0
1-7-74

Dr-980

UCRL-51577 Pt. 2

**A SYSTEM FOR PLUTONIUM ANALYSIS
BY GAMMA RAY SPECTROMETRY
PART 2: COMPUTER PROGRAMS FOR
DATA REDUCTION AND INTERPRETATION**

R. Gunnink

April 18, 1974

Prepared for U.S. Atomic Energy Commission under contract No. W-7405-Eng-48

**LAWRENCE
LIVERMORE
LABORATORY**

University of California/Livermore

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NOTES

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5385 Port Royal Road
Springfield, Virginia 22151

Price: Printed Copy \$ 2.50; Microfiche \$1.45

Pages

1-50
51-100
101-150
151-200
201-250
251-300

NTIS Selling Price

\$4.00
\$5.45
\$7.35
\$10.00
\$13.00

LAWRENCE LIVERMORE LABORATORY
University of California, Livermore, California 94550

UCRL-51577 Pt. 2

**A SYSTEM FOR PLUTONIUM ANALYSIS
BY GAMMA RAY SPECTROMETRY
PART 2: COMPUTER PROGRAMS FOR
DATA REDUCTION AND INTERPRETATION**

R. Gunnink

MS. date: April 18, 1974

— NOTICE —

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MAILED

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

fy

FOREWORD

This work was sponsored by the Office of Nuclear Materials Safeguards of the U.S. Atomic Energy Commission.

CONTENTS

	Page No.
ABSTRACT	1
I. INTRODUCTION	1
II. BINARY LOADING SCHEDULES	1
General Purpose Programs	2
DATARD	3
GRPANL	4
FILES	12
PKANAL	14
SANAL	20
SHAPE	23
Specific Purpose Programs	26
AMPUB	26
AXPUB	33
LEPA	41
LEPA2	46
PUB	55
Subroutine Programs	64
AVE	64
BKGD	65
BKGRD	66
BOUNDS	67
CHFIT	68
GFIT	69
MAXVAL	70
MINVAL	71
MLR	72
MLRA	73
MLRS	74
MLRE	76
NCTS	77
NCNTS	78
RDSK	79
RDDSK	80
STRIP	81
WDSK	82
Special I/O Subroutines	83
FLDAT	83
FSIO Package	89

Abstract

With the use of a computer-based spectrometer system, techniques have been developed whereby the isotopic abundances and total analysis of plutonium can be made by detecting and properly analyzing gamma rays emitted by a solution sample. Isotopic ratio mea-

surements within 0.05% for ^{239}Pu and within 0.5 to 1% for ^{240}Pu and ^{241}Pu can be obtained in counting periods as short as 10 minutes.

This report describes and lists a set of FORTRAN programs which have been used to perform these analyses.

I. INTRODUCTION

The traditional method for measuring the isotopic abundances of plutonium has been a combination of mass spectrometry and alpha pulse height analysis on chemically prepared samples. We have developed techniques using high-resolution gamma-ray spectrometry which is capable of making these measurements non-destructively, using less expensive equipment, and in counting periods as short as 10 minutes.

Part I in this series of reports describes the basis of the technique, the experimental approach used, and gives a summary of the performance. In this report, we describe and list a set of programs which we have successfully used to perform the types of analyses described in Part I. Other parts in the series will provide additional information on the performance, the system components, and interface designs, and on the data acquisition and display software.

Many of the data analysis procedures we developed are quite lengthy and complicated. This meant that the software language had to be well established and convenient to use, yet quite powerful. To obtain reasonable execution times, a language capable of being compiled was preferred to an interpretive language

such as BASIC or FOCAL. For these reasons, the data reduction and interpretation programs were all written in FORTRAN.

The programs were written to run on a 12-bit 12,000-word mini-computer such as the PDP-8E or DCC-112H interfaced to a disk mass-storage device. System operating programs such as PS-8, DS-8, or OS-8 can be used to provide the necessary library of system programs for editing and compiling files. Complete descriptions of these programs and instructions on how to use them can be found in a number of handbooks and publications produced by Digital Equipment Corporation. Since this information is available and well documented, we will not, in this report, elaborate on the implementation of the programs.

