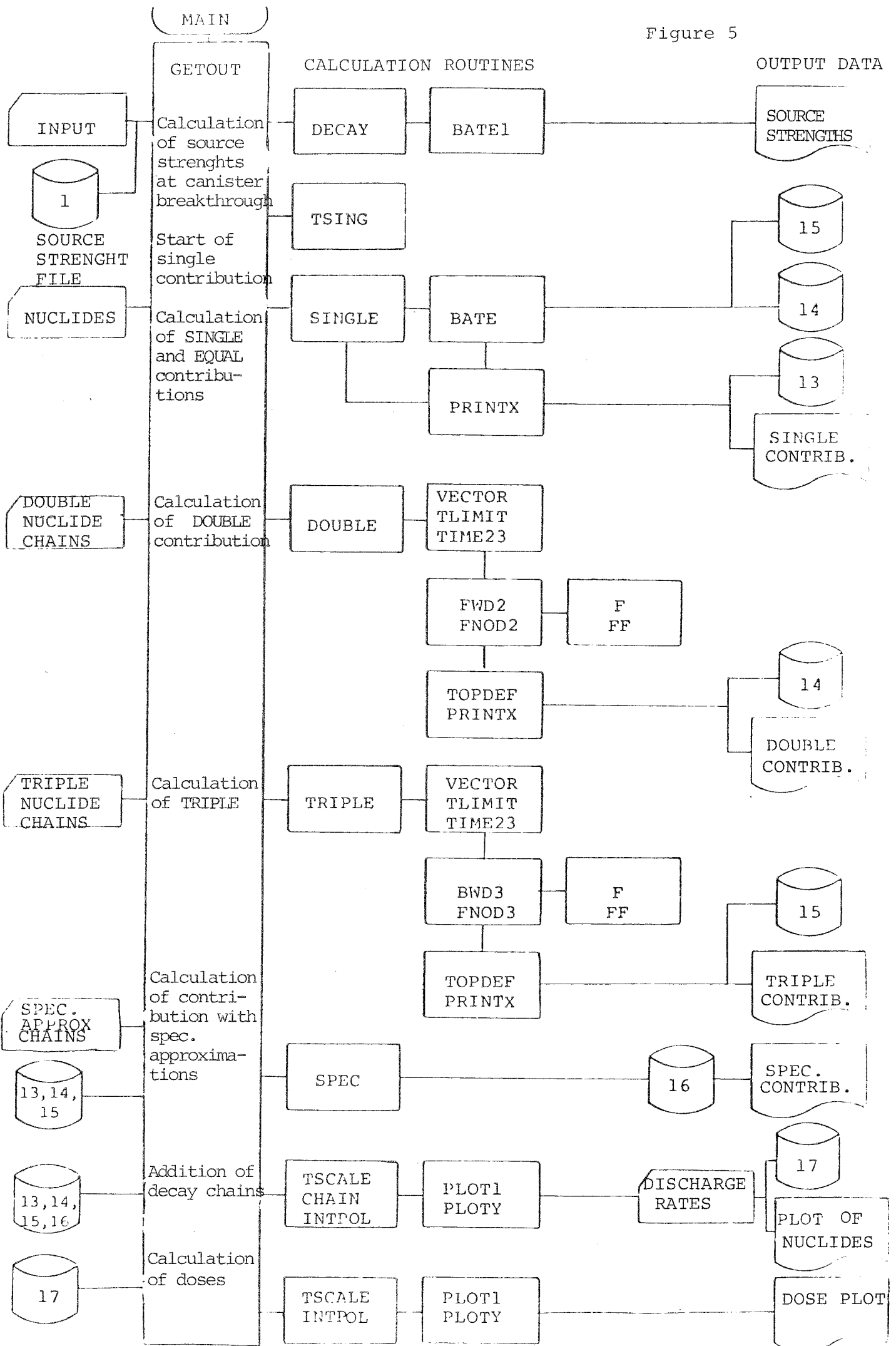


Figure 6

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
SE70	6.50+04	3.95-01
KR35	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 HALFLIVES FOR SPENT PWR-FUELS, HALFLIVES ARE ACCORDING TO ORIGEN'S DATA LIBRARY. SOURCE STRENGTHS ARE TAKEN FROM KRS TR 01 (1977-04-05)

Program listing

<u>Routine</u>	<u>Card No</u>	<u>Routine</u>	<u>Card No</u>
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		

```

1 LOGICAL MIGOUT,NOPUN,NOLINK,LINK,PUNCH,WRITEX
2 COMMON/PRINT/ MIGOUT
3 CALL ERRSET(208,0,-1,1,1)
4 CALL ERRSET(207,50,-1,1,1)
5 READ(5,1)LINK,PUNCH,WRITEX,MIGOUT
6 1 FORMAT(5L5)
7 IF(.NOT.(LINK.OR.MIGOUT))GO TO 100
8 NOLINK=.NOT.LINK
9 NOPUN=.NOT.PUNCH
10 CALL GETOUT(NOLINK,NOPUN,WRITEX)
11 STOP
12 100 WRITE(6,2)
13 2 FORMAT('1*** NO OUTPUT WAS SPECIFIED AND AS IT IS LATE THE COMPUT
14 *ER CHOSED TO TERMINATE EXECUTION ***')
15 STOP
16 END
17 BLOCK DATA
18 COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
19 C
20 C BLOCKDATA INITIATES NUCLIDE NAMES AND RETENTION COEFFICIENTS TO
21 C DUMMY VALUES
22 C
23 * COMMON/DEY1/ GAP(6),BREAK,DUR,PE,ETA,TD,V,PATH
24 COMMON/DEY2/ FISS(40),CHAIN(6,3,4),IFI,ICH
25 COMMON/DEY2/ ETEXP(75),AN(75),C(75),ANAME(76),SF(75),NUK(75),
26 * IN,ISL
27 COMMON/PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
28 * CMAX(75,10),DOSE2(75)
29 COMMON/HALVA/ THALV(75),RET(75)
30 REAL*16 THALV
31 REAL*16 ETEXP,AN,C
32 REAL*8 ENAME,ANAME,FISS,CHAIN
33 REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,BREAK,DUR,PE,ETA,TD,V,PATH
34 DATA EQ/3*0.0/
35 DATA ETA /1.D0/
36 DATA ICONTR /75*0/
37 DATA RET/75*0.0/
38 DATA SF/75*1./
39 DATA IFI/24/,
40 1 FISS/'H 3 ', 'C 14 ', 'KR85 ', 'SR89 ', 'SR90 ', 'Y 91 ',
41 *'ZR93 ',
42 2 'ZR95 ', 'NB95 ', 'TC99 ', 'RU103 ', 'RU106 ', 'SB125 ', 'TE125M',
43 3 'TE127M', 'I 129 ', 'CS134 ', 'CS135 ', 'CS137 ', 'CE144 ',
44 4 'PM147 ', 'SM151 ', 'EU154 ', 'EU155 ', '16*'XX000 '/
45 DATA ICH/4/,
46 1 CHAIN/'CM246 ', 'PU242 ', 'U 238 ', 'U 234 ', 'TH230 ', 'RA226 ',
47 2 'AM242M', 'CM242 ', 'PU238 ', 'U 234 ', 'TH230 ', 'RA226 ',
48 3 6*
49 4 'CM244 ', 'PU240 ', 'U 236 ', 'TH232 ', 'TH228 ',
50 5 'CM248 ', 'PU244 ', 'PU240 ', 'U 236 ', 'TH232 ', 'TH228 ',
51 6 3*
52 7 'PU236 ', 'U 232 ', 'TH228 ',
53 8 2* 'CM243 ', 'PU239 ', 'U 235 ', 'PA231 ',
54 9 2* 'AM243 ', 'PU239 ', 'U 235 ', 'PA231 ',
55 6*

```

```

55      *      'CM245 ', 'PU241 ', 'AM241 ', 'NP237 ', 'U 233 ', 'TH229 ' ,
56      1      12* '  ' /
57      DATA ENAME /3*'XX000'/
58      END
59      SUBROUTINE GETOUT(NLINK, NPUNCH, WRITEX)
60
61      C      THIS ROUTINE ADMINISTRATES THE CALCULATIONS OF THE NUCLIDE
62      C      RELEASE RATES
63      C
64      COMMON/FFFFF/ ENAME(3), THALF(3), AQ(3), ALAMB(3), EQ(3), DNR(3),
65      *      GAP(6), BREAK, DUR, PE, ETA, TD, WATVEL, PATH
66      COMMON/C2/   NSTEP, U(304), Y(304), DOSE1(304), TIME(304),
67      *      TBEGIN, TFIN, TTOPP, BRPEAK, DINO
68      COMMON/HALVA/ THALV(75), RET(75)
69      COMMON/DEY2/  ETEXP(75), AN(75), C(75), ANAME(76), SF(75), NUK(75),
70      *      IN, ISL
71      COMMON /PEAKS/ ICONTR(75), TSTART(75,10), TPEAK (75,10), TEND(75,10),
72      *      CMAX(75,10), DOSE2(75)
73      COMMON /LGTEST/ LG(6)
74      REAL*16 T, TUPP, THALV, ETEXP, AN, C
75      REAL*8 RNAME(75), ENAME, SNAME, SNAME1, SNAME2,
76      *      SNAME3, COMP, IC1(20), IBL, IDOS, BNAME,
77      *      ANAME, TEXT1(2), TEXT2(6), TEXT(2), GNAME(3), NNN
78      REAL*8 THALF, AQ, ALAMB, EQ, DNR, GAP, BREAK, DUR, PE, ETA, TD, WATVEL, PATH
79      LOGICAL AKT, CLAY, INS, SUB, NLINK, NPUNCH, WRITEX
80      LOGICAL LG
81      DIMENSION CA(75), YMAX(10), TIMY(10), DOSMAX(75), TPE(75),
82      *      TSLASK(90), TMI(75), TMA(75), UMAX(75)
83      DATA IUPP/0/
84      DATA TEXT /'GLASS', 'FUEL'/
85      DATA TEXT1 /'POROSITY', 'SPACING' //
86      DATA TEXT2 /'CU. METER', 'METERS' ', 'S/CU. MET', ' ', 'ER'
87      *      ' ' /
88      DATA IDOS/'DOSES'/
89      DATA COMP/'-1'/, IBL/' //
90      DATA ALN2 /0.6931472/
91      805    FORMAT(I2)
92      810    FORMAT(8F10.0)
93      820    FORMAT(A6,4X,F10.0)
94      920    FORMAT(1X,A6,1PE10.2,10E10.2)
95      930    FORMAT('1TIME ',10X,1PE10.2,10E10.2)
96      READ(5,1)NS,J3
97
98      C
99      1    FORMAT(6I5)
100     READ(5,2)BREAK, DUR, DIFF, TONNE
101     2    FORMAT(5E10.2)
102     READ(5,3)PERM, GRAD, SPACE, PATH, CLAY
103     3    FORMAT(4E15.4, L5)
104
105      C
106      IF(CLAY)GO TO 80
107      WATVEL=2.842E9*GRAD*(SPACE*PERM)**(2.0/3.0)
108      DISP=DIFF+5.04E-3*(SPACE*PERM*GRAD)**2/DIFF
109      DISP=DISP*3.1536E7
110      GO TO 90
111     80    WATVEL=3.1536E7*PERM*GRAD/SPACE
112     DISP=DIFF*3.1536E7*SPACE/2.0

```

```

111
112 C
113 C   INITIATE THE DECAY ROUTINE
114 C
115   CALL DECAY(1)
116 C
117   T=BREAK
118 C
119   READ(5,810) TUPP
120   IF (TUPP.EQ.0.) GO TO 20
121 C
122 C   IF REPROCESSING READ THE SPLITFACTORS DEVIATING FROM 1.
123 C
124   READ(5,805) IANT
125   DO 11 I=1, IANT
126   READ(5,820) BNAME, VARDE
127   SF(INTNUK(BNAME))=VARDE
128   11 CONTINUE
129 C
130 C   CALCULATION OF SOURCE STRENGTHS AT THE TIME OF LEACH INCIDENT
131 C
132 20   IF (TUPP.EQ.0.) TUPP=1.E20
133   IF (T.LT.TUPP.OR.IUPP.EQ.2) GO TO 25
134   CALL DEC(TUPP,TUPP,1,J,1)
135   IUPP=2
136 25   IF (IUPP.EQ.0) CALL DEC(T,T,IUPP,J,0)
137   IF (IUPP.EQ.2) CALL DEC(T-TUPP,T,IUPP,J,0)
138   DO 29 L=1, ISL
139   CA(L)=C(L)*TONNE
140 29   CONTINUE
141   WRITE(6,930) T
142   DO 32 I=1, ISL
143   WRITE(6,920) ANAME(I), THALV(I), CA(I)
144 32   CONTINUE
145   REWIND 1
146   JK=2
147   IF(CLAY) JK=1
148   JJ=2
149   IF(TUPP.NE.1.E20) JJ=1
150   WATIM=PATH/WATVEL
151   WRITE(6,41) TEXT(JJ), BREAK, DUR, PATH, WATVEL, WATIM, DISP, TONNE
152   *, PERM, GRAD, TEXT1(JK), SPACE, TEXT2(JK), TEXT2(JK+2), TEXT2(JK+4)
153 41   FORMAT('1) CALCULATION OF RADIONUCLIDE CHAIN MIGRATION FROM AN UNDER
154   *GROUND REPOSITORY', /1X, 74('*') / 'OWASTE FORM: ', A8 / 'OTIME OF LEACH
155   *INCIDENT ', 1PE10.1, ' YEARS AFTER REACTOR DISCHARGE' / ' LEACH DURATI
156   *ON ', 8X, E10.1, ' YEARS' / ' MIGRATION PATH LENGTH ', 1X, E10.1, ' METERS
157   *' / ' GROUNDWATER VELOCITY ', 2X, E10.1, ' METERS/YEAR' / ' GROUNDWATER T
158   *RAVEL TIME', E10.1, ' YEARS' /
159   *
160   * EFFICIENT ', E10.1, ' SQ.METERS/YEAR' / ' INVOLVED WASTE AMOUNT ', 1X, E
161   *10.1, ' TONNES' / ' PERMEABILITY', 11X, E10.1, ' METERS/SECOND' / ' GRADIE
162   *NT ', 14X, E10.1, ' METERS/METER' / 1X, A8, 15X, E10.1, 1X, A8, A8, A8 / )
163   WRITE(6,61)
164 61   FORMAT(1X / ' NUCL.', 5X, 'H.LIFE (Y)', 2X, 'SOURCE (CI)', 1X,
165   * 'RET.', 9X, 'DOSE FACT' / )
166   III=1

```

```

111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166

```

5-CMPACK	PAGE	9		
167	C			167
168	C		STARTING POINT OF TIMESCALE IN SINGLE	168
169	C			169
170			PE=PATH*WATVEL/DISP	170
171			CALL TSING(PE,TH1)	171
172	C			172
173	C		START OF SINGLE LOOP	173
174	C			174
175		100	CONTINUE	175
176			READ(5,44)SNAME,AKT,AREAK,VOLK,DOSE	176
177		44	FORMAT(A6 ,L2/3E10.2)	177
178			IF(SNAME.EQ.COMP)GO TO 200	178
179			IJ=INTNUK(SNAME)	179
180			IF(CLAY)RET(IJ)=1+VOLK*(1-SPACE)/SPACE	180
181			IF(.NOT.CLAY)RET(IJ)=1+AREAK*2./0.0105/	181
182			* (PERM*SPACE)**.333333	182
183			THALF(1)=THALV(IJ)	183
184			AQ(1)=CA(IJ)	184
185			ENAME(1)=SNAME	185
186			EQ(1)=RET(IJ)	186
187			ALAMB(1)=ALN2/THALF(1)	187
188			DOSE2(IJ)=DOSE	188
189			WRITE(6,77)SNAME,THALF(1),AQ(1),EQ(1),DOSE	189
190		77	FORMAT(1H .A6 ,1PE12.2,9E12.2)	190
191			CALL SINGLE(TH1,III,AKT,NS)	191
192			GO TO 100	192
193	C			193
194	C		START OF DOUBLE LOOP	194
195	C			195
196		200	CONTINUE	196
197			WRITE(6,205)	197
198		205	FORMAT('0')	198
199			II=0	199
200			III=2	200
201		210	CONTINUE	201
202			READ(5,5)SNAME1,SNAME2,INS,SUB	202
203		5	FORMAT(A6 .4X,A6 ,2L2)	203
204			IF(SNAME1.EQ.COMP)GO TO 300	204
205			IJ1=INTNUK(SNAME1)	205
206			IJ2=INTNUK(SNAME2)	206
207			IF(RET(IJ1).EQ.0..OR.RET(IJ2).EQ.0.)GO TO 250	207
208			THALF(1)=THALV(IJ1)	208
209			THALF(2)=THALV(IJ2)	209
210			ALAMB(1)=ALN2/THALF(1)	210
211			ALAMB(2)=ALN2/THALF(2)	211
212			AQ(1)=CA(IJ1)	212
213			AQ(2)=CA(IJ2)	213
214			ENAME(1)=SNAME1	214
215			ENAME(2)=SNAME2	215
216			EQ(1)=RET(IJ1)	216
217			EQ(2)=RET(IJ2)	217
218			IF(II)260,260,320	218
219		250	CONTINUE	219
220			WRITE(6,211)SNAME1,SNAME2	220
221		211	FORMAT('XDOUBLE: ',2A8,'NO INPUT PROVIDED IN SINGLE-LOOP')	221
222			GO TO 210	222