The set of programs divides qualitatively into three classifications. One is a set of general purpose programs which are useful for setting up and defining the experimental parameters. The second set of programs performs very specific spectral analyses and requires essentially no user interaction. The third group of programs is subroutines containing sets of instructions which are frequently used.

II. BINARY LOADING SCHEDULES

The following files are to be loaded for the programs listed.

PROGRAM	LOADING SCHEDULE
DATARD	DATARD, FSIO, FLDAT, RDSK
GRPANL	GRPANL, FSIO, FLDAT/I
LOADI	LOADI, FSIO, RDSK, MAXVAL, CHFIT, NCTS, AVE
FITI	FITI
PMLR1	PMLR1, MLR/O
PMLA1	PMLA1, MLRA/I
RETRN1	RETRN1, MLRE
OUTI	OUTI
FILES	FILES/I/O
PKANAL	PKANAL, FSIO, FLDAT, RDSK, MAXVAL, CHFIT, NCTS, AVE
FIT	FIT
PMLR	PMLR, MLR/O
PMLRA	PMLRA, MLRA/I

PROGRAM	LOADING SCHEDULE
PKANAL (CONT.)	
RETRN OUT Ø	RETRN OUT Ø
SANAL SHAP	SANAL SHAP/I/O
SHAPE NETCT	SHAPE, FSIO, FLDAT/I NETCT, FSIO, RDSK, MAXVAL, CHFIT, NCTS, AVE (This program also uses FIT, PMLR, PMLRA, RETRN, OUTØ, and SHAP.)
AMPUB MATRX4 XMLR4 XMLA4 OUT4	AMPUB, FSIO, FLDAT, RDSK, MAXVAL, NCNTS, NCTS, AVE/I MATRX4, MAXVAL, NCTS, AVE, CHFIT XMLR4, MLR/C XMLA4, MLRA/I OUT4, MLRE
AXPUB MATRX5 XMLR5 XMLA5 OUT5	AXPUB, FSIO, FLDAT, RDSK, MAXVAL, BOUNDS, NCTS, AVE, CHFIT, STRIP/I MATRX5 XMLR5, MLR/O XMLA5, MLRA/I OUT5, MLRE
LEPA PK45 REPORT	LEPA, FSIO, FLDAT, RDSK, MAXVAL, NCTS, AVE/I PK45, FSIO, RDSK, MAXVAL, AVE, CHFIT, NCTS/I REPORT
LEPA2 PKS45 PK1ØØ LSFIT REPRT	LEPA2, FSIO, FLDAT, RDSK, MAXVAL, NCNTS, NCTS, AVE/I PKS45, FSIO, RDSK, MAXVAL, AVE, CHFIT, NCTS/I PK1ØØ, FSIO, RDSK, MAXVAL, CHFIT, BOUNDS, STRIP/I, NCNTS, NCTS, AVE MLR, MLRS REPRT, MLRE
PUB FIT3 PMLR3 PMLA3 MATRX3 XMLR3 XMLA3 OUT3	PUB, FSIO, FLDAT, RDSK, MAXVAL, CHFIT, NCNTS/I, NCTS, BOUNDS, AVE FIT3 PMLR3, MLR3/O PMLA3, MLRA3, MLRE/I MATRX3, FSIO, RDSK, MAXVAL, NCTS, AVE/I XMLR3, MLR/O XMLA3, MLRA/I OUT3, MLRE

General Purpose Programs

The following set of programs are useful for defining certain parameters, particularly peak shape parameters, or for analyzing certain regions of a spectrum for such quantities as peak position, area, or shape.

The first routine to be used in setting up a new system is PKANAL. It allows one to determine the shape

characteristics of single peaks using only the spectral data and information provided to the program by teletype. Once the shape characteristics have been obtained for a low- and high-energy peak, these values can be used by SANAL to compute the generalized shape parameters. These then are stored as a disk file (called CFILEB in the following listings) using the

FILES routine. Once such a file exists, the set of shape parameters can automatically be redetermined and updated using the SHAPE Program. The general program GRPANL can now also be used for the analysis of complex peak groupings using the file of shape parameters that was previously stored.

CFILEB File Structure

The file contains peak shape parameters which are used by nearly all the analysis programs.

Entries 1 to 5 are the shape constants which are described in Part I of this report.

Entry 7 is the x-ray to gamma-ray area ratio at 100 keV.

Entries 1 to 7 are generated by the SANAL and SHAPE programs.

Entry 8 is the system gain expressed in keV/channel.

PROGRAM:
DATARD

PURPOSE:

To print by teletype the data contained in any designated region of the spectrum.

INPUT:

- 1) Spectrum ID.
- 2) Device on which spectrum is found.
- 3) The starting channel position of the listing.
- 4) The number of data points desired.
- 5) Name of floating point data file. (This name in "RAYXXX" if generated by "FLDAT" sub-routine.)
- 6) Device on which floated data is found. (This is "SYS@@@" if "FLDAT" is used.)

```

C      DATARD (FSIO,FLDAT,RDSK)

C      THIS PROGRAM PRINTS A SELECTED PORTION OF THE SPECTRUM ON THE
C      TELEPRINTER

      COMMON Y
      DIMENSION Y(1200)
      READ (1,5) FNAME
5      FORMAT ('SPECTRUM ID IS ',A6)
      READ (1,6) DEV
6      FORMAT ('DEVICE IS ',A6)
      IF (DEV .EQ. ' ') DEV = '08.7.8'
8      CALL FLDAT (DEV,FNAME,'RD')
7      READ (1,10) ST
10     FORMAT ('STARTING CHANNEL = ',F4.0)
      READ (1,12) XDPTS
12     FORMAT ('NUMBER OF CHANNELS = ',F4.0)
      NDPTS = XDPTS
C      FILE NAME GENERATED BY "FLDAT" IS "RAYXXX" ON DEVICE "SYS@@@"
      READ (1,20) FILE
20     FORMAT ('DATA FILE NAME IS ',A6)
      READ (1,22) DEV
22     FORMAT ('DEVICE IS ',A6)
      CALL RDSK (DEV,ST,NDPTS,FILE,Y)
      WRITE(1,101)(Y(I), I = 1,NDPTS)
101    FORMAT (10F7.0)
      CALL EXIT
      END

```


PROGRAM:
GRPANL

PURPOSE:

To evaluate a group of overlapping peaks with respect to position, area, and shape characteristics.

INPUT:

- 1) Spectrum ID.
- 2) Device on which spectrum is stored.
- 3) Starting and ending channels of the multiplet.
- 4) Number of channels to be used for backgrounds.
- 5) System gain (keV/Ch).
- 6) Energy of some clean nearby reference peak.
- 7) List of peaks (by energy) of the multiplet.
The negative value of the energy must be entered for x-rays.

- 8) Indicate which peaks are to have their energies re-evaluated.
- 9) File name for pertinent peak shape constants.
- 10) Answer whether the width, tailing amplitude, or slope should be redetermined.

CHAIN OF PROGRAMS:

GRPANL→LOAD1→FIT1→PMLR1→PMLA1

↑
—RETURN1→OUT1

COMMENTS:

This routine will allow for eight degrees of freedom and a maximum grouping of 100 data points. An iterative nonlinear least-squares procedure is used in the fitting process. The output consists of peak energies, intensities, and slope parameters, and a listing of the net counts together with the residual counts of the last fitting operation.

C GRPANL (FSIO,FLDAT/I)

C THIS PROGRAM IS A GENERAL PURPOSE PROGRAM FOR FITTING MULTIPLETS.
C IT WILL ACCOMMODATE UP TO 8 FREE PARAMETERS. THE PEAK HEIGHT
C PARAMETERS (AREAS) ARE AUTOMATICALLY FREED. THE OTHERS CAN BE
C SELECTED.

```

COMMON NUMS, NUME, NDPTS, NP, NWRDS, NFLG, NYFLG, NXFLG, FNAME, ST, END,
1 GAIN, REFEN, START, ALPHA, FWHM, ATAIL, BTAIL, R1, R2, C1, C2, AVEBG,
2 AWRDS, EN, SHAPC
DIMENSION NWRDS(6), NFLG(10), NYFLG(10), NXFLG(10), EN(9), SHAPC(5)
1 READ (1,100) FNAME
100 FORMAT ('SPECTRUM ID IS ',A6)
READ (1,120) DEV
120 FORMAT ('DEVICE IS ',A6)
IF (DEV - ' ') 30,31,30
30 CALL FLDAT (DEV,FNAME,'RD')
31 READ (1,101) ST
101 FORMAT ('GROUP START AT ',F4.0)
READ (1,102) END
102 FORMAT ('GROUP ENDS AT ',F4.0)
READ (1,103) NUMS
103 FORMAT ('CHANNELS IN FRONT = ',I2)
READ (1,104) NUME
104 FORMAT ('CHANNELS IN BACK = ',I2)
READ (1,105) GAIN
105 FORMAT ('GAIN = ',F7.5)
READ (1,106) REFEN
106 FORMAT ('REFERENCE PK ENERGY = ',F8.3)
WRITE (1,107)
107 FORMAT ('LIST PK ENERGIES IN THIS GROUP (NEG VALUES FOR XRAYS)')
1 / 'TERMINATE WITH A 0')
I = 0
2 I = I + 1
NXFLG(I) = 0
READ (1,108) EN(I)
108 FORMAT (F9.3)
NYFLG(I) = 1