316-CMPACK	PAGE	10		
223	260	CONTINUE		223
224	C			224
225	C	TWO MEMBER CHAINS WITH EQUAL RETENTION COEFFICIENTS CALCULATED BY		225
226	C	SINGLE.		226
227	C			227
228		IF(DABS(EQ(1)-EQ(2))-0.01*EQ(1))261,261,265		228
229	261	CONTINUE		229
230		CALL SINGLE(TH1,III,AKT,NS)		230
231		WRITE(6,262)ENAME(1),ENAME(2)		231
232	262	FORMAT(' EQUAL: ',2X,2A8)		232
233		GO TO 210		233
234	265	CONTINUE		234
235		WRITE(6,8)((ENAME(I),EQ(I)),I=1,2)		235
236	8	FORMAT(1H0,42HATTEMPT TO TREAT UNEQUAL COEFF. AS EQUAL: ,A6 ,4H		236
237		*K= ,1PE10.2/1H ,42X,A6 ,4H K= ,E10.2/1H ,34HCHAIN DELETED,EXECUT		237
238		*ION CONTINUING)		238
239		GO TO 210		239
240	300	CONTINUE		240
241		IF(II-1)310,350,350		241
242	310	CONTINUE		242
243		WRITE(6,205)		243
244		II=1		244
245		GO TO 210		245
246	C			246
247	C	CALL FOR DOUBLE		247
248	C			248
249	320	CONTINUE		249
250		CALL DOUBLE(INS,SUB,J3)		250
251		WRITE(6,330)ENAME(1),ENAME(2)		251
252		IF(.NOT.INS)WRITE(6,340)LG(1),LG(2)		252
253	340	FORMAT('+',T28,'TERMS OMITTED IN BWD2: 2(' ,L1,'), 4(' ,L1,'))'		253
254	330	FORMAT(' DOUBLE: ',1X,2A8)		254
255		GO TO 210		255
256	350	CONTINUE		256
257	C			257
258	C	END OF DOUBLE LOOP, START OF TRIPLE LOOP		258
259	C			259
260		WRITE(6,205)		260
261		III=3		261
262		II=0		262
263	400	CONTINUE		263
264		READ(5,16)SNAME1,SNAME2,SNAME3,INS		264
265	16	FORMAT(A6 ,4X,A6 ,4X,A6 ,L2)		265
266		IF(SNAME1.EQ.COMP)GO TO 500		266
267		IJ1=INTNUK(SNAME1)		267
268		IJ2=INTNUK(SNAME2)		268
269		IJ3=INTNUK(SNAME3)		269
270		IF(RET(IJ1).EQ.0..OR.RET(IJ2).EQ.0..OR.RET(IJ3).EQ.0.)GO TO 452		270
271		THALF(1)=THALV(IJ1)		271
272		THALF(2)=THALV(IJ2)		272
273		THALF(3)=THALV(IJ3)		273
274		ALAMB(1)=ALN2/THALF(1)		274
275		ALAMB(2)=ALN2/THALF(2)		275
276		ALAMB(3)=ALN2/THALF(3)		276
277		AQ(1)=CA(IJ1)		277
278		AQ(2)=CA(IJ2)		278

```

279 AQ(3)=CA(IJ3)
280 ENAME(1)=SNAME1
281 ENAME(2)=SNAME2
282 ENAME(3)=SNAME3
283 EQ(1)=RET(IJ1)
284 EQ(2)=RET(IJ2)
285 EQ(3)=RET(IJ3)
286 IF(II.EQ.0)GO TO 455
287 C
288 C CALL FOR TRIPLE
289 C
290 CALL TRIPLE(INS,J3)
291 WRITE(6,450) ENAME
292 IF(.NOT.INS)WRITE(6,440)LG(6),LG(5),LG(5),LG(4),LG(6),LG(3),LG(4),
293 * LG(5),LG(1),LG(3),LG(5),LG(1),LG(2),LG(3)
294 440 FORMAT('+.T34,TERMS OMITTED IN BWD3: 5('.,2L1,'), 6('.,2L1,'),',',
295 * 7('.,L1,'), 8('.,L1,'), 9('.,2L1,'), 10('.,L1,'), 11('.,L1,'), 12('.,L
296 *1,'), 13('.,L1,'), 14('.,L1,'), 15('.,L1,')')
297 450 FORMAT(' TRIPLE: ',1X,3A8)
298 GO TO 400
299 452 CONTINUE
300 WRITE(6,411)SNAME1,SNAME2,SNAME3
301 411 FORMAT('XTRIPLE: ',3A8,'NO INPUT PROVIDED IN SINGLE LOOP')
302 GO TO 400
303 C
304 C THREE MEMBER CHAINS WITH EQUAL RETENTION COEFFICIENTS CALCULATED
305 C BY SINGLE
306 C
307 455 CONTINUE
308 IF(EQ(1).NE.EQ(2).OR.EQ(1).NE.EQ(3))GO TO 460
309 CALL SINGLE(TH1,III,AKT,NS)
310 WRITE(6,451)ENAME
311 451 FORMAT(' EQUAL: ',3A8)
312 GO TO 400
313 460 CONTINUE
314 WRITE(6,461)ENAME,EQ
315 461 FORMAT('OATTEMPT TO TREAT UNEQUAL COEFFICIENTS AS EQUAL IN THE CHA
316 *IN: ',3(4X,A6)/T58,'EQ= ',3F10.0)
317 GO TO 400
318 C
319 500 CONTINUE
320 IF(II.EQ.1)GO TO 508
321 II=1
322 WRITE(6,205)
323 GO TO 400
324 508 CONTINUE
325 C
326 C END OF TRIPLE LOOP
327 C
328 DO 513 JJ=13,15
329 ENDFILE JJ
330 BACKSPACE JJ
331 REWIND JJ
332 513 CONTINUE
333 C
334 C THE "SPECIAL APPROXIMATION LOOP".

```

279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334

335	C		335
336		REWIND 16	336
337		WRITE(6,205)	337
338	11000	CONTINUE	338
339		DO 11001 K=1,3	339
340		GNAME(K)=IBL	340
341	11001	ENAME(K)=IBL	341
342		READ(5,11002)GNAME,IAPPR,ENAME	342
343	11002	FORMAT(3A8,16,3A8)	343
344		IF(GNAME(1).EQ.COMP)GO TO 12000	344
345		IF(IAPPR.EQ.1.AND.ENAME(1).NE.GNAME(2))GO TO 11300	345
346		IFILE=3	346
347		DO 11100 K=1,3	347
348		IF(ENAME(K).NE.IBL)GO TO 11100	348
349		IFILE=K-1	349
350		GO TO 11200	350
351	11100	CONTINUE	351
352	11200	IF (IFILE.EQ.0) GO TO 11300	352
353		IF(IAPPR.EQ.1)GO TO 11250	353
354		IF (ENAME(IFILE).NE.GNAME(1))GO TO 11300	354
355		IJ=INTNUK(ENAME(1))	355
356		Q1=CA(IJ)*THALV(IJ)	356
357		IF(Q1.LE.0.0)GO TO 11400	357
358		IJ=INTNUK(GNAME(2))	358
359		Q2=CA(IJ)*THALV(IJ)	359
360		FACT=Q2/Q1	360
361		IF(GNAME(3).EQ.IBL)GO TO 11250	361
362		IJ=INTNUK(GNAME(3))	362
363		Q2=CA(IJ)*THALV(IJ)	363
364		FACT=FACT+Q2/Q1	364
365		IF(FACT.LE.0)GO TO 11500	365
366	11250	CONTINUE	366
367		CALL SPEC(GNAME,ENAME,IFILE,IAPPR,FACT,&11000)	367
368		WRITE(6,11251)GNAME,ENAME,IAPPR	368
369	11251	FORMAT(' SPEC:',3X,3A8,5X,3A8,' IAPPR= ',I3)	369
370		GO TO 11000	370
371	11300	CONTINUE	371
372		WRITE(6,11301)ENAME,GNAME,IAPPR	372
373	11301	FORMAT('0SORRY I CANNOT DO THE IMPOSSIBLE,LOOK FOR YOURSELF: ENAME	373
374		*= ',3A8/55X,'GNAME= ',3A8/55X,' IAPPR= ',I3/)	374
375		GO TO 11000	375
376	11400	CONTINUE	376
377		WRITE(6,11401)(GNAME(I),I=1,2)	377
378		IF(GNAME(3).NE.IBL)WRITE(6,11402)GNAME(3)	378
379		WRITE(6,11403)Q1,ENAME(1)	379
380	11401	FORMAT('0THE CHAIN ',A6,' FROM ',A6)	380
381	11402	FORMAT('+' ,29X,' AND ',A6)	381
382	11403	FORMAT(' WAS NOT CALCULATED BY SPEC BECAUSE OF ',1PE10.2,'--INVENTO	382
383		*RY FOR ',A6)	383
384		GO TO 11000	384
385	11500	CONTINUE	385
386		WRITE(6,11501)GNAME,ENAME,FACT	386
387	11501	FORMAT('0THE CHAIN ',3A8,' WAS NOT CALCULATED BY SPEC BECAUSE '/' T	387
388		*HE CHAIN ',3A8,' GAVE FACT= ',1PE10.2/)	388
389		GO TO 11000	389
390	12000	CONTINUE	390

```

391 ENDFILE 16
392 BACKSPACE 16
393 REWIND 16
394 C
395 C END OF "SPECIAL APPROXIMATION LOOP".
396 C
397 IF(NLINK)RETURN
398 C
399 C START OF PEAK COMBINATION LOOP
400 C
401 DO 600 I=1,ISL
402 IJ=I
403 TMA(IJ)=0.0
404 TMI(IJ)=0.0
405 TPE(IJ)=0.0
406 UMAX(IJ)=0.0
407 TMAX=0.0
408 TMIN=0.0
409 NN=ICONTR(IJ)
410 DO 505 J=1,20
411 IC1(J)=IBL
412 505 CONTINUE
413 IF(NN.EQ.0)GO TO 600
414 IF(NN.EQ.1.AND.CMAX(IJ,1).EQ.0)GO TO 590
415 IF(NN.EQ.1)GO TO 570
416 C
417 DO 510 J=1,NN
418 YMAX(J)=CMAX(IJ,J)
419 TIMY(J)=TPEAK(IJ,J)
420 TSLASK(3*J-2)=TSTART(IJ,J)
421 TSLASK(3*J-1)=TPEAK(IJ,J)
422 TSLASK(3*J)=TEND(IJ,J)
423 510 CONTINUE
424 C
425 TMAX=0.0
426 II=3*NN
427 DO 530 J=3,II,3
428 IF(YMAX(J/3).LT.1.E-20)GO TO 530
429 IF(TSLASK(J).GT.TMAX)TMAX=TSLASK(J)
430 530 CONTINUE
431 TMIN=TMAX
432 DO 540 J=1,II,3
433 IF(YMAX((J+2)/3).LT.1.E-20)GO TO 540
434 IF(TSLASK(J).LT.TMIN.AND.TSLASK(J).GE.10.)TMIN=TSLASK(J)
435 540 CONTINUE
436 IF(TMIN.EQ.TMAX)GO TO 590
437 IF(TMIN.LE.0.0.OR.TMAX.LE.0.0)GO TO 590
438 TMA(IJ)=TMAX
439 TMI(IJ)=TMIN
440 CALL TSCALE(YMAX,TIMY,NN,TMAX,TMIN,U,NT,&590)
441 CALL CHAIN(ANAME(I),NT,IC1,I2)
442 RMAX=0.0
443 DO 564 J=1,NT
444 IF(Y(J).LE.RMAX)GO TO 564
445 RMAX=Y(J)
446 TPE(IJ)=U(J)

```

```

391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446

```

```

447 564 CONTINUE 447
448 UMAX(IJ)=RMAX 448
449 IF(NPUNCH)GO TO 8000 449
450 WRITE(7,561)ANAME(I),NT,U(1),U(NT),RMAX 450
451 561 FORMAT(A6 ,4X,I4,1X,1PE15.5,2E15.5) 451
452 WRITE(7,1000)((U(J),Y(J)),J=1,NT) 452
453 1000 FORMAT(1PE9.2,7E9.2) 453
454 8000 CONTINUE 454
455 CALL PLOT1(ANAME(I),IC1,I2,NT,RMAX,WRITEX) 455
456 WRITE(17)ANAME(I),NT 456
457 WRITE(17)(U(J),Y(J),J=1,NT) 457
458 I2=I2-1 458
459 GO TO 600 459
460 570 READ(13,END=590)ENAME,ISTEP 460
461 IF(ENAME(1).EQ.ANAME(I))GO TO 575 461
462 IF(ISTEP.EQ.0)GO TO 570 462
463 READ(13)(U(J),Y(J),J=1,ISTEP) 463
464 GO TO 570 464
465 575 CONTINUE 465
466 IF(ISTEP.EQ.0)GO TO 590 466
467 READ(13)(U(J),Y(J),J=1,ISTEP) 467
468 TMI(IJ)=U(1) 468
469 TMA(IJ)=U(ISTEP) 469
470 RMAX=CMAX(IJ,1) 470
471 TPE(IJ)=TPEAK(IJ,1) 471
472 UMAX(IJ)=RMAX 472
473 IF(NPUNCH)GO TO 585 473
474 WRITE(7,561)ANAME(I),ISTEP,U(1),U(ISTEP),RMAX 474
475 WRITE(7,1000)((U(J),Y(J)),J=1,ISTEP) 475
476 585 CONTINUE 476
477 I2=1 477
478 CALL PLOT1(ANAME(I),IC1,I2,ISTEP,RMAX,WRITEX) 478
479 WRITE(17)ENAME(1),ISTEP 479
480 WRITE(17)((U(J),Y(J)),J=1,ISTEP) 480
481 GO TO 599 481
482 590 CONTINUE 482
483 WRITE(6,591)ANAME(I),TMIN,TMAX 483
484 591 FORMAT('0WHILE EXECUTING PEAK COMBINATION LOOP OF MAIN PROGRAM NO 484
485 *PEAKS WHERE FOUND FOR ',A6 / 485
486 * ' TMIN= ',1PE12.4,' TMAX= ',E12.4) 486
487 599 CONTINUE 487
488 REWIND13 488
489 600 CONTINUE 489
490 C 490
491 C END OF PEAK COMBINATION LOOP 491
492 C 492
493 C ENDFILE 17 493
494 C BACKSPACE 17 494
495 C REWIND 17 495
496 C 496
497 C SUMMARY PRINTOUT LOOP 497
498 C 498
499 C WRITE(6,13000) 499
500 13000 FORMAT('1SUMMARY OF NUCLIDE CONTRIBUTIONS'/1X,34('*')// 500
501 * ' NUCLIDE',6(4X,'TPEAK',5X,'CMAX',2X)// 501
502 DO 14000 J=1,ISL 502