```

```

      IF (EN(I)) 3,4,2
3     NYFLG(I) = 2
      EN(I) = - EN(I)
      GO TO 2
4     NP = I - 1
      READ (1,109) M
109    FORMAT ('IS THE ENERGY OF ANY OF THESE PEAKS UNCERTAIN? ',A1)
      IF (M - 1632) 10,5,10
5     WRITE (1,110)
110    FORMAT ('WHICH PEAKS? (TYPE 9 FOR ALL)')
1     'TERMINATE WITH 0')
12    READ (1,111) N
111    FORMAT (I1)
      IF (N) 6,10,6
6     IF (N - 9) 7,8,7
7     NXFLG(N) = 1
      GO TO 12
8     DO 9 N = 1,NP
9     NXFLG(N) = 1
10    READ (1,112) CFILE
112    FORMAT ('FILE NAME OF CONSTANTS IS ',A6)
11    CALL IOPEN ('SYS',CFILE)
      READ (4,113) (SHAPC(I), I = 1,5)
113    FORMAT (A6)
      NFLG(1) = 0
      NFLG(2) = 0
      NFLG(3) = 0
      READ (1,115) M
115    FORMAT ('DO YOU WANT TO FREE ANY SHAPE PARAMETERS? ',A1)
      IF (M - 1632) 22,13,22
13    READ (1,116) M
116    FORMAT ('ALPHA? ',A1)
      IF (M - 1632) 15,14,15
14    NFLG(1) = 1
15    READ (1,117) M
117    FORMAT ('TAIL AMPLITUDE? ',A1)
      IF (M - 1632) 17,16,17
16    NFLG(2) = 1
17    READ (1,118) M
118    FORMAT ('TAIL SLOPE? ',A1)
      IF (M - 1632) 22,21,22
21    NFLG(3) = 1
22    AVECH = 0.5 * (ST + END)
      GSQ = GAIN * GAIN
      SG = SHAPC(1) / GSQ + SHAPC(2) * ST / GAIN + .462
      FWHM = SQRT (SG)
      ALPHA = -2.7726 / SG
      AVEN = AVECH * GAIN
      ATAIL = EXP (SHAPC(3) + AVEN * SHAPC(4))
      BTAIL = SHAPC(5) * GAIN
      C1 = 1. / (FWHM * GAIN)
      C1 = 0.3 * C1 * C1 / 1.05
      C2 = -2.44 / FWHM
      CALL CHAIN ('LOAD1')
      END

```

C LOAD1 (FSIO, RDSK, MAXVAL, CHFIT, NCTS, AVE)

C THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "GRPANL".

```
COMMON NUMS, NUME, NDPTS, NP, ITER, NWRDS, NFLG, NYFLG, NXFLG, FNAME,
1  ST, END, GAIN, REFEN, START, ALPHA, FWHM, ATAIL, BTAIL, R1, R2, C1, C2,
2  AVEBG, ASLP, EN, SHAPC, Y, PKPOS, PKHT
DIMENSION NWRDS(5), NFLG(10), NYFLG(10), NXFLG(10), EN(9),
1  SHAPC(5), Y(100), PKPOS(10), PKHT(10)
REFCH = REFEN / GAIN
NXF = 0
IF (REFCH - ST) 22, 21, 21
21 IF (REFCH - END) 30, 30, 22
22 ST1 = REFCH - 10.
NXF = 1
ME = 20.
CALL RDSK ('SYS', ST1, ME, 'RAYXXX', Y)
MS = 1
GO TO 34
30 NDPTS = END - ST + 1.
NDATA = NDPTS + NUMS + NUME
XNUMS = NUMS
ST2 = ST - XNUMS
CALL RDSK ('SYS', ST2, NDATA, 'RAYXXX', Y)
START = ST - 1.
XN = 0.
IF (START - 2048.) 11, 10, 10
10 XN = 2047.
11 NX = START - XN
START = FLOAT (NX) + XN
IF (NXF) 40, 31, 40
31 ST1 = ST2
MS = REFCH - ST2 - 10.
ME = MS + 20
IF (ME - NDATA) 34, 32, 32
32 ME = NDATA
34 CALL MAXVAL (MS, ME, JPK, BIGST, Y)
CALL CHFIT (JPK, REFPK, Y)
XN = 0.
IF (ST1 - 2048.) 36, 35, 35
35 XN = 2047.
36 NX = ST1 - XN
ST1 = FLOAT (NX) + XN
REFCH = ST1 + REFPK - 1.
GAIN = REFEN / REFCH
IF (NXF) 40, 40, 30
40 NE1 = NDPTS + NUMS
CALL NCTS (1, NUMS, NE1, NDATA, NDPTS, BGD, YSUM, AVEBG1, AVEBG2, Y, Y)
AVEBG = 0.5 * (AVEBG1 + AVEBG2)
DO 42 J = 1, NP
PKPOS(J) = EN(J) / GAIN - START
NPK = PKPOS(J) + 0.5
```

```

42      PKHT(J) = Y(NPK)
        ITER = 1
        ASLP = SHAPC(2) / GAIN
        NFLG(4) = 0
        IF (REFEN - 25 ) 44,46,46
44      NFLG(4) = 1
46      CALL CHAIN ('FIT1')
        ENR

```

C FIT1

```

C      THIS SUBPROGRAM PERFORMS A TAYLOR'S EXPANSION ABOUT THE TRIAL
C      VALUES TO OBTAIN A SET OF LINEAR EQUATIONS.  IT IS THE THIRD LINK
C      TO THE PROGRAM "GRPANL".

```

```

        COMMON K, KK, NDPTS, NP, ITER, NWRDS, NFLG1, NFLG2, NFLG3, NXF, MWRDS,
1      NYFLG, NXFLG, FNAME, EXP1, EXP2, GAIN, ALFA, START, ALPHA, FWHM,
2      ATAIL, BTAIL, R1, R2, C1, C2, AVEBG, ASLP, BWRDS, YNET, PKPOS, PKHT, DELY,
3      DEL, RM, VM, ACOEF
        DIMENSION NWRDS(5), MWRDS(6), NYFLG(10), NXFLG(10), BWRDS(14),
1      YNET(100), FKPOS(10), PKHT(10), DELY(100), DEL(10), RM(100), VM(10),
2      ACOEF(100,8)
        IF (ITER - 1) 2,1,2
1      EXP1 = ATAIL
        EXP2 = BTAIL
        KK = NFLG1 + NFLG2 + NFLG3
2      SG = -2.7726 / ALPHA
        DO 30 I = 1, NDPTS
            XI = I
            FX = 0.0
            DALFA = 0.0
            DBETA = 0.0
            DGAMA = 0.0
            WT = 1.0 / SQRT (YNET(I) + AVEBG)
            K = KK
            DO 20 J = 1, NP
                DLTAX = XI - PKPOS(J)
                DX = DLTAX
                DLX = DLTAX
                DLXSQ = DLTAX * DLTAX
                EXPN1 = 0.0
                PHT = PKHT(J)
                ALFA = -2.7726 / (ASLP * PKPOS(J) + SG)
                EXPNT = ALFA * DLXSQ
                FXX = 0.
                EXPN3 = 0.
                IF (EXPNT + 18.) 4,3,3
            3      EXPN1 = EXP (EXPNT)
                FXX = EXPN1
                EXPN3 = EXP (0.4 * EXPNT)
            4      IF (NFLG1) 6,6,5
            5      DALFA = DALFA + PHT * DLXSQ * EXPN1
            6      IF (DLTAX) 7,18,18
            7      EXPN = EXP2 * DLTAX