```

```

503 IF(ICONTR(J).EQ.0) GO TO 14000 503
504 NN=ICONTR(J) 504
505 WRITE(6,13100)ANAME(J),(TPEAK(J,K),CMAX(J,K),K=1,NN) 505
506 13100 FORMAT((1X,A7,1PE9.2,E9.2,2X,5(2E9.2,2X))/8X,4(2E9.2,2X)) 506
507 14000 CONTINUE 507
508 WRITE(6,15000) 508
509 15000 FORMAT('1SUMMARY OF NUCLIDE INFLOWS AND APPROXIMATIVE DOSES'/1X, 509
510 *50('*')// ' NUCLIDE',4X,'TPEAK',7X,'CMAX',8X,'DOSEMAX'//) 510
511 DO 16000 J=1,ISL 511
512 NN=ICONTR(J) 512
513 DOSMAX(J)=UMAX(J)*DOSE2(J) 513
514 IF(NN.EQ.0)GO TO 16000 514
515 WRITE(6,15100)ANAME(J),TPE(J),UMAX(J),DOSMAX(J) 515
516 15100 FORMAT(1X,A7,1P3E12.2) 516
517 16000 CONTINUE 517
518 C 518
519 C END OF SUMMARY PRINTOUT LOOP 519
520 C 520
521 C TMATOT=0.0 521
522 C 522
523 C CALCULATION OF DOSES 523
524 C 524
525 DO 711 I=1,ISL 525
526 IF(TMA(I).GT.TMATOT.AND.TMA(I).GT.10.0) TMATOT=TMA(I) 526
527 711 CONTINUE 527
528 TMITOT=TMATOT 528
529 DO 712 I=1,ISL 529
530 IF(TMI(I).LT.TMITOT.AND.TMI(I).GT.10.0) TMITOT=TMI(I) 530
531 712 CONTINUE 531
532 CALL TSCALE(DCSMAX,TPE,ISL,TMATOT,TMITOT,U,IS,&850) 532
533 DO 701 I=1,IS 533
534 Y(I)=0.0 534
535 701 CONTINUE 535
536 710 READ(17,END=800)NNN,ISTEP 536
537 READ(17)(TIME(I),DOSE1(I),I=1,ISTEP) 537
538 IJ=INTNUK(NNN) 538
539 DO 750 I=1,ISTEP 539
540 DOSE1(I)=DOSE1(I)*DOSE2(IJ) 540
541 750 CONTINUE 541
542 CALL INTPOL(IS,ISTEP) 542
543 GO TO 710 543
544 800 CONTINUE 544
545 RMAX=0.0 545
546 DO 801 I=1,IS 546
547 IF(Y(I).GT.RMAX)RMAX=Y(I) 547
548 801 CONTINUE 548
549 CALL PLOT1(IDOS,IC1,0,IS,RMAX,WRITEX) 549
550 RETURN 550
551 850 CONTINUE 551
552 WRITE(6,860) 552
553 860 FORMAT('0WHILE EXECUTING PEAK COMBINATION LOOP OF MAIN PROGRAM NO 553
554 *DOSES WHERE FOUND') 554
555 RETURN 555
556 END 556
557 SUBROUTINE DECAY(IPRINT) 557
558 C 558

```

```

559 C THIS ROUTINE IS FOR CALCULATING THE NUCLIDE INVENTORY AT THE
560 C TIME OF LEACH INCIDENT, WITH OR WITHOUT REPROCESSING
561 C
562 REAL*16 TEXP1,QLOG
563 REAL*16 ALN2,T,THALV(75),CNOLL(75),ALAMB(75),ANNOLL(75)
564 COMMON/HALVA/ THALV,RET(75)
565 REAL*8 SLUT,BLANK
566 DIMENSION IDENT(40),IFISS(40),ICHAIN(6,3,4),JCHAIN(6,3,4)
567 REAL*8 FISS(40),CHAIN(6,3,4)
568 COMMON/DEY1/ FISS,CHAIN,IFI,ICH
569 REAL*16 ETEXP(75),AN(75),C(75)
570 REAL*8 ANAME(76)
571 COMMON/DEY2/ ETEXP,AN,C,ANAME,SF(75),NUK(75),IN,ISL
572 REAL*16 AL(6),ET(6),ANN(6)
573 COMMON/DEY3/ AL,ET,ANN
574 DATA SLUT/'-1 ' /, BLANK/' ' /, JCHAIN/72*1/
575 DATA ALAMB/75*0./,ANNOLL/75*0./
576 DATA ALN2/.6931471806000/
577 800 FORMAT(A6,2E10.2)
578 810 FORMAT(20A4)
579 900 FORMAT(' ' INITIATION OF DECAY WITH'//1X,20A4/1X,20A4//
580 1 15X,'NUCL. ',3X,' THALF',7X,' CO ' ,7X,' ALAMB',7X,' ANZERO'//
581 2 75(I11,4X,A6.4(2X,1PE10.3)/))
582 910 FORMAT(' ' FISSION PRODUCTS:'//30(I11,4X,A6/))
583 920 FORMAT(' ' DECAY CHAINS:'//4(I11,4X,3(A6,3X)/
584 1 5(15X,A6,3X,A6,3X,A6/)/))
585 930 FORMAT(' ' CHANGED ACTIVITIES AT THE TIME OF REPROCESSING=' ,
586 1 F14.3 , ' YR'///' NR FNR NUKLID SF',8X,' CZERO',8X,' ANZERO'//
587 940 FORMAT(1X,I2,3X,I2,3X,A6,3X,F6.4,2(3X,1PE10.3))
588 950 FORMAT(' ' ACTIVITIES AT TIME=' ,F14.3, ' YEARS'///
589 1 ' NR FNR NUKLIDE CURIE'//
590 960 FORMAT(1X,I2,3X,I2,3X,A6,2X,1PE10.3)
591 996 FORMAT('+' ,T36,A6, ' NOT IN DATASET, ICHAIN=' ,3(I2,1X), ' ***'//
592 1 75(10X,A6/))
593 997 FORMAT('+' ,T36,A6, ' NOT IN DATASET, IFISS=' ,I2, ' ***'//
594 1 75(10X,A6/))
595 998 FORMAT('+' ,T36, ' NO ENDMARK IN DATASET ***'//75(10X,A6/))
596 999 FORMAT('1'///10X, ' *** INITIATION OF DECAY:' )
597 DO 10 I=1,76
598 READ(1,800) ANAME(I),THALV(I),CNOLL(I)
599 IF (ANAME(I).EQ.SLUT) GO TO 20
600 10 CONTINUE
601 WRITE(6,999)
602 WRITE(6,998) ANAME
603 STOP
604 20 READ(1,810) IDENT
605 IBL=I
606 ANAME(IBL)=BLANK
607 ISL=IBL-1
608 DO 50 I=1,IFI
609 DO 40 IX=1,ISL
610 IF (FISS(I).NE.ANAME(IX)) GO TO 40
611 IFISS(I)=IX
612 NUK(I)=IX
613 GO TO 50
614 40 CONTINUE

```

```

559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614

```

316-CMPACK	PAGE	17		
615			WRITE(6,999)	615
616			WRITE(6,997) FISS(I),I,(ANAME(IX),IX=1,ISL)	616
617			STOP	617
618	50		CONTINUE	618
619			IN=IFI	619
620			DO 100 I=1,ICH	620
621			DO 100 K=1,6	621
622			DO 70 J=1,3	622
623			DO 60 IX=1,IBL	623
624			IXX=IBL-IX+1	624
625			IF (CHAIN(K,J,I).NE.ANAME(IXX)) GO TO 60	625
626			ICHAIN(K,J,I)=IXX	626
627			IF (IXX.EQ.IBL) GO TO 70	627
628			DO 55 IY=IFI,IN	628
629			IF (NUK(IY).EQ.IXX) GO TO 70	629
630	55		CONTINUE	630
631			IN=IN+1	631
632			NUK(IN)=IXX	632
633			GO TO 70	633
634	60		CONTINUE	634
635			WRITE(6,999)	635
636			WRITE(6,996) CHAIN(K,J,I),K,J,I,(ANAME(IX),IX=1,ISL)	636
637			STOP	637
638	70		CONTINUE	638
639			DO 90 J=1,2	639
640			DO 80 JX=J,3	640
641			IF (J.EQ.JX) GO TO 80	641
642			IF (ICHAIN(K,J,I).NE.ICHAIN(K,JX,I)) GO TO 80	642
643			JCHAIN(K,J,I)=JCHAIN(K,J,I)+1	643
644			JCHAIN(K,JX,I)=JCHAIN(K,JX,I)+1	644
645	80		CONTINUE	645
646	90		CONTINUE	646
647	100		CONTINUE	647
648			DO 105 I=1,IN	648
649			IND=NUK(I)	649
650			ALAMB(IND)=ALN2/THALV(IND)	650
651			ANNOLL(IND)=CNOLL(IND)/ALAMB(IND)	651
652	105		CONTINUE	652
653			IF (IPRINT.NE.1) RETURN	653
654			WRITE(6,900) IDENT,(I,ANAME(I),THALV(I),CNOLL(I),ALAMB(I),	654
655	1		ANNOLL(I),I=1,ISL)	655
656			WRITE(6,910) (I,FISS(I),I=1,IFI)	656
657			WRITE(6,920) (I,((CHAIN(K,J,I),J=1,3),K=1,6),I=1,ICH)	657
658			RETURN	658
659			ENTRY DEC(T,TX,IUPP,ISTEG,IPRINT)	659
660			DO 110 I=1,ISL	660
661			C(I)=0.	661
662			AN(I)=0.	662
663			ETEXP(I)=TEXP1(-T*ALAMB(I))	663
664	110		CONTINUE	664
665			DO 120 I=1,IFI	665
666			INDEX=IFISS(I)	666
667			AN(INDEX)=ANNOLL(INDEX)*ETEXP(INDEX)	667
668	120		CONTINUE	668
669			DO 170 I=1,ICH	669
670			DO 160 J=1,3	670

671		DO 125 K=1,6	671
672		AL(K)=0.	672
673		ET(K)=0.	673
674		ANN(K)=0.	674
675	125	CONTINUE	675
676		DO 130 K=1,6	676
677		IF (ICHAIN(K,J,I).EQ.IBL) GO TO 130	677
678		KST=K	678
679		GO TO 140	679
680	130	CONTINUE	680
681		GO TO 160	681
682	140	DO 150 K=KST,6	682
683		KX=K-KST+1	683
684		INDEX=ICHAIN(K,J,I)	684
685		AL(KX)=ALAMB(INDEX)	685
686		ET(KX)=ETEXP(INDEX)	686
687		ANN(KX)=ANNOLL(INDEX)/FLOAT(JCHAIN(K,J,I))	687
688		AN(INDEX)=AN(INDEX)+BATE1(KX,ANAME(INDEX),T)	688
689	150	CONTINUE	689
690	160	CONTINUE	690
691	170	CONTINUE	691
692		DO 180 I=1,IN	692
693		IND=NUK(I)	693
694		C(IND)=AN(IND)*ALAMB(IND)	694
695	180	CONTINUE	695
696		IF (IUPP.EQ.1) GO TO 186	696
697		IF (IPRINT.EQ.0) RETURN	697
698		WRITE(6,950) TX	698
699		DO 185 I=1,IN	699
700		IND=NUK(I)	700
701		WRITE(6,960) I,IND,ANAME(IND),C(IND)	701
702	185	CONTINUE	702
703		RETURN	703
704	186	DO 190 I=1,IN	704
705		IND=NUK(I)	705
706		CNOLL(IND)=SF(IND)*C(IND)	706
707		ANNOLL(IND)=CNOLL(IND)/ALAMB(IND)	707
708	190	CONTINUE	708
709		IF (IPRINT.EQ.0) RETURN	709
710		WRITE(6,930) TX	710
711		DO 200 I=1,IN	711
712		IND=NUK(I)	712
713		WRITE(6,940) I,IND,ANAME(IND),SF(IND),CNOLL(IND),ANNOLL(IND)	713
714	200	CONTINUE	714
715		RETURN	715
716		END	716
<hr/>			
717		FUNCTION BATE1(INR,ANAME,T)	717
718	C		718
719	C	THIS ROUTINE CALCULATES THE DECAY OF THE INITIAL NUCLIDE INVENTORY	719
720	C		720
721		REAL*16 TERM1,BATSUM,T	721
722		REAL*8 ANAME	722
723		REAL*16 AL(6),ET(6),ANN(6)	723
724		COMMON/DEY3/ AL,ET,ANN	724
725		REAL*16 FAKT(6),TERM(6,6),DUMMY(42)	725
726		EQUIVALENCE (FAKT(1),DUMMY(1)),(TERM(1),DUMMY(7))	726

```

316-CMPACK PAGE 19
727 900 FORMAT(///1X,A6,10X,'BATE=',1PE14.7// ' AL= ',6(3X,E14.7)// ' ET= '
728 1 ',6(3X,E14.7)// ' ANN= ',6(3X,E14.7)// ' FAKT=',6(3X,E14.7)//
729 2 ' TERM=',6(3X,E14.7)/5(9X,E14.7,5(3X,E14.7)/))
730 910 FORMAT(1X,I5,2X,3(Q37.30))
731 920 FORMAT(// ' *** TERM1=',1PE14.7, ' I=',I1, ' T=',0PF14.3,3X,A6//)
732 930 FORMAT('1 ')
733 999 FORMAT(/////10X,'*** BATE HAR ANROPATS MED',A6,' OCH INR=',
734 1 I2,' ***')
735 IF (INR.GT.0.AND.INR.LT.7) GC TC 10
736 WRITE(6,999) ANAME,INR
737 STOP
738 10 BATSUM=0.
739 DO 20 I=1,42
740 DUMMY(I)=0.
741 20 CONTINUE
742 DO 70 I=1,INR
743 FAKT(I)=ANN(I)
744 IXSL=I-1
745 IF (IXSL.EQ.0) GO TO 40
746 DO 30 IX=1,IXSL
747 FAKT(IX)=FAKT(IX)*AL(IXSL)
748 30 CONTINUE
749 40 DO 60 J=I,INR
750 TERM(I,J)=ET(J)
751 DO 50 JX=I,INR
752 IF (J.NE.JX) TERM(I,J)=TERM(I,J)/(AL(JX)-AL(J))
753 50 CONTINUE
754 60 CONTINUE
755 70 CONTINUE
756 DO 90 I=1,INR
757 TERM1=0.
758 DO 80 J=I,INR
759 TERM1=TERM1+TERM(I,J)
760 80 CONTINUE
761 BATSUM=BATSUM+FAKT(I)*TERM1
762 90 CONTINUE
763 BATE1=BATSUM
764 RETURN
765 END
766 SUBROUTINE SINGLE(TH1,III,AKT,NS)
767 C
768 C THIS ROUTINE CALCULATES THE RELEASE RATES FOR SINGLE-NUCLIDES
769 C
770 COMMON/PRINT/ MIGOUT
771 COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
772 * GAP(6),T0,DUR,PE,ETA,TD,V0,PATH
773 COMMON/C2/ NSTEP,BRD(304),BRND(304),TIME(304),THETA(304),
774 * TBEGIN,TFIN,TTOPP,BRPEAK,DIND
775 COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
776 * CMAX(75,10),DOSE2(75)
777 COMMON /TMAXND/ TND1(75),TND2(75)
778 REAL*8 ENAME,B1,B5,B2,B6,AM1,G1,G2,G3,B9,TDERFC
779 REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,T0,DUR,PE,ETA,TD,V0,PATH
780 LOGICAL AKT,MIGOUT
781 J1=0
782 J2=0

```

```

727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782

```

783	R=1.0/EQ(1)	783
784	V1=V0*R	784
785	IPSEUD=0	785
786	DNR(1)=ALAMB(1)*PATH/V0	786
787	AN1=AQ(1)	787
788	BRD(1)=0.0	788
789	BRD(2)=0.0	789
790	AL1=ALAMB(1)	790
791	G3=V1*DUR/PATH	791
792	GAP(1)=AQ(1)/AL1	792
793	GAP(4)=AQ(1)*TEXP(-AL1*DUR)/AL1	793
794	IF(III.LT.2)GO TO 20	794
795	CALL VECTOR(III,IPSEUD)	795
796	20 CONTINUE	796
797	TFIN=T0+PATH/V1*(2-TH1)+DUR	797
798	T1=TH1*PATH/V1	798
799	TBEGIN=T0+T1	799
800	TTOPP=T0+T1	800
801	TTOPND=PATH/V1+T0	801
802	NSTEP=0.0	802
803	BRPEAK=0.0	803
804	PEAKND=0.0	804
805	DIND=0.0	805
806	IF(AQ(1).LT.1.E-15)GO TO 3740	806
807	IF((TBEGIN-T0)*ALAMB(1).GT.40.0)GO TO 3740	807
808	IF(AKT)GO TO 370	808
809	S3=(TFIN/TBEGIN)**(1.0/FLOAT(NS-1))	809
810	TIME(1)=TBEGIN	810
811	THETA(1)=T1*V0/PATH	811
812	I=2	812
813	350 CONTINUE	813
814	TIME(I)=S3*TIME(I-1)	814
815	IF(TIME(I).LT.TTOPND.OR.TIME(I-1).GE.TTOPND)GO TO 360	815
816	TIME(I)=TTOPND	816
817	360 CONTINUE	817
818	THETA(I)=(TIME(I)-T0)*V0/PATH	818
819	I=I+1	819
820	IF(I.GT.NS) GO TO 390	820
821	GO TO 350	821
822	370 S1=(TFIN-TBEGIN)/FLOAT(NS)	822
823	DO 380 I=2,NS	823
824	TIME(I)=TIME(I-1)+S1	824
825	380 THETA(I)=(TIME(I)-T0)*V0/PATH	825
826	390 CONTINUE	826
827	DO 750 J=1,NS	827
828	G2=THETA(J)*R	828
829	G1=G2-1.0D0	829
830	BLJ=AL1*(TIME(J)-T0)	830
831	B1=0.5*DSQRT(PE/G2)*G1	831
832	B9=G2-G3	832
833	IF(B9)540,540,550	833
834	540 B9=1.D-20	834
835	550 B2=0.5*DSQRT(PE/B9)*(G1-G3)	835
836	B5=.5*DSQRT(PE/G2)*(G1+2.0D0)	836
837	B6=.5*DSQRT(PE/B9)*(G1+2.0D0-G3)	837
838	AM1=TDERFC(-B1)+TEXP(PE)*TDERFC(B5)	838

```

839 IF(G2.GT.G3)AM1=AM1-(TDERFC(-B2)+TEXP(PE)*TDERFC(B6))
840 F1=AM1/2.0
841 IF(III-1)621,621,622
842 621 CONTINUE
843 IF(1.E50*F1/DUR*AN1.LE.1.E-03)GO TO 619
844 BRD(J)=TEXP(DLOG(F1*AN1/DUR)-BL1)
845 GO TO 625
846 622 BRD(J)=BATE(TIME(J),TO,III)/DUR*F1
847 GO TO 625
848 619 CONTINUE
849 BRD(J)=0.0
850 625 CONTINUE
851 IF(G1.LT.0.0.OR.G1-G3.GT.0.0)GO TO 700
852 IF(III-1)680,680,685
853 680 CONTINUE
854 BRND(J)=AN1/DUR*TEXP(-BL1)
855 GO TO 710
856 685 BRND(J)=BATE(TIME(J),TO,III)/DUR
857 GO TO 710
858 700 BRND(J)=0.0
859 710 IF(BRD(J-1).LT.1.E-15.AND.BRD(J).GE.1.E-15.AND.J1.EQ.0)J1=J
860 IF(BRD(J-1).GE.1.E-15.AND.BRD(J).LT.1.E-15)J2=J-1
861 IF(BRD(J).LE.BRPEAK)GO TO 720
862 BRPEAK=BRD(J)
863 TTOPP=TIME(J)
864 720 CONTINUE
865 IF(BRND(J).LE.PEAKND)GO TO 730
866 PEAKND=BRND(J)
867 730 CONTINUE
868 IF(J2.GT.0)GO TO 760
869 750 CONTINUE
870 760 CONTINUE
871 3282 IF(PEAKND)3285,3285,3283
872 3283 DINO =BRPEAK/PEAKND
873 GO TO 3286
874 3285 DINO =0.0
875 3286 CONTINUE
876 IF(J1.EQ.0.AND.J2.EQ.0)GO TO 3740
877 IF(J1.GT.0.AND.J2.EQ.0)J2=NS
878 IF(J1.EQ.0)J1=1
879 TBEGIN=TIME(J1)
880 TFIN=TIME(J2)
881 NSTEP=J2-J1+1
882 IF(J1.EQ.1)GO TO 3740
883 DO 3710 I=1,NSTEP
884 II=J1+I-1
885 TIME(I)=TIME(II)
886 THETA(I)=THETA(II)
887 BRD(I)=BRD(II)
888 BRND(I)=BRND(II)
889 3710 CONTINUE
890 3740 CONTINUE
891 ID=III+12
892 WRITE(ID)ENAME,NSTEP
893 IF(NSTEP.LE.0)GO TO 3750
894 WRITE(ID)(TIME(I),BRD(I),I=1,NSTEP)

```

```

839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894

```

```

895 3750 IJ=INTNUK(ENAME(III))
896 ICONTR(IJ)=ICONTR(IJ)+1
897 NN=ICONTR(IJ)
898 TSTART(IJ,NN)=TBEGIN
899 TEND(IJ,NN)=TFIN
900 TPEAK(IJ,NN)=TTOPP
901 CMAX(IJ,NN)=BRPEAK
902 IF(.NOT.MIGOUT)RETURN
903 TND1(IJ)=TTOPND
904 TND2(IJ)=TTOPND+DUR
905 NCHAIN=III
906 CALL PRINTX(NCHAIN)
907 RETURN
908 END
909 SUBROUTINE DOUBLE(INS,SUB,J3)
910 C
911 C THIS ROUTINE CALCULATES THE RELEASE RTTES FOR TWO-NUCLIDE CHAINS
912 C
913 COMMON/PRINT/ MIGOUT
914 COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
915 * GAP(6),TO,DUR,PE,ETA,TD,VO,PATH
916 COMMON/C2/ IS,BAND(304),BANDND(304),TIME(304),THETA(304),
917 * TBEGIN,TFIN,TTOPP,BMAX,DINO
918 COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
919 * CMAX(75,10),DOSE2(75)
920 COMMON/HALVA/ THALV(75),RET(75)
921 COMMON /LGTEST/ LG(6)
922 COMMON /XXX/ XX5,XX6,XX7,XX19,XX29,XX39
923 REAL*16 THALV
924 REAL*8 XX5,XX6,XX7,XX19,XX29,XX39
925 REAL*8 SWD1,SWD2,SWD3,SNOD1,SNOD2,SNOD3
926 REAL*8 ENAME,PS1,PS2,PS3
927 REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,TO,DUR,PE,ETA,TD,VO,PATH
928 LOGICAL MIGOUT,INSCON,INS,SUB
929 LOGICAL LDISP,LNDISP
930 LOGICAL LG
931 DIMENSION Q(15)
932 DATA PS1/'U 238'/,PS2/'TH230'/,PS3/'U 234'/
933 NCHAIN=2
934 INSCON=INS
935 D9=0.0
936 DINO=0.0
937 IPSEUD=0
938 AK=DMAX1(EQ(1),EQ(2))
939 AL=DMIN1(EQ(1),EQ(2))
940 BMAX=0.0
941 IF(ENAME(1).NE.PS1.OR.ENAME(2).NE.PS2)GO TO 100
942 IPSEUD=1
943 100 CONTINUE
944 CALL VECTOR(NCHAIN,IPSEUD)
945 CALL TLIMIT(ENAME,NCHAIN,Q,ICOND)
946 IF(ICOND)641,641,9999
947 641 CONTINUE
948 CALL TIME23(NCHAIN,Q,J3)
949 200 CONTINUE
950 INS=.TRUE.

```

```

895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950

```

```

951 LG(1)=.TRUE.
952 LG(2)=.TRUE.
953 LDISP=.TRUE.
954 LNDISP=.TRUE.
955 XX5=DUR*(V0/PATH*(EQ(1)*DNR(1)-EQ(2)*DNR(2))/(EQ(1)-EQ(2)))
956 XX6=0.
957 XX7=0.
958 XX19=DUR*(DNR(1)*V0/PATH)
959 XX29=DUR*(DNR(2)*V0/PATH-ALAMB(1))
960 XX39=0.
961 G1=4*EQ(2)*(DNR(2)-DNR(1))/PE
962 G2=4.D0*EQ(1)*EQ(2)*(DNR(2)-DNR(1))/(PE*(EQ(1)-EQ(2)))
963 IF(G1.LT.-1.)LG(1)=.FALSE.
964 IF(G2.LT.-1.)LG(2)=.FALSE.
965 IF(.NOT.LG(1).OR..NOT.LG(2))INS=.FALSE.
966 CALL FNCD2
967 IF(LDISP)CALL FWD2
968 DO 1060 J=1,IS
969 TD=THETA(J)
970 DSM=AK*ETA+DUR*V0/PATH
971 IF(1.-DSM/TD.GT.1.E-7)LNDISP=.FALSE.
972 IF(.NOT.LNDISP)SNOD2=0.0
973 IF(LDISP)CALL BWD2(SWD1,SWD2,SWD3,2,J)
974 IF(LNDISP)CALL BNOD2(SNOD1,SNOD2,SNOD3,2,J)
975 IF(.NOT.LDISP)SWD2=SNOD2
976 SWD2=SWD2*ALAMB(2)
977 SNOD2=SNOD2*ALAMB(2)
978 IF(IPSEUD-2)1040,1050,1050
979 1040 CONTINUE
980 BAND(J)=SWD2
981 BANDND(J)=SNOD2
982 GO TO 1055
983 1050 CONTINUE
984 BAND(J)=BAND(J)-SWD2
985 BANDND(J)=BANDND(J)-SNOD2
986 1055 CONTINUE
987 IF(TD.LT.AL.OR.TD.GT.AK+DUR*V0/PATH)BANDND(J)=0.0
988 1060 CONTINUE
989 IF(IPSEUD-1)1100,1070,1090
990 1070 CONTINUE
991 IJ=INTNUK(PS3)
992 EQ(3)=EQ(1)
993 THALF(3)=THALF(1)
994 DNR(3)=DNR(1)
995 SWAP1=GAP(1)
996 SWAP2=GAP(4)
997 EQ(1)=RET(IJ)
998 THALF(1)=THALV(IJ)
999 ALAMB(1)=0.69314718/THALF(1)
1000 IPSEUD=2
1001 CALL VECTOR(NCHAIN,IPSEUD)
1002 GO TO 200
1003 1050 CONTINUE
1004 EQ(1)=EQ(3)
1005 THALF(1)=THALF(3)
1006 DNR(1)=DNR(3)

```

```

951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006

```

```

1007      GAP(1)=SWAP1
1008      GAP(4)=SWAP2
1009 1100 CONTINUE
1010      CALL TOPDEF
1011      IF(MIGOUT) CALL PRINTX(NCHAIN)
1012      WRITE(14)(ENAME(I),I=1,3),IS
1013      IF(IS.EQ.0)GO TO 9999
1014      WRITE(14)(TIME(I),BAND(I),I=1,IS)
1015      IJ=INTNUK(ENAME(2))
1016      ICONTR(IJ)=ICONTR(IJ)+1
1017      NN=ICONTR(IJ)
1018      TSTART(IJ,NN)=TBEGIN
1019      TEND(IJ,NN)=TFIN
1020      TPEAK(IJ,NN)=TTOPP
1021      CMAX(IJ,NN)=BMAX
1022 9999 CONTINUE
1023      RETURN
1024      END
1025
1026      SUBROUTINE TRIPLE(INS,J3)
1027
1028  C
1029  C THIS ROUTINE CALCULATES THE RELEASE RATES FOR THREE-NUCLIDE CHAINS
1030  C
1031      COMMON/PRINT/ MIGOUT
1032      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
1033      *             GAP(6),TO,DUR,PE,ETA,TD,VO,PATH
1034      COMMON/C2/ IS,BAND(304),BANDND(304),TIME(304),THETA(304),
1035      *           TBEGIN,TFIN,TTOPP,BMAX,DINO
1036      COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
1037      *             CMAX(75,10),DOSE2(75)
1038      COMMON/HALVA/ THALV(75),RET(75)
1039      COMMON /XXX/ XX5,XX6,XX7,XX19,XX29,XX39
1040      COMMON /DIV/ DSW,DSN
1041      COMMON /LGTEST/ LG(6)
1042      REAL*16 THALV
1043      REAL*8 SWD1,SWD2,SWD3,SNOD1,SNOD2,SNOD3
1044      REAL*8 XX5,XX6,XX7,XX19,XX29,XX39
1045      REAL*8 DSW,DSN
1046      REAL*8 ENAME,PS1,PS2,PS3,PS4,SWAP1,Y
1047      REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,TO,DUR,PE,ETA,TD,VO,PATH
1048      LOGICAL MIGOUT,INS,INSCCN
1049      LOGICAL LDISP,LNDISP
1050      LOGICAL LG
1051      DATA PS1/'U 238'/,PS2/'TH230'/,PS3/'RA226'/,PS4/'U 234'/'
1052      DIMENSION Q(15)
1053      NCHAIN=3
1054      BMAX=0.0
1055      DINO=0.0
1056      IPSEUD=0
1057      AK=DMAX1(EQ(1),EQ(2),EQ(3))
1058      AL=DMIN1(EQ(1),EQ(2),EQ(3))
1059      IF(ENAME(1).NE.PS1.OR.ENAME(2).NE.PS2.OR.ENAME(3).NE.PS3)GO TO 100
1060 100 CONTINUE
1061      CALL VECTOR(NCHAIN,IPSEUD)
1062      CALL TLIMIT(ENAME,NCHAIN,Q,ICONTR)
1063      IF(ICONTR)641,641,9999

```

1063	641	CONTINUE	1063
1064		CALL TIME23(NCHAIN,Q,J3)	1064
1065	200	CONTINUE	1065
1066		INS=.TRUE.	1066
1067		LDISP=.TRUE.	1067
1068		LNDISP=.TRUE.	1068
1069		XX5=DUR*(V0/PATH*(EQ(1)*DNR(1)-EQ(2)*DNR(2))/(EQ(1)-EQ(2)))	1069
1070		XX6=DUR*(V0/PATH*(EQ(2)*DNR(2)-EQ(3)*DNR(3))/(EQ(2)-EQ(3)))	1070
1071		XX7=DUR*(V0/PATH*(EQ(1)*DNR(1)-EQ(3)*DNR(3))/(EQ(1)-EQ(3)))	1071
1072		XX19=DUR*(DNR(1)*V0/PATH)	1072
1073		XX29=DUR*(DNR(2)*V0/PATH)	1073
1074		XX39=DUR*(DNR(3)*V0/PATH)	1074
1075		DO 250 I=1,6	1075
1076		LG(I)=.TRUE.	1076
1077	250	CONTINUE	1077
1078		CALL G1(1,2,3,LG(1),&300)	1078
1079		CALL G3(1,2,3,LG(1),&300)	1079
1080		CALL G2(1,2,LG(1))	1080
1081	300	CALL G2(1,3,LG(2))	1081
1082		CALL G2(2,3,LG(3))	1082
1083		CALL G5(1,2,LG(4))	1083
1084		CALL G5(1,3,LG(5))	1084
1085		CALL G5(2,3,LG(6))	1085
1086		DO 350 I=1,6	1086
1087		IF(LG(I))GO TO 350	1087
1088		INS=.FALSE.	1088
1089	350	CONTINUE	1089
1090		CALL FNOD3	1090
1091		DO 1060 J=1,IS	1091
1092		TD=THETA(J)	1092
1093		DSM=AK*ETA+DUR*V0/PATH	1093
1094		IF(1.-DSM/TD.GT.1.E-7)LNDISP=.FALSE.	1094
1095		IF(.NOT.LNDISP)SNOD3=0.0	1095
1096		IF(LDISP)CALL BWD3(SWD1,SWD2,SWD3,3,J)	1096
1097		IF(LNDISP)CALL BNOD3(SNOD1,SNOD2,SNOD3,3,J)	1097
1098		IF(.NOT.LDISP)SWD3=SNOD3	1098
1099		BRIID=SWD3*ALAMB(3)	1099
1100		BRND=SNOD3*ALAMB(3)	1100
1101		IF(IPSEUD-2)1040,1050,1050	1101
1102	1040	CONTINUE	1102
1103		BAND(J)=BRIID	1103
1104		BANDND(J)=BRND	1104
1105		GO TO 1060	1105
1106	1050	CONTINUE	1106
1107		BAND(J)=BAND(J)-BRIID	1107
1108		BANDND(J)=BANDND(J)-BRND	1108
1109	1060	CONTINUE	1109
1110		IF(IPSEUD-1)1100,1070,1090	1110
1111	1070	CONTINUE	1111
1112		IJ=INTNUK(PS4)	1112
1113		SWAP2=DNR(1)	1113
1114		SWAP5=GAP(1)	1114
1115		SWAP6=GAP(4)	1115
1116		SWAP3=THALF(1)	1116
1117		SWAP4=EQ(1)	1117
1118		THALF(1)=THALV(IJ)	1118

1119		ALAMB(1)=0.69314718/THALF(1)	1119
1120		EQ(1)=RET(IJ)	1120
1121		IPSEUD=2	1121
1122		CALL VECTOR(NCHAIN,IPSEUD)	1122
1123		GO TO 200	1123
1124	1090	CONTINUE	1124
1125		DNR(1)=SWAP2	1125
1126		GAP(1)=SWAP5	1126
1127		GAP(4)=SWAP6	1127
1128		THALF(1)=SWAP3	1128
1129		EQ(1)=SWAP4	1129
1130	1100	CONTINUE	1130
1131		CALL TOPDEF	1131
1132		IF(MIGOUT) CALL PRINTX(NCHAIN)	1132
1133		WRITE(15)(ENAME(I),I=1,3),IS	1133
1134		IF(IS.EQ.0)GO TO 9999	1134
1135		WRITE(15)(TIME(I),BAND(I),I=1,IS)	1135
1136		IJ=INTNUK(ENAME(3))	1136
1137		ICONTR(IJ)=ICONTR(IJ)+1	1137
1138		NN=ICONTR(IJ)	1138
1139		TSTART(IJ,NN)=TBEGIN	1139
1140		TEND(IJ,NN)=TFIN	1140
1141		TPEAK(IJ,NN)=TTOPP	1141
1142		CMAX(IJ,NN)=BMAX	1142
1143	9999	RETURN	1143
1144		END	1144
1145		SUBROUTINE TSING(PE,TH1)	1145
1146	C		1146
1147	C	THIS ROUTINE CALCULATES THE START POINT FOR SINGLE-NUCLIDE	1147
1148	C	RELEASE WITH DISPERSION	1148
1149	C		1149
1150		REAL*8 PE	1150
1151		DATA PI/3.14159/,EP/1.E-12/,EPS/1.E-15/	1151
1152		FUNK(TH,PE)=0.5*TDERFC(0.5*(DSQRT(PE/TH)-DSQRT(PE*TH))+TEXP(PE)*	1152
1153		*TDERFC(0.5*DSQRT(PE/TH)+DSQRT(PE*TH)))	1153
1154		DER(TH,PE)=0.5*DSQRT(PE/PI/TH**3)*TEXP(-PE/4./TH*(1-TH)**2)	1154
1155		AK=1	1155
1156		BK=1.E-5	1156
1157	10	CONTINUE	1157
1158		AM=0.5*(AK+BK)	1158
1159		F=FUNK(AM,PE)	1159
1160		IF(F.LT.EPS)GO TO 20	1160
1161		IF(F.GT.EP)GO TO 30	1161
1162		THE=AM	1162
1163		GO TO 40	1163
1164	20	CONTINUE	1164
1165		BK=AM	1165
1166		GO TO 10	1166
1167	30	CONTINUE	1167
1168		AK=AM	1168
1169		GO TO 10	1169
1170	40	CONTINUE	1170
1171		A=FUNK(TH,PE)	1171
1172		IF(A.LT.EPS)GO TO 50	1172
1173		B=DER(TH,PE)	1173
1174		THE=THE-A/B	1174