```

```

      IF (EXPN + 10.) 18,8,8
8     EXPN2 = EXP (EXPN)
      CB = EXPN2 * (1.0 - EXPN3)
      FXX = FXX + EXP1 * CB
      IF (NFLG3) 10,10,9
9     DGAMA = DGAMA + DLTAX * EXP1 * PHT * CB
10    IF (NFLG2) 14,14,11
11    DBETA = DBETA + CB * PHT
14    DLX = -DLX
18    IF (NYFLG(J) -1) 13,13,12
12    FXX = FXX + C1 * (1.05 - EXPN3) * EXP (C2 * DLX)
13    IF (NYFLG(J)) 16,16,15
15    K = K + 1
      ACOEF(I,K) = EXPN1 * WT
16    IF (NXFLG(J)) 19,19,17
17    K = K + 1
      ACOEF(I,K) = -2. *ALFA *PHT* DX *EXPN1 *WT
19    FX = FX + FXX * PHT
20    CONTINUE
      L = 0
      IF (NFLG1) 22,22,21
21    L = L + 1
      ACOEF(I,L) = DALFA * WT
22    IF (NFLG2) 24,24,23
23    L = L + 1
      ACOEF(I,L) = DBETA * WT
24    IF (NFLG3) 26,26,25
25    L = L + 1
      ACOEF(I,L) = DGAMA * WT
26    DELY(I) = (YNET(I) - FX ) * WT
30    CONTINUE
      CALL CHAIN ('PMLR1')
      END

```

C PMLR1 (MLR/O)

C THIS SUBPROGRAM IS THE FOURTH LINK TO THE PROGRAM "GRPANL".

```

      COMMON K, KK, NDPTS, NP, ITER, LFLG, NWRDS, AWRDS, Y, BWRDS, DELY,
1     DEL, RM, VM, ACOEF, WM
      DIMENSION NWRDS(34), AWRDS(30), Y(100), BWRDS(20), DELY(100),
1     DEL(10), RM(100), VM(10), ACOEF(100,8), WM(10)
      LFLG = 1
      CALL OOPEN ('DSK1', 'AMTRX')
      NM = 100 * K
      WRITE (4,100) (ACOEF(I), I = 1, NM)
100    FORMAT (10A6)
      CALL OCLOSE
      CALL MLR (100, NDPTS, K, DELY, DEL, ACOEF, RM, WM, VM, LFLG)
      CALL CHAIN ('PMLA1')
      END

```

C PMLA1 (MLRA/I)

C THIS SUBPROGRAM IS THE FIFTH LINK TO THE PROGRAM "GRPANL".

```
COMMON K, KK, NDPTS, NP, ITER, LFLG, NWRDS, AWRDS, Y, BWRDS,
1 DELY, DEL, RM, VM, BM, WM
DIMENSION NWRDS(34), AWRDS(30), Y(100), BWRDS(20), DELY(100),
1 DEL(10), RM(100), BM(800), WM(10), VM(10)
IF (LFLG - 2) 2, 4, 2
2 CALL MLRA (100, NDPTS, K, DELY, DEL, BM, RM, WM, VM, LFLG)
4 CALL CHAIN ('RETRN1')
END
```

C RETRN1 (MLRE)

C THIS SUBPROGRAM IS THE SIXTH LINK TO THE PROGRAM "GRPANL".

```
COMMON K, KK, NDPTS, NP, ITER, NERR, NWRDS, NFLG1, NFLG2, NFLG3, MWRDS,
1 NYFLG, NXFLG, FNAME, EXP1, EXP2, GAIN, ALFA, START, ALPHA, AWRDS, PKPOS,
2 PKHT, DELY, DEL, RM, DM, BM, WM
DIMENSION NWRDS(4), MWRDS(7), NYFLG(10), NXFLG(10), AWRDS(123),
1 PKPOS(10), PKHT(10), DELY(100), DEL(10), RM(100), BM(800),
2 WM(10), DM(10)
GO TO (1, 2, 3) NERR
2 WRITE (1, 101)
101 FORMAT ('MATRIX IS SINGULAR')
GO TO 10
3 WRITE (1, 102)
102 FORMAT ('SUM OF SQUARES OF R > THAN THOSE OF Y')
10 CALL EXIT
C INCREMENT THE TRIAL