1175		GO TO 40	1175
1176	50	CONTINUE	1176
1177		TH1=THE	1177
1178		RETURN	1178
1179		END	1179
<hr/>			
1180		SUBROUTINE TIME23(NCHAIN,Q,J3)	1180
1181	C		1181
1182	C	THIS ROUTINE CALCULATES A TIME VECTOR FOR DOUBLE AND TRIPLE.	1182
1183	C	WITHIN THE LIMITS GIVEN BY TLIMIT	1183
1184	C		1184
1185		COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3).	1185
1186	*	GAP(6),T0,T1,PE,ETA,TD,V,AL	1186
1187	COMMON/C2/	IB,BAND(304),BANDND(304),T(304),TDL(304).	1187
1188	*	TBEGIN,TFIN,TTOPP,BRPEAK,DINO	1188
1189		REAL*8 ENAME	1189
1190		REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,T0,T1,PE,ETA,TD,V,AL	1190
1191		DIMENSION Q(15)	1191
1192		II=5*NCHAIN	1192
1193		N=II-1	1193
1194		DO 400 JC=1,N	1194
1195		NQ=JC	1195
1196		DO 300 J=JC,II	1196
1197		IF(Q(J).LT.Q(NQ))NQ=J	1197
1198	300	CONTINUE	1198
1199		SWAP=Q(NQ)	1199
1200		Q(NQ)=Q(JC)	1200
1201		Q(JC)=SWAP	1201
1202	400	CONTINUE	1202
1203		J1=0	1203
1204		IF(Q(1).GT.0.0)J1=1	1204
1205		DO 500 N=2,II	1205
1206		IF(Q(N).EQ.0.0.OR.Q(N).EQ.Q(N-1))GO TO 500	1206
1207		J1=J1+1	1207
1208		Q(J1)=Q(N)	1208
1209	500	CONTINUE	1209
1210		IB=1	1210
1211		DO 700 JC=2,J1	1211
1212		STEP=(Q(JC)-Q(JC-1))/J3	1212
1213		T(IB)=Q(JC-1)	1213
1214		TDL(IB)=(T(IB)-T0)*V/AL	1214
1215		IB=IB+1	1215
1216		DO 600 J=2,J3	1216
1217		T(IB)=T(IB-1)+STEP	1217
1218		TDL(IB)=(T(IB)-T0)*V/AL	1218
1219		IB=IB+1	1219
1220	600	CONTINUE	1220
1221	700	CONTINUE	1221
1222		IB=IB-1	1222
1223		RETURN	1223
1224		END	1224
<hr/>			
1225		SUBROUTINE TLIMIT(ENAME,NCHAIN,Q,ICOND)	1225
1226	C		1226
1227	C	THIS ROUTINE DEFINES THE LIMITS BETWEEN WHICH THE TIMESCALE FOR	1227
1228	C	THE CALCULATION OF DOUBLE AND TRIPLE CHAINS ARE TO BE LAID OUT.	1228
1229	C		1229
1230		DIMENSION Q(15),IJ(3)	1230

1231		REAL*8 ENAME(3)	1231
1232		COMMON/PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),	1232
1233		* CMAX(75,10),DOSE2(75)	1233
1234		COMMON /TMAXND/TND1(75),TND2(75)	1234
1235	C		1235
1236		ICOND=0	1236
1237	C		1237
1238		DO 10 I=1,NCHAIN	1238
1239		IJ(I)=INTNUK(ENAME(I))	1239
1240		IF(ICONTR(IJ(I)).GT.0)GO TO 10	1240
1241		WRITE(6,7)(ENAME(J),J=1,2)	1241
1242		IF(NCHAIN.EQ.3)WRITE(6,8)ENAME(3)	1242
1243		WRITE(6,9)ENAME(I)	1243
1244		ICOND=1	1244
1245		10 CONTINUE	1245
1246	C		1246
1247		7 FORMAT('0 WHILE EXECUTING TLIMIT FOR ',2A8)	1247
1248		8 FORMAT('+',43X,A8)	1248
1249		9 FORMAT(' NO PEAK WAS FOUND FOR ',A6)	1249
1250	C		1250
1251		IF(ICOND.EQ.1)RETURN	1251
1252	C		1252
1253		DO 20 I=1,NCHAIN	1253
1254		Q(5*I-4)=TSTART(IJ(I),1)	1254
1255		Q(5*I-3)=TND1(IJ(I))	1255
1256		Q(5*I-2)=TPEAK(IJ(I),1)	1256
1257		Q(5*I-1)=TND2(IJ(I))	1257
1258		Q(5*I)=TEND(IJ(I),1)	1258
1259		20 CONTINUE	1259
1260	C		1260
1261		RETURN	1261
1262		END	1262
1263		SUBROUTINE VECTOR(NCHAIN,IPSEUD)	1263
1264	C		1264
1265	C	THIS ROUTINE IS USED BY DOUBLE AND TRIPLE FOR THE	1265
1266	C	CALCULATION OF THE DECAY CONSTANTS	1266
1267	C		1267
1268		COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DECAY(3),	1268
1269		* GAP(6),T0,T1,PE,ETA,TD,V,AL	1269
1270		REAL*8 ENAME	1270
1271		REAL*8 THALF,AQ,ALAMB,EQ,DECAY,GAP,T0,T1,PE,ETA,TD,V,AL	1271
1272		SLASK=0.69314718	1272
1273		SLASK1=SLASK*AL/V	1273
1274		DO 100 I=1,NCHAIN	1274
1275		DECAY(I)=SLASK1/THALF(I)	1275
1276	100	CONTINUE	1276
1277		X1=TEXP(-ALAMB(1)*T1)	1277
1278		X2=TEXP(-ALAMB(2)*T1)	1278
1279		IF(NCHAIN.EQ.3)X3=TEXP(-ALAMB(3)*T1)	1279
1280		GAP(1)=AQ(1)/ALAMB(1)	1280
1281		IF(IPSEUD.GE.1)GAP(1)=GAP(1)/2.44E5/(1.0/2.44E5-1.0/4.47E9)	1281
1282		GAP(2)=AQ(2)/ALAMB(2)	1282
1283		GAP(3)=0.0	1283
1284		IF(NCHAIN.EQ.3)GAP(3)=AQ(3)/ALAMB(3)	1284
1285		GAP(4)=GAP(1)*X1	1285
1286		GAP(5)=GAP(2)*X2+GAP(1)*ALAMB(1)/(ALAMB(2)-ALAMB(1))	1286

```

1287      *)*(X1-X2)
1288      IF(NCHAIN.EQ.3)GO TO 200
1289      GAP(6)=0.0
1290      GO TO 300
1291 200  GAP(6)=GAP(3)*X3+GAP(2)*ALAMB(2)/(ALAMB(3)-ALAMB(2)
1292      *)*(X2-X3)+GAP(1)*ALAMB(1)*ALAMB(2)*
1293      *X1/(ALAMB(2)-ALAMB(1))/(ALAMB(3)-ALAMB(1))
1294      GAP(6)=GAP(6)+GAP(1)*ALAMB(1)*ALAMB(2)*X2/(ALAMB(1)
1295      *-ALAMB(2))/(ALAMB(3)-ALAMB(2))
1296      GAP(6)=GAP(6)+GAP(1)*ALAMB(1)*ALAMB(2)*X3/(ALAMB(2)
1297      *-ALAMB(3))/(ALAMB(1)-ALAMB(3))
1298 300  CONTINUE
1299      RETURN
1300      END

```

```

1301      SUBROUTINE GTEST(ARGDUM)
1302  C
1303  C      THIS ROUTINE TEST FOR NEGATIVE SQUARE ROOT ARGUMENTS
1304  C
1305      IMPLICIT REAL*8 (A-H,O-Z)
1306      COMMON /FFFFF/ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),
1307      *      GAP(6),TO,DUR,PE,ETA,TD,VO,PATH
1308      LOGICAL LG1,LG2,LG3,LG4,LG5
1309      LG1=.TRUE.
1310      LG2=.TRUE.
1311      LG3=.TRUE.
1312      LG4=.TRUE.
1313      LG5=.TRUE.
1314      ENTRY G1(I,J,K,LG1,*)
1315      G1A=1.00+4.00*EQ(K)*DNR(K)/PE-4.*EQ(K)*(EQ(I)*DNR(I)-EQ(J)*
1316      *DNR(J))/(PE*(EQ(I)-EQ(J)))
1317      IF(G1A.GE.0.000)RETURN
1318      LG1=.FALSE.
1319      RETURN1
1320      ENTRY G2(I,J,LG2)
1321      G2A=1.00+4.00*EQ(I)*EQ(J)*(DNR(J)-DNR(I))/(PE*(EQ(I)-EQ(J)))
1322      IF(G2A.LT.0.000)LG2=.FALSE.
1323      RETURN
1324      ENTRY G3(I,J,K,LG3,*)
1325      G3A=PE/(4.00*EQ(K))+DNR(K)-(EQ(I)*DNR(I)-EQ(J)*DNR(J))/
1326      *(EQ(I)-EQ(J))
1327      IF(G3A.GE.0.000)RETURN
1328      LG3=.FALSE.
1329      RETURN1
1330      ENTRY G4(I,J,K,L,LG4)
1331      G4A=PE/(4.00*EQ(K))+EQ(L)*(DNR(J)-DNR(I))/(EQ(I)-EQ(J))
1332      IF(G4A.LT.0.000)LG4=.FALSE.
1333      RETURN
1334      ENTRY G5(I,J,LG5)
1335      G5A=1.00+4.00*EQ(J)*(DNR(J)-DNR(I))/PE
1336      IF(G5A.LT.0.000)LG5=.FALSE.
1337      RETURN
1338      END

```

```

1339      FUNCTION F(DUMDUM)
1340      IMPLICIT REAL*8 (A-H,O-Z)
1341      REAL*8 NO
1342      COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),AC(3),R(3),NO(6),

```

1343		*T0,T,P,ETA,THETA,V,SCL	1343
1344		F=0	1344
1345		ENTRY F4(I,J,K,L)	1345
1346		F4=(AC(K)*R(K)-AC(L)*R(L))/(AC(K)-AC(L))-(AC(I)*R(I)-AC(J)*R(J)) /	1346
1347		*(AC(I)-AC(J))	1347
1348		RETURN	1348
1349	C		1349
1350		ENTRY F11(I,J)	1350
1351		F11=N0(I)/(2.D0*T)*R(I)/(R(J)-R(I))	1351
1352		RETURN	1352
1353	C		1353
1354		ENTRY F12(I,J)	1354
1355		F12=N0(I)/(2.D0*T)*R(I)*R(J)	1355
1356		RETURN	1356
1357	C		1357
1358		ENTRY F13(I,J,K)	1358
1359		F13=N0(I)/(2.D0*T)*AC(J)*R(I)*R(J)/(AC(J)-AC(K))	1359
1360		RETURN	1360
1361	C		1361
1362		ENTRY F14(I,J,K,L)	1362
1363		F14=N0(I)/(2.D0*T)*AC(I)*AC(J)*R(I)*R(J)/((AC(I)-AC(J))*(AC(L)-	1363
1364		*AC(K)))	1364
1365		RETURN	1365
1366	C		1366
1367		ENTRY F15(I,J)	1367
1368		F15=1.D0-AC(I)/AC(J)	1368
1369		RETURN	1369
1370	C		1370
1371		ENTRY F16(I,J)	1371
1372		F16=AC(I)/AC(J)	1372
1373		RETURN	1373
1374	C		1374
1375		ENTRY F17(I,J)	1375
1376		F17=R(J)-R(I)	1376
1377		RETURN	1377
1378	C		1378
1379		ENTRY F18(I,J,K)	1379
1380		F18=(AC(J)*R(J)-AC(K)*R(K))/(AC(J)-AC(K))-R(I)	1380
1381		RETURN	1381
1382	C		1382
1383		ENTRY F34(I)	1383
1384		F34=0.0D0	1384
1385		IF(AC(I)*ETA/THETA-1.0GT.1.D-7)RETURN	1385
1386		F34=1.0D0	1386
1387		RETURN	1387
1388		END	1388
1389		SUBROUTINE FF(DUM)	1389
1390		IMPLICIT REAL*8 (A-H,O-Z)	1390
1391		REAL*8 NO	1391
1392		COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),AC(3),R(3),NO(6),	1392
1393		*T0,T,P,ETA,THETA,V,SCL	1393
1394		COMMON/NEGXER/ ERFCM1,EXPCM1	1394
1395	C		1395
1396		ENTRY FF5(I,J,F5E)	1396
1397		F5E=-THETA*(AC(I)*R(I)-AC(J)*R(J))/(AC(I)-AC(J))	1397
1398		RETURN	1398

1399	C		1399
1400		ENTRY FF6P(I,J,K,F6PE)	1400
1401		N=1	1401
1402		GO TO 50	1402
1403	20	F6PE=P*ETA/2.D0*(1.D0+F6PMT)	1403
1404		RETURN	1404
1405	C		1405
1406		ENTRY FF6M(I,J,K,F6ME)	1406
1407		N=2	1407
1408		GO TO 50	1408
1409	30	CONTINUE	1409
1410		IF(G.LE.3.D-9)GO TO 35	1410
1411		TERM=1.-F6PMT	1411
1412		GO TO 40	1412
1413	35	G=-G/4.D0	1413
1414		TERM=2.D0*G*(1.D0+G*(1.D0+G*(2.D0+G*(5.D0+G*	1414
1415		*(14.D0+G*(42.D0+G*132.D0))))))	1415
1416	40	F6ME=P*ETA/2.D0*TERM	1416
1417		RETURN	1417
1418	50	CONTINUE	1418
1419		G=4.D0*AC(K)*R(K)/P-4.*AC(K)*(AC(I)*R(I)-AC(J)*R(J))/(P*(AC(I)-	1419
1420		*AC(J)))	1420
1421		IF(G.LT.-1.D0)G=-1.D0	1421
1422		F6PMT=DSQRT(1.D0+G)	1422
1423		IF(N.EQ.1)GO TO 20	1423
1424		IF(N.EQ.2)GO TO 30	1424
1425	C		1425
1426		ENTRY FF7P(I,J,F7PE)	1426
1427		N=1	1427
1428		GO TO 100	1428
1429	60	CONTINUE	1429
1430		F7PE=P*ETA/2.D0*(1.D0+F7PMT)	1430
1431		RETURN	1431
1432	C		1432
1433		ENTRY FF7M(I,J,F7ME)	1433
1434		N=2	1434
1435		GO TO 100	1435
1436	70	CONTINUE	1436
1437		IF(G.LE.3.D-9)GO TO 80	1437
1438		TERM=1.D0-F7PMT	1438
1439		GO TO 90	1439
1440	80	G=-G/4.D0	1440
1441		TERM=2.D0*G*(1.D0+G*(1.D0+G*(2.D0+G*(5.D0+G*	1441
1442		*(14.D0+G*(42.D0+G*132.D0))))))	1442
1443	90	F7ME=P*ETA/2.D0*TERM	1443
1444		RETURN	1444
1445	100	CONTINUE	1445
1446		G=4.D0*AC(I)*AC(J)*(R(J)-R(I))/(P*(AC(I)-AC(J)))	1446
1447		IF(G.LT.-1.D0)G=-1.D0	1447
1448		F7PMT=DSQRT(1.D0+G)	1448
1449		IF(N.EQ.1)GO TO 60	1449
1450		IF(N.EQ.2)GO TO 70	1450
1451	C		1451
1452		ENTRY FF8P(I,J,K,F8PE,F8PA)	1452
1453		N=1	1453
1454		GO TO 150	1454

1455		110 CONTINUE	1455
1456		ARG=ETA/2.DO*DSQRT(AC(K)*P/THETA)+F8PMT	1456
1457		CALL FERRNT(ARG,ERF,ERFCE,ERFCA)	1457
1458		F8PE=ERFCE	1458
1459		F8PA=ERFCA	1459
1460		RETURN	1460
1461	C		1461
1462		ENTRY FF8M(I,J,K,F8ME,F8MA,F8MARG,F8MME,F8MMA)	1462
1463		N=2	1463
1464		GO TO 150	1464
1465		120 CONTINUE	1465
1466		ARG=ETA/2.DO*DSQRT(AC(K)*P/THETA)-F8PMT	1466
1467		CALL FERRNT(ARG,ERF,ERFCE,ERFCA)	1467
1468		F8ME=ERFCE	1468
1469		F8MA=ERFCA	1469
1470		F8MARG=ARG	1470
1471		F8MME=EXPCM1	1471
1472		F8MMA=ERFCM1	1472
1473		RETURN	1473
1474	C		1474
1475		150 CONTINUE	1475
1476		G=((P/(4.DO*AC(K)))+R(K)-(AC(I)*R(I)-AC(J)*R(J))/ *(AC(I)-AC(J))*THETA)	1476
1477		IF(G.LT.0.0D0)G=0.0D0	1477
1478		F8PMT=DSQRT(G)	1478
1479		IF(N.EQ.1)GO TO 110	1479
1480		IF(N.EQ.2)GO TO 120	1480
1481			1481
1482	C		1482
1483		ENTRY FF9M(I,J,K,L,F9ME,F9MA,F9MARG,F9MME,F9MMA)	1483
1484		N=1	1484
1485		GO TO 200	1485
1486		160 CONTINUE	1486
1487		ARG=ETA/2.DO*DSQRT(AC(K)*P/THETA)-F9PMT	1487
1488		CALL FERRNT(ARG,ERF,ERFCE,ERFCA)	1488
1489		F9ME=ERFCE	1489
1490		F9MA=ERFCA	1490
1491		F9MARG=ARG	1491
1492		F9MME=EXPCM1	1492
1493		F9MMA=ERFCM1	1493
1494		RETURN	1494
1495	C		1495
1496		ENTRY FF9P(I,J,K,L,F9PE,F9PA)	1496
1497		N=2	1497
1498		GO TO 200	1498
1499		170 CONTINUE	1499
1500		ARG=ETA/2.DO*DSQRT(AC(K)*P/THETA)+F9PMT	1500
1501		CALL FERRNT(ARG,ERF,ERFCE,ERFCA)	1501
1502		F9PE=ERFCE	1502
1503		F9PA=ERFCA	1503
1504		RETURN	1504
1505		200 CONTINUE	1505
1506		G=((P/(4.DO*AC(K)))+AC(L)*(R(J)-R(I))/(AC(I)-AC(J)) **THETA)	1506
1507		IF(G.LT.0.0D0)G=0.0D0	1507
1508		F9PMT=DSQRT(G)	1508
1509		IF(N.EQ.1)GO TO 160	1509
1510			1510

```

1511 IF(N.EQ.2)GO TO 170
1512 C
1513 ENTRY FF19(I,F19E)
1514 F19E=-R(I)*THETA
1515 RETURN
1516 C
1517 ENTRY FF20P(I,J,F20PE)
1518 N=1
1519 GO TO 250
1520 210 CONTINUE
1521 F20PE=P*ETA/2.D0*(1.D0+F20PMT)
1522 RETURN
1523 C
1524 ENTRY FF20M(I,J,F20ME)
1525 N=2
1526 GO TO 250
1527 220 CONTINUE
1528 IF(G.LE.3.D-9)GO TO 230
1529 TERM=1.D0-F20PMT
1530 GO TO 240
1531 230 CONTINUE
1532 G=-G/4.D0
1533 TERM=2.D0*G*(1.D0+G*(1.D0+G*(2.D0+G*(5.D0+G*
1534 *(14.D0+G*(42.D0+G*132.D0))))))
1535 240 CONTINUE
1536 F20ME=P*ETA/2.*TERM
1537 RETURN
1538 250 CDNTINUE
1539 G=4.D0*AC(J)*(R(J)-R(I))/P
1540 IF(G.LT.-1.D0)G=-1.D0
1541 SARG=1.D0+G
1542 F20PMT=DSQRT(DABS(SARG))
1543 IF(N.EQ.1)GO TO 210
1544 IF(N.EQ.2)GO TO 220
1545 C
1546 ENTRY FF21P(I,F21E,F21A)
1547 ARG=ETA/2.D0*DSQRT(AC(I)*P/THETA)+DSQRT(P*THETA/(4.D0*AC(I)))
1548 CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
1549 F21E=ERFCE
1550 F21A=ERFCA
1551 RETURN
1552 C
1553 ENTRY FF21(I,F21E,F21A,F21ARG,F21ME,F21MA)
1554 ARG=ETA/2.D0*DSQRT(AC(I)*P/THETA)-DSQRT(P*THETA/(4.D0*AC(I)))
1555 CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
1556 F21E=ERFCE
1557 F21A=ERFCA
1558 F21ARG=ARG
1559 F21ME=EXPCM1
1560 F21MA=ERFCM1
1561 RETURN
1562 C
1563 ENTRY FF22(I,F22E,F22A)
1564 ARG=ETA/2.D0*DSQRT(AC(I)*P/THETA)+DSQRT(P*THETA/(4.D0*AC(I)))
1565 CALL FERRNT(ARG,ERF,ERFCE,ERFCA)
1566 F22E=ERFCE+P*ETA

```

```

1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566

```


1567		F22A=ERFCA	1567
1568		RETURN	1568
1569	C		1569
1570		ENTRY FF23P(I,J,F23PE,F23PA)	1570
1571		G=((P/4.D0*AC(J))+R(J)-R(I))*THETA)	1571
1572		IF(G.LT.0.0D0)G=0.0D0	1572
1573		PMT=DSQRT(G)	1573
1574		ARG=ETA/2.D0*DSQRT(AC(J)*P/THETA)+PMT	1574
1575		CALL FERRNT(ARG,ERF,ERFCE,ERFCA)	1575
1576		F23PE=ERFCE	1576
1577		F23PA=ERFCA	1577
1578		RETURN	1578
1579	C		1579
1580		ENTRY FF23M(I,J,F23ME,F23MA,F23MAR,F23MME,F23MMA)	1580
1581		G=((P/(4.D0*AC(J))+R(J)-R(I))*THETA)	1581
1582		IF(G.LT.0.0D0)G=0.0D0	1582
1583		PMT=DSQRT(G)	1583
1584		ARG=ETA/2.D0*DSQRT(AC(J)*P/THETA)-PMT	1584
1585		CALL FERRNT(ARG,ERF,ERFCE,ERFCA)	1585
1586		F23ME=ERFCE	1586
1587		F23MA=ERFCA	1587
1588		F23MAR=ARG	1588
1589		F23MME=EXPCM1	1589
1590		F23MMA=ERFCM1	1590
1591		RETURN	1591
1592	C		1592
1593		ENTRY FF30(I,J,F30E)	1593
1594		F30E=-AC(I)*AC(J)*(R(J)-R(I))*ETA/(AC(I)-AC(J))	1594
1595		RETURN	1595
1596	C		1596
1597		ENTRY FF31(I,J,K,F31)	1597
1598		F31=-ETA*(AC(K)*R(K)-AC(K)*(AC(I)*R(I)-AC(J)-R(J)))/(AC(I)-AC(J))	1598
1599		RETURN	1599
1600	C		1600
1601		ENTRY FF33(I,J,F33E)	1601
1602		F33E=-AC(J)*(R(J)-R(I))*ETA	1602
1603		RETURN	1603
1604		END	1604
1605		SUBROUTINE SUM2(A,B,C,D,E,F)	1605
1606	C		1606
1607	C	THIS ROUTINE COMPUTES THE SUM OF TWO EXPCNETIAL FUNCTIONS :	1607
1608	C	EXP(E)*F=EXP(A)*B+EXP(C)*D	1608
1609	C		1609
1610		IMPLICIT REAL*16 (G,T-Z)	1610
1611		REAL*8 A,B,C,D,E,F	1611
1612		IF(B.EQ.0.D0)GO TO 30	1612
1613		IF(D.EQ.0.D0)GO TO 50	1613
1614	C	----B.NE.0..AND.D.NE.0.	1614
1615		ZB=B	1615
1616		ZJB=QABS(ZB)	1616
1617		ZLB=QLOG(ZBB)	1617
1618		ZILB=QINT(ZLB)	1618
1619		ZL=ZLB-ZILB	1619
1620		TL=QSIGN(QEXP(ZL),ZB)	1620
1621		XL=A+ZILB	1621
1622		ZD=D	1622

1623		ZDD=QABS(ZD)	1623
1624		ZLD=QLOG(ZDD)	1624
1625		ZILD=QINT(ZLD)	1625
1626		ZR=ZLD-ZILD	1626
1627		TR=QSIGN(QEXP(ZR),ZD)	1627
1628		XR=C+ZILD	1628
1629		ZE=XR	1629
1630		IF(XL.GT.XR)ZE=XL	1630
1631		G=XR-ZE	1631
1632		IF(G.GT.-174.D0) GO TO 10	1632
1633		E=XL	1633
1634		F=TL	1634
1635		RETURN	1635
1636	10	CONTINUE	1636
1637		IF(G.NE.0.D0)TR=QEXP(G)*TR	1637
1638		G=XL-ZE	1638
1639		IF(G.GT.-174.D0)GO TO 20	1639
1640		E=XR	1640
1641		F=TR	1641
1642		RETURN	1642
1643	20	CONTINUE	1643
1644		IF(G.NE.0.D0)TL=QEXP(G)*TL	1644
1645		E=ZE	1645
1646		F=TL+TR	1646
1647		RETURN	1647
1648	30	IF(D.EQ.0.D0)GO TO 40	1648
1649	C	-----B.EQ.0.AND D.NE.0.	1649
1650		E=C	1650
1651		F=D	1651
1652		RETURN	1652
1653	C	-----B.EQ.0. AND D.EQ.0.	1653
1654	40	E=0.D0	1654
1655		F=0.D0	1655
1656		RETURN	1656
1657	C	-----B.NE.0. AND D.EQ.0.	1657
1658	50	E=A	1658
1659		F=B	1659
1660		RETURN	1660
1661		END	1661
<hr/>			
1662		SUBROUTINE FERRNT(X,ERF1,EXPC1,ERFC1)	1662
1663		IMPLICIT REAL*8 (A-H,O-Z)	1663
1664		COMMON /NEGXER/ ERFCM1,EXPCM1	1664
1665		IF(X.EQ.0.0D0)GO TO 10	1665
1666		IF(X.GT.13.0)GO TO 20	1666
1667		Y=DABS(X)	1667
1668		ERFCM1=DERFC(Y)	1668
1669		EXPCM1=0.0D0	1669
1670		ERFC1=ERFCM1	1670
1671		EXPC1=EXPCM1	1671
1672		IF(X.GT.0.0D0)RETURN	1672
1673		ERF1=-ERF1	1673
1674		ERFC1=2.0D0-ERFCM1	1674
1675		RETURN	1675
1676	10	CONTINUE	1676
1677		ERF1=0.0D0	1677
1678		ERFC1=1.0D0	1678

```

1679      EXPC1=0.0D0
1680      ERFCM1=ERFC1
1681      EXPCM1=EXPC1
1682      RETURN
1683      20 CONTINUE
1684      ERF1=0.0D0
1685      ERFC1=0.0D0
1686      EXPC1=0.0D0
1687      ERFCM1=ERFC1
1688      EXPCM1=EXPC1
1689      RETURN
1690      END

```

```

1691      SUBROUTINE DIFERF(X1,E1,A1,X2,E2,A2,E,A)
1692      C
1693      C      ----- THIS ROUTINE COMPUTES E AND A SUCH THAT
1694      C      EXP(E) * A = ERFC(X1) - ERFC(X2)
1695      C
1696      C      WHERE: ERFC (ABS(X1)) = EXP(E1) * A1
1697      C      ERFC (ABS(X2)) = EXP(E2) * A2
1698      C
1699      IMPLICIT REAL*8 (A-H,O-Z)
1700      J=1
1701      IF(X1) 3,2,1
1702      1   J=J+1
1703      2   J=J+1
1704      3   IF(X2) 6,5,4
1705      4   J=J+3
1706      5   J=J+3
1707      C
1708      C      -----X1   X2   J
1709      C      -       -   1
1710      C      0       -   2
1711      C      +       -   3
1712      C      -       0   4
1713      C      0       0   5
1714      C      +       0   6
1715      C      -       +   7
1716      C      0       +   8
1717      C      +       +   9
1718      C
1719      6   GO TO (10,20,20,20,30,10,20,10,10),J
1720      10  A12=A1-A2
1721      11  GO TO (12,13,13,12,13,13,12,13,13),J
1722      12  A12=-A12
1723      13  E=0.D0
1724          A=A12
1725          RETURN
1726      C
1727      20  A12=A1+A2
1728          A12=A12-2.D0
1729          GO TO 11
1730      C
1731      30  E12=0.D0
1732          A12=A1-A2
1733          GO TO 13
1734      END

```

1735		FUNCTION BATE(T,T0,III)	1735
1736	C		1736
1737	C	THIS ROUTINE CALCULATES THE DECAY OF TWO- AND THREE- NUCLIDE	1737
1738	C	CHAINS WITH EQUAL RETENTION COEFFICIENTS	1738
1739	C		1739
1740		COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DNR(3),	1740
1741	*	GAP(6),BREAK,DUR,PE,ETA,TD,WATVEL,PATH	1741
1742		REAL*8 ENAME	1742
1743		REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,BREAK,DUR,PE,ETA,TD,WATVEL,PATH.	1743
1744	*	ARG1,ARG2,ARG3	1744
1745		BATE=0.0	1745
1746		AN1=AQ(1)	1746
1747		IF(AN1.LE.0.0)RETURN	1747
1748		AL1=ALAMB(1)	1748
1749		AL2=ALAMB(2)	1749
1750		ARG1=-AL1*(T-T0)	1750
1751		ARG2=-AL2*(T-T0)	1751
1752		X1=TEXP(ARG1)	1752
1753		X2=TEXP(ARG2)	1753
1754		IF(III.EQ.3)GO TO 10	1754
1755		BATE=AN1*AL2/(AL2-AL1)*(X1-X2)	1755
1756		RETURN	1756
1757	10	CONTINUE	1757
1758		AL3=ALAMB(3)	1758
1759		ARG3=-AL3*(T-T0)	1759
1760		X3=TEXP(ARG3)	1760
1761		BATE=AN1*AL2*AL3*(X1/(AL2-AL1)/(AL3-AL1)+X2/(AL1-AL2)/(AL3-AL2)+	1761
1762	*	*X3/(AL1-AL3)/(AL2-AL3))	1762
1763		RETURN	1763
1764		END	1764
1765		SUBROUTINE TOPDEF	1765
1766	C		1766
1767	C	THIS ROUTINE SORTS OUT ALL ZERO DISCHARGE RATES. IT ALSO	1767
1768	C	IDENTIFYS THE MAXIMUM DISCHARGE RATE.	1768
1769	C		1769
1770		COMMON/C2/ IS,BAND(304),BANDND(304),TIME(304),THETA(304),	1770
1771	*	TSTART,TFIN,TPEAK,BMAX,DINO	1771
1772		DATA C /1.E-15/	1772
1773		J1=0	1773
1774		J2=0	1774
1775		TSTART=0.0	1775
1776		TFIN=0.0	1776
1777		TPEAK=0.0	1777
1778		BMAX=0.0	1778
1779		AMAX=0.0	1779
1780		IF(BAND(1).LE.1.E-15)GO TO 1120	1780
1781		J1=1	1781
1782		TSTART=TIME(1)	1782
1783	1120	CONTINUE	1783
1784		DO 1160 J=2,IS	1784
1785		IF(BAND(J-1).LE.C.AND.BAND(J).GT.C.AND.J1.EQ.0)J1=J	1785
1786		IF(BAND(J-1).GT.C.AND.BAND(J).LE.C)J2=J-1	1786
1787		IF(BAND(J).LE.BMAX)GO TO 1140	1787
1788		TPEAK=TIME(J)	1788
1789		BMAX=BAND(J)	1789
1790	1140	CONTINUE	1790

1791		IF(BANDND(J).LE.AMAX)GO TO 1160	1791
1792		AMAX=BANDND(J)	1792
1793	1160	CONTINUE	1793
1794		IF(AMAX)1200,1200,1180	1794
1795	1180	DINO=BMAX/AMAX	1795
1796	1200	CONTINUE	1796
1797		IF(J1.EQ.0.AND.J2.EQ.0)GO TO 100	1797
1798		IF(J1.GT.0.AND.J2.EQ.0)J2=IS	1798
1799		IF(J1.EQ.0.AND.J2.GT.0)J1=1	1799
1800		TSTART=TIME(J1)	1800
1801		TFIN=TIME(J2)	1801
1802	100	IF(J1.GT.1)GO TO 200	1802
1803		IS=J2	1803
1804		GO TO 9999	1804
1805	200	IS=J2-J1+1	1805
1806		DO 300 I=1,IS	1806
1807		II=J1+I-1	1807
1808		TIME(I)=TIME(II)	1808
1809		THETA(I)=THETA(II)	1809
1810		BAND(I)=BAND(II)	1810
1811		BANDND(I)=BANDND(II)	1811
1812	300	CONTINUE	1812
1813	9999	CONTINUE	1813
1814		RETURN	1814
1815		END	1815
<hr/>			
1816		SUBROUTINE PRINTX(NCHAIN)	1816
1817	C		1817
1818	C	THIS ROUTINE WRITES THE LINEPRINTER OUTPUT FOR THE CHAIN FRAGMENTS	1818
1819	C		1819
1820		COMMON/FFFFF/ ENAME(3),THALF(3),AQ(3),ALAMB(3),EQ(3),DEC(3),	1820
1821	*	GAP(6),TO,DUR,PE,ETA,TD,V0,PATH	1821
1822		COMMON/C2/ IS,BAND(304),BANDND(304),TIME(304),THETA(304),	1822
1823	*	TSTART,TFIN,TPEAK,BMAX,DINO	1823
1824		REAL*8 ENAME,TEXT1(3)	1824
1825		REAL*8 THALF,AQ,ALAMB,EQ,DEC,GAP,TO,DUR,PE,ETA,TD,V0,PATH	1825
1826		DATA TEXT1 /'SINGLE','DOUBLE','TRIPLE'/	1826
1827		WATTIM=PATH/V0	1827
1828	1	FORMAT('1',A6,'-CHAIN MIGRATION CALCULATION'/' ' ,34('*')/'0NUCLIDE	1828
1829	*	: ',A6)	1829
1830	2	FORMAT('+',T17,'FROM ',A6)	1830
1831	3	FORMAT('+',T28,'DECAYING VIA ',A6)	1831
1832		WRITE(9,1) TEXT1(NCHAIN),ENAME(NCHAIN)	1832
1833		IF(NCHAIN.GT.1)WRITE(9,2) ENAME(1)	1833
1834		IF(NCHAIN.EQ.3)WRITE(9,3)ENAME(2)	1834
1835	4	FORMAT('0DESCRIPTION OF LEACH SCENARIO:'/'+' ,29('_'))/'0TIME OF LEA	1835
1836	*	CH INCIDENT:' ,1PG16.6 ,' YEARS AFTER REACTOR DISCHARGE'/' LEACH D	1836
1837	*	URATION:' ,10X,G14.6,' YEARS'/' MIGRATION PATH LENGTH:' ,3X,G14.6,	1837
1838	*	' METERS'/' GROUNDWATER VELOCITY:' ,4X,G14.6,' METERS/YEAR'/' GRO	1838
1839	*	UNDWATER TRAVEL TIME:' ,1X,G14.6,' YEARS'/' PECLET NUMBER:' ,11X,	1839
1840	*	G14.6/'0NUCLIDE DATA'/'+' ,12('_'),40X,A6,6X,A6,6X,A6)	1840
1841		WRITE(9,4) TO,DUR,PATH,V0,WATTIM,PE,(ENAME(I),I=1,NCHAIN)	1841
1842	5	FORMAT('0ACTIVITY AT THE TIME OF LEACH INCIDENT (CURIES)' ,1X,1PE12	1842
1843	*	.2,2E12.2)	1843
1844		WRITE(9,5)(AQ(I),I=1,NCHAIN)	1844
1845		DO 15 I=1,NCHAIN	1845
1846		GAP(I+3)=GAP(I+3)*1.93729E-6	1846

```

1847 15 GAP(I)=GAP(I)*1.93729E-6
1848 6 FORMAT(' MASS AT THE TIME OF LEACH INCIDENT',5X,'(MOLES)',2X,1PE12
1849 *.2,2E12.2)
1850 WRITE(9,6)(GAP(I),I=1,NCHAIN)
1851 7 FORMAT(' MASS AT THE END OF LEACH INCIDENT',6X,'(MOLES)',2X,1PE12.
1852 *.2,2E12.2)
1853 NN=NCHAIN+3
1854 WRITE(9,7)(GAP(I),I=4,NN)
1855 9 FORMAT(' DECAY NUMBER',36X,1PE12.2,2E12.2)
1856 WRITE(9,9)(DEC(I),I=1,NCHAIN)
1857 8 FORMAT(' RETENTION COEFFICIENT',27X,3F12.0)
1858 WRITE(9,8)(EQ(I),I=1,NCHAIN)
1859 10 FORMAT('OBRIEF PEAK INFORMATION'/'+' ,22(' ')/'OTIME OF INITIAL DIS
1860 *CHARGE',1PG19.6 , ' YEARS AFTER REACTOR DISCHARGE'/' TIME OF PEAK
1861 *TAIL',13X,G14.6,' YEARS AFTER REACTOR DISCHARGE'/' TIME OF PEAK M
1862 *AXIMUM',10X,G14.6,' YEARS AFTER REACTOR DISCHARGE'/' MAXIMUM DISC
1863 *HARGE RATE',8X,1PE14.2,' CURIES/YEAR'/' NUMBER OF TIMESTEPS USED'
1864 *.15X,IS/' DISP TO NO-DISP PEAKS',9X,0PF14.8)
1865 WRITE(9,10) TSTART,TFIN,TPEAK,BMAX,IS,DINO
1866 11 FORMAT('///6X,'TIME',11X,'DIM.LESS TIME',2X,'BAND RELEASE',3X,'D:0
1867 * NO-DISP'/'
1868 IF(IS.EQ.0)GO TO 100
1869 WRITE(9,11)
1870 12 FORMAT(1X,1PE15.4,3E15.4)
1871 DO 20 I=1,IS
1872 20 WRITE(9,12) TIME(I),THETA(I),BAND(I),BANDND(I)
1873 100 CONTINUE
1874 RETURN
1875 END
1876 SUBROUTINE SPEC(GNAME,ENAME,IFILE,IAPPR,FACT,*)
1877 C
1878 C THIS ROUTINE CALCULATES SPECIAL NUCLIDE CONTRIBUTIONS BASED ON
1879 C EITHER OF TWO DECAY APPROXIMATIONS:
1880 C IAPPR=1: SHORTLIVED DAUGHTER NUCLIDE IN RADIOACTIVE EQUILLIB-
1881 C RIUM WHITH ITS PREDECESSOR.
1882 C IAPPR=2: SHORTLIVED PARENT NUCLIDE WHICH IS ASSUMED TO DECAY
1883 C COMPLETELY BEFORE LEACHING.
1884 C GNAME IS THE CONTRIBUTION TO BE CALCULATED. ENAME IS THE ALREADY
1885 C CALCULATED CHAIN UPON WICH THE APPROXIMATIVE CALCULATION
1886 C OF GNAME IS BASED. ENAME IS TO BE FOUND IN DATASET NR 12+IFILE
1887 C IF IAPPR=2 THE DISCHARGE RATE FOR GNAME IS CALCULATED BY MULTI-
1888 C PLYING THE DISCHARGE RATE FOR ENAME BY FACT. IF NO DISCHARGE
1889 C RATE CAN BE CALCULATED THE CONTROL IS TRANSFERRED TO THE STATE-
1890 C MENT LABELED * IN THE CALLING ROUTINE.
1891 C
1892 COMMON /PRINT/ MIGOUT
1893 COMMON /C2/ NSTEP,B(304),DUM1(304),T(304),DUM3(304),
1894 * TD1,TD2,TD3,BD,DD
1895 COMMON /PEAKS/ ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),
1896 * CMAX(75,10),DOSE2(75)
1897 REAL*8 GNAME(3),ENAME(3),ANAME(3),TEXT(2),BLANK
1898 LOGICAL MIGOUT
1899 DATA BLANK/' ' ,TEXT/'DAUGHTER','PARENT ' /
1900 C
1901 IF=IFILE
1902 ID=12+IF

```

1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902

316-CMPACK	PAGE	40	
1903	10	CONTINUE	1903
1904		READ(ID,END=900)ANAME,IS	1904
1905		DO 20 I=1,IF	1905
1906		IF(ANAME(I).NE.ENAME(I))GO TO 30	1906
1907	20	CONTINUE	1907
1908		GO TO 50	1908
1909	30	IF(IS.EQ.0)GO TO 10	1909
1910		READ(ID)(T(I),B(I),I=1,IS)	1910
1911		GO TO 10	1911
1912	50	IF(IS.EQ.0)GO TO 800	1912
1913		READ(ID)(T(I),B(I),I=1,IS)	1913
1914		WRITE(16)GNAME,IS	1914
1915		IF(IAPPR.EQ.1)GO TO 70	1915
1916		DO 60 I=1,IS	1916
1917		B(I)=B(I)*FACT	1917
1918	60	CONTINUE	1918
1919	70	CONTINUE	1919
1920		WRITE(16)(T(I),B(I),I=1,IS)	1920
1921		BMAX=B(1)	1921
1922		IMAX=1	1922
1923		DO 80 I=2,IS	1923
1924		IF(B(I).LE.BMAX)GO TO 80	1924
1925		BMAX=B(I)	1925
1926		IMAX=I	1926
1927	80	CONTINUE	1927
1928		IJ=INTNUK(GNAME(1))	1928
1929		ICONTR(IJ)=ICONTR(IJ)+1	1929
1930		NN=ICONTR(IJ)	1930
1931		TSTART(IJ,NN)=T(1)	1931
1932		TPEAK(IJ,NN)=T(IMAX)	1932
1933		CMAX(IJ,NN)=BMAX	1933
1934		TEND(IJ,NN)=T(IS)	1934
1935		REWIND ID	1935
1936		IF(.NOT.MIGOUT)RETURN	1936
1937		WRITE(9,91)GNAME(1),GNAME(2)	1937
1938		IF(GNAME(3).NE.BLANK)WRITE(9,92)GNAME(3)	1938
1939	91	FORMAT('1APPROXIMATIVE DISCHARGE RATE FOR ',A6,' FROM ',A6)	1939
1940	92	FORMAT('+',51X,' AND ',A6)	1940
1941		WRITE(9,93)TEXT(IAPPR),ENAME	1941
1942	93	FORMAT(' A SHORTLIVED ',A8,'-NUCLIDE APPROXIMATION WAS MADE ON THE	1942
1943		* BASIS OF THE CHAIN ',3A8/)	1943
1944		WRITE(9,94)	1944
1945	94	FORMAT('0',5X,' TIME ',11X,' BAND-RELEASE'/)	1945
1946		DO 95 I=1,IS	1946
1947		WRITE(9,96)T(I),B(I)	1947
1948	95	CONTINUE	1948
1949	96	FORMAT(1PE16.4,E15.4)	1949
1950		RETURN	1950
1951	800	CONTINUE	1951
1952		WRITE(6,801)GNAME(1),GNAME(2)	1952
1953		IF(GNAME(3).NE.BLANK)WRITE(6,802)GNAME(3)	1953
1954	801	FORMAT('0THE CHAIN ',A6,' FROM ',A6)	1954
1955	802	FORMAT('+',29X,' AND ',A6)	1955
1956		WRITE(6,803)ID,(ENAME(I),I=1,IF)	1956
1957	803	FORMAT(' WAS NOT CALCULATED BECAUSE OF ZERO RESULT IN DATASET NR '	1957
1958		*,I3,' FOR ',3A8/)	1958

1959		REWIND ID	1959
1960		RETURN 1	1960
1961	900	CONTINUE	1961
1962		WRITE(6,901)GNAME(1),GNAME(2)	1962
1963		IF(GNAME(3).NE.BLANK)WRITE(6,902)GNAME(3)	1963
1964	901	FORMAT('OWHILE EXECUTING SPEC FOR ',A6,' FROM ',A6)	1964
1965	902	FORMAT('+',44X,' AND ',A6)	1965
1966		WRITE(6,903)ID,(ENAME(I),I=1,IF)	1966
1967	903	FORMAT(' DATASET NR ',I3,' DID NOT CONTAIN THE CHAIN ',3A8/)	1967
1968		REWIND ID	1968
1969		RETURN 1	1969
1970		END	1970
1971		SUBROUTINE CHAIN(NAME,N,IC1,I2)	1971
1972	C		1972
1973	C	THIS ROUTINE RULES THE ADDITION OF DIFFERENT DECAY CHAIN	1973
1974	C	FRAGMENTS TO A NUCLIDES TOTAL DISCHARGE RATE	1974
1975	C		1975
1976		COMMON/C2/ NSTEP,T(304),Y(304),BAND(304),TIME(304),	1976
1977		* TBEGIN,TFIN,TTOPP,BRPEAK,DINO	1977
1978		COMMON/PEAKS/ICONTR(75),TSTART(75,10),TPEAK(75,10),TEND(75,10),	1978
1979		* CMAX(75,10),DOSE2(75)	1979
1980		REAL*8 NAME,IC1(20),IBL,NN(3)	1980
1981		DATA IBL/' /	1981
1982		DO 15 I=1,N	1982
1983	15	Y(I)=0.0	1983
1984		IJ=INTNUK(NAME)	1984
1985		NJ=ICONTR(IJ)	1985
1986		DOSE=DOSE2(IJ)	1986
1987		I2=0	1987
1988		J2=0	1988
1989		DO 150 JJJ=13,16	1989
1990		JJ=JJJ	1990
1991		II=JJ-12	1991
1992		IF(JJ.GT.15)II=1	1992
1993	110	CONTINUE	1993
1994		IF(J2.GE.NJ)GO TO 155	1994
1995		READ(JJ,END=150)(NN(I),I=1,3),ISTEP	1995
1996		IF(ISTEP.EQ.0)GO TO 110	1996
1997		READ(JJ)(TIME(I),BAND(I),I=1,ISTEP)	1997
1998		IF(NN(II).NE.NAME)GO TO 110	1998
1999		CALL INTPOL(N,ISTEP)	1999
2000		IF(JJ.GT.15)GO TO 120	2000
2001		J2=J2+1	2001
2002		I2=I2+1	2002
2003		IC1(I2)=NN(1)	2003
2004		GO TO 110	2004
2005	120	CONTINUE	2005
2006		J2=J2+1	2006
2007		I2=I2+1	2007
2008		IC1(I2)=NN(2)	2008
2009		IF(NN(3).EQ.IBL)GO TO 110	2009
2010		I2=I2+1	2010
2011		IC1(I2)=NN(3)	2011
2012		GO TO 110	2012
2013	150	CONTINUE	2013
2014	155	CONTINUE	2014

2015		DO 160 JJ=13,16	2015
2016		REWIND JJ	2016
2017	160	CONTINUE	2017
2018		RETURN	2018
2019		END	2019
<hr/>			
2020		SUBROUTINE INTPOL(N,ISTEP)	2020
2021	C		2021
2022	C	THIS ROUTINE IS USED BY CHAIN FOR THE ADDITION OF	2022
2023	C	DIFFERENT CHAIN FRAGMENTS	2023
2024	C		2024
2025		COMMON/C2/ NSTEP,T(304),Y(304),BAND(304),TIME(304),	2025
2026	*	TBEGIN,TFIN,TTOPP,ERPEAK,DINO	2026
2027		J1=1	2027
2028		DO 120 I=1,N	2028
2029	116	IF(J1.GT.ISTEP-1)GC TO 125	2029
2030		IF(T(I).GE.TIME(J1).AND.T(I).LT.TIME(J1+1).AND.BAND(J1).GT.0.0	2030
2031	*	.AND.BAND(J1+1).GT.0.0) GO TO 117	2031
2032		IF(T(I).LT.TIME(J1))GO TO 120	2032
2033		J1=J1+1	2033
2034		GO TO 116	2034
2035	117	Y(I)=Y(I)+BAND(J1)+(T(I)-TIME(J1))/(TIME(J1+1)-TIME(J1))*	2035
2036		*(BAND(J1+1)-BAND(J1))	2036
2037	120	CONTINUE	2037
2038	125	CONTINUE	2038
2039		RETURN	2039
2040		END	2040
<hr/>			
2041		SUBROUTINE TSCALE(YMAX,TIMY,NN,TIMAX,TIMIN,U,NT,*)	2041
2042	C		2042
2043	C	THIS SUBROUTINE CALCULATES A LOGARITHMIC TIMESCALE, U, BETWEEN	2043
2044	C	THE POINTS T1 AND T2. T1 AND T2 ARE CHOSEN TO BE SO CLOSE TO	2044
2045	C	TMIN AND TMAX RESPECTIVELY AS POSSIBLE. ONE OR TWO POINTS IN U ARE	2045
2046	C	FORCED TO BE PLACED AT TP1 AND TP2.	2046
2047		REAL*8 THALF,AQ,ALAMB,EQ,DNR,GAP,T0,T1,PE,ETA,TD,V,AL	2047
2048	C		2048
2049		DIMENSION U(304)	2049
2050		DIMENSION YMAX(1),TIMY(1)	2050
2051		DATA IS/100/	2051
2052		REAL*8 BASE,T,TP1,TP2,TMAX,TMIN,T1	2052
2053		AMAX=0.0	2053
2054		IMAX=0	2054
2055		TMAX=TIMAX	2055
2056		TMIN=TIMIN	2056
2057		DO 10 I=1,NN	2057
2058		IF(YMAX(I).LE.AMAX)GO TO 10	2058
2059		AMAX=YMAX(I)	2059
2060		IMAX=I	2060
2061	10	CONTINUE	2061
2062		IF(IMAX.EQ.0)RETURN 1	2062
2063		JMAX=0.0	2063
2064		BMAX=0.0	2064
2065		DO 20 I=1,NN	2065
2066		IF(YMAX(I).LE.BMAX)GO TO 20	2066
2067		IF(I.EQ.IMAX)GO TO 20	2067
2068		JMAX=I	2068
2069		BMAX=YMAX(I)	2069
2070	20	CONTINUE	2070

2071		IF(JMAX.EQ.0.OR.BMAX.LT.1.E-20)GO TO 30	2071
2072		TP2=AMAX1(TIMY(IMAX),TIMY(JMAX))	2072
2073		TP1=AMIN1(TIMY(IMAX),TIMY(JMAX))	2073
2074	C		2074
2075	C	CALCULATE TIMESTEPS FOR TWO NON-SIMULTANEOUS PEAKS	2075
2076	C		2076
2077		A1=DLOG(TP2/TP1)/DLOG(TMAX/TMIN)*IS	2077
2078		IF(A1.LT.0.5)GO TO 30	2078
2079		N1=IFIX(A1)+1	2079
2080		BASE=(TP2/TP1)**(1/FLOAT(N1))	2080
2081		A2=DLOG(TP1/TMIN)/DLOG(BASE)+1	2081
2082		N2=IFIX(A2)	2082
2083		T1=TP1/BASE**(N2)	2083
2084		A3=DLOG(TMAX/TP2)/DLOG(BASE)+1	2084
2085		N3=IFIX(A3)	2085
2086		T2=TP2*BASE**N3	2086
2087		NT=N1+N2+N3	2087
2088		GO TO 70	2088
2089	C		2089
2090	C	CALCULATE TIMESTEPS FOR SINGLE OR SIMULTANEOUS PEAKS	2090
2091	C		2091
2092		30 CONTINUE	2092
2093		TP2=TIMY(IMAX)	2093
2094		IF(TP2.EQ.TMIN.OR.TP2.EQ.TMAX)GO TO 60	2094
2095		A1=DLOG(TP2/TMIN)/DLOG(TMAX/TMIN)*IS	2095
2096		IF(A1.LT.0.5)GO TO 50	2096
2097		N1=IFIX(A1)+1	2097
2098		BASE=(TP2/TMIN)**(1/FLOAT(N1))	2098
2099		A2=DLOG(TMAX/TP2)/DLOG(BASE)+1	2099
2100		N2=IFIX(A2)	2100
2101		T2=TP2*BASE**N2	2101
2102		T1=TMIN	2102
2103		NT=N1+N2	2103
2104		GO TO 70	2104
2105		50 CONTINUE	2105
2106		A2=DLOG(TMAX/TP2)/DLOG(TMAX/TMIN)*IS	2106
2107		N2=IFIX(A2)+1	2107
2108		T2=TMAX	2108
2109		BASE=(TMAX/TP2)**(1/FLOAT(N2))	2109
2110		T1=TP2/BASE	2110
2111		NT=N2+1	2111
2112		GO TO 70	2112
2113	C		2113
2114	C	CALCULATE TIMESTEPS WHEN PEAKS COINCIDE WITH MAX OR MIN	2114
2115	C		2115
2116		60 CONTINUE	2116
2117		BASE=(TMAX/TMIN)**(1/FLOAT(IS)-1)	2117
2118		T1=TMIN	2118
2119		T2=TMAX	2119
2120		NT=IS	2120
2121	C		2121
2122	C	CALCULATE TIMESCALE	2122
2123	C		2123
2124		70 CONTINUE	2124
2125		U(1)=T1	2125
2126		T=T1	2126

2127		DO 100 J=2,NT	2127
2128		T=T*BASE	2128
2129	100	U(J)=T	2129
2130		RETURN	2130
2131		END	2131
<hr/>			
2132		SUBROUTINE PLOT1(NAME,NN,NR,NS,YMAX,WRITEX)	2132
2133	C		2133
2134	C	THIS ROUTINE ARRANGES THE DIFFERENT VARIABLES BEFORE PLOTTING	2134
2135	C		2135
2136		COMMON/C2/ NSTEP,X(304),Y(304),DOSE1(304),Y1(304),	2136
2137		* TBEGIN,TFIN,TTOPP,BMAX,DINO	2137
2138		DIMENSION YREAL(304)	2138
2139		REAL*8 YNAME(21),NAME,NN(20)	2139
2140		LOGICAL WRITEX,AREADY	2140
2141		DATA NCOL/110/,IU/20/,NPC/1/,IDEC/2/	2141
2142		YNAME(1)=NAME	2142
2143		DO 1 I=1,NR	2143
2144	1	YNAME(I+1)=NN(I)	2144
2145		NR=NR+1	2145
2146		AREADY=.FALSE.	2146
2147		IS=-1	2147
2148		YIMIN=-12	2148
2149		YL=3.	2149
2150		IF(YMAX.LE.0.0)WRITE(6,3)YNAME(1)	2150
2151	3	FORMAT('OYMAX.LE.0 IN PLOT1 FOR ',A6)	2151
2152		IF(YMAX.LE.0.0)RETURN	2152
2153		YIMAX=ALOG10(YMAX)	2153
2154		IND2=NS	2154
2155		I1=1	2155
2156		I2=NS	2156
2157	50	CONTINUE	2157
2158		IND1=I1	2158
2159		IF(I1.GE.IND2)RETURN	2159
2160		DO 300 I=IND1,IND2	2160
2161		IF(Y(I).LE.0.0.AND.I.EQ.I1)GO TO 200	2161
2162		IF(Y(I).LE.0.0)GO TO 100	2162
2163		Y1(I)=ALOG10(Y(I))	2163
2164		YREAL(I)=Y(I)	2164
2165		GO TO 300	2165
2166	100	CONTINUE	2166
2167		I2=I-1	2167
2168		CALL PLOTY(I1,I2,NPC,YNAME,YIMIN,YIMAX,YL,0.,NCCL,IU,IS,	2168
2169		* WRITEX,IDEC,NR,YREAL)	2169
2170		AREADY=.TRUE.	2170
2171		I1=I+1	2171
2172		GO TO 50	2172
2173	200	CONTINUE	2173
2174		I1=I1+1	2174
2175	300	CONTINUE	2175
2176		IF(AREADY)RETURN	2176
2177		CALL PLOTY(I1,I2,NPC,YNAME,YIMIN,YIMAX,YL,0.,NCCL,IU,IS,	2177
2178		* WRITEX,IDEC,NR,YREAL)	2178
2179		RETURN	2179
2180		END	2180
2181		FUNCTION TERFC(X)	2181
2182		IF(X.GT.12.0)GO TO 1	2182

2183		TERFC=ERFC(X)	2183
2184		GO TO 2	2184
2185	1	TERFC=0.0	2185
2186	2	CONTINUE	2186
2187		RETURN	2187
2188		END	2188

2189		SUBROUTINE PLOTY(I1,I2,NPC,NAME,YMI,YMA,YL,FI,NCCL,IU,IS,	2189
2190	*	WRITEX,IDEC,NR,YREAL)	2190
2191	*	COMMON/C2/ NSTEP,X(304),Z(304),TIME(304),Y(304),	2191
2192	*	TBEGIN,TFIN,TTOPP,BRPEAK,DINO	2192
2193	C		2193
2194	C	LINE PRINTER PLOTTING ROUTINE WITH VERTICAL X-AXIS.	2194
2195	C	THE STEP IN X-DIRECTION IS CONSTANT = ONE LINE.	2195
2196	C	Y = THE ARRAY CONTAINING THE VALUES TO BE PLOTTED	2196
2197	C	I1 = THE FIRST Y-ELEMENT TO BE PLOTTED AND THE FIRST LINE	2197
2198	C	NUMBER AS WELL.	2198
2199	C	I2 = THE LAST LINE NUMBER TO BE PLOTTED AND THE LAST	2199
2200	C	Y-ELEMENT AS WELL, IF NPC=1.	2200
2201	C	NPC = THE NUMBER OF CURVES TO BE PLOTTED SIMULTANEOUSLY.	2201
2202	C	NPC MAY NOT EXCEED NPMAX..	2202
2203	C	IF FOR INSTANCE NPC=3 AND THE VARIABLES TO BE PLOTTED	2203
2204	C	ARE A, B AND C, THE VALUES SHALL BE STORED IN THE	2204
2205	C	Y-ARRAY AS FOLLOWS: A1,B1,C1, A2,B2,C2, A3,B3,C3, ...	2205
2206	C	NAME = THE NAME(S) OF THE PLOTTED VARIABLE(S) (6H-HOLLERITH)	2206
2207	C	IF NPC>1, THIS PARAMETER IN THE CALLING PROGRAM MUST	2207
2208	C	BE AN ARRAY, WHERE THE NAMES ARE STORED IN THE SAME	2208
2209	C	ORDER AS ARE THE VALUES IN THE Y-ARRAY.	2209
2210	C	YMI = EXTERNALLY SET LOWER Y-LIMIT (FOR THE SCALE-CALC.)	2210
2211	C	YMA = EXTERNALLY SET UPPER Y-LIMIT (FOR THE SCALE-CALC.)	2211
2212	C	YL = 0: IF YMI<YMA, THE LIMITS ARE TAKEN FROM YMI AND YMA,	2212
2213	C	ELSE THE LIMITS ARE COMPUTED FROM Y, INDEPENDENTLY	2213
2214	C	OF YMI AND YMA.	2214
2215	C	= 1: TAKING THE LIMITS FROM YMI AND YMA, BUT ADJUSTING	2215
2216	C	YMIN, IF ANY Y<YMI.	2216
2217	C	= 2: TAKING THE LIMITS FROM YMI AND YMA, BUT ADJUSTING	2217
2218	C	YMAX, IF ANY Y>YMA.	2218
2219	C	= 3: 1. AND 2.	2219
2220	C	FI > 0: THE SPACE BETWEEN X-AXIS AND THE Y-POSITION IS	2220
2221	C	FILLED WITH XMARKS (STAPLE DIAGRAM).	2221
2222	C	ORIGO IS POSITIONED AT Y=0. NO XMARK FOR Y=0.	2222
2223	C	THE EVERY 100 LINES NUMBER MARKING IS SUPPRESSED.	2223
2224	C	THE STAPLE DIAGRAM IS POSSIBLE ONLY IF NPC=1.	2224
2225	C	NCOL = MAXIMAL NUMBER OF COLUMNS AVAILABLE DURING PRINTING	2225
2226	C	IU = OUTPUT UNIT NUMBER	2226
2227	C	IS = A SWITCH, THAT SHALL BE SET EITHER <0 (XPLOT THEN	2227
2228	C	STARTS ITS PRINTING ON NEW PAGE) OR =0 (THE PRINTING	2228
2229	C	STARTS AFTER A SINGLE LINE-SKIP) BEFORE THE FIRST	2229
2230	C	CALL. PLOTY RESETS IS ITSELF. AT A REPEATED CALL THE	2230
2231	C	PLOTTING CONTINUES ON THE ORIGINAL X-AXIS.	2231
2232	C	THE SUBROUTINE SCALE(A,B,C,I) COMPUTES THE SCALE OUT OF THE	2232
2233	C	PREL. VALUE A, AND PUTS THE RESULT IN B. C IS THE PRINTED	2233
2234	C	SCALE AND I IS THE POWER OF 10.	2234
2235	C	THE SUBROUTINE DECODH(N,A,I,J) DECODES THE INTEGER NUMBER N	2235
2236	C	AND PUTS THE J RIGHTMOST FIGURES CONVERTED INTO 1H-HOLLERITH	2236
2237	C	INTO THE ARRAY A FROM A(I) TO A(I+J-1). LEFT ZEROS BLANKED.	2237
2238	C		2238

316-CMPACK	PAGE	46	
2239		LOGICAL WRITEY,WRITEX	2239
2240		LOGICAL JUMP,OUT,FILL,SINGLE	2240
2241	C		2241
2242		DIMENSION IX(16),IY(15),XD(2),YAXIS(2),HNXO(3),FMX(9),FM1(9),	2242
2243	1	FM2(8),FM3(14),FM4(14),FM5(7),FM6(11),FM7(5),FM8(9),	2243
2244	2	NAME(1)	2244
2245		DIMENSION YREAL(1)	2245
2246		REAL*8 NAME	2246
2247		REAL*8 FM9(11),FM10(11),RUBR(3),FME(2),FMT(2),FMS(3)	2247
2248		REAL*8 DUM1(3),DUM2(2),DUM3(2),FM1,FM2,FM3,FM4,FMS,	2248
2249	1	FM6,FM7,FM8,FMX,YAXIS,BLANK,S,PLURAL	2249
2250	2	,YMARK,XMARK,DCT	2250
2251	3	,HNXO,XD	2251
2252		EQUIVALENCE (FM1(7),DUM1(1)),(FM2(7),DUM2(1)),	2252
2253	1	(FM5(6),DUM3(1))	2253
2254	C		2254
2255		EQUIVALENCE (YMARK(1),XMARK)	2255
2256	C		2256
2257	C	TO INCREASE THE PERMITTED NUMBER OF SIMULTANEOUSLY PLOTTED	2257
2258	C	CURVES, JUST INCREASE NPMAX AND EXTEND THE YMARK DIMENSION	2258
2259	C	AND DATA ITEM LIST..	2259
2260		DIMENSION YMARK(4)	2260
2261		DATA NPMAX/4/, YMARK/1HX,1HO,1HH,1HS/	2261
2262	C		2262
2263		DATA FM9(1)/5H(1H+/,FM9(5)/2HX,/,FM9(6)/8H7HX-VALU/,	2263
2264	1	FM9(7)/2HE,/,FM9(9)/8HX,7HY-VA/,FM9(10)/4HLUE)/	2264
2265	2	,FM9(11)/1H /,FME/5H2(1PE,2H))/	2265
2266	3	,FMS/3H1PE,1H),2HE)/	2266
2267		DATA MAXCOL/132/	2267
2268	C		2268
2269		DATA DOT/1H./, IAST/1H*/, IPLUS/1H+/, BLANK/1H /, PLURAL/1HS/,	2269
2270	1	YAXIS/5H*.....5H...../, FMX/2HX,,6H(5X,A1,5H,4X),,3H(I6,	2270
2271	2	4H(A1,,4H9X),,2HA1,4H(4X,,3HI6)/	2271
2272	C		2272
2273		DATA FM1(1)/5H(1H ,/, DUM1/6H(I6,4X,5H),5X,,4H1HY)/,	2273
2274	1	FM2(1)/5H(1H ,/,FM2(4)/2HX,/,FM3(1)/5H(1H ,/,DUM2	2274
2275	2	/6H(2A5),,4H1H>)/, FM3(1)/5H(1H ,/, FM3(9)/3HI5,/, FM3(14)/	2275
2276	3	3HA1)/, FM4(1)/5H(1H+/, FM4(9)/3H6X,/, FM4(14)/1H)/, FM5(1)/	2276
2277	4	5H(1H ,/, DUM3/3HI5,,4H1H.)/, FM6(1)/5H(1H ,/.	2277
2278	5	FM6(5)/5HX,A1,/, FM6(11)/1H)/,FM7(1)/5H(1H+/,FM7(5)/5HX,A1)/	2278
2279	6	,FM8(1)/5H(1H+/, FM8(5)/2HX,/, FM8(9)/3HA1)/	2279
2280	C		2280
2281		INDEX1 = I1	2281
2282		INDEX2 = I2	2282
2283		IF(IS.GT.0) GOTO 300	2283
2284	C		2284
2285		IF(NCOL.GE.19.AND.IU.GT.0.AND.NPC.LE.NPMAX) GOTO 10	2285
2286	5	PRINT 801, NAME	2286
2287		RETURN	2287
2288	10	NP = MAX0(1,NPC)	2288
2289		IO = IU	2289
2290		SINGLE = NP.EQ.1	2290
2291		WRITEX=WRITEX.AND.MAXCOL-NCCL.GT.IDEC+7	2291
2292		WRITEY=WRITEX.AND.SINGLE.AND.MAXCOL-NCOL.GT.2*IDEC+16	2292
2293		FMT(1)=FME(1)	2293
2294		FMT(2)=FME(2)	2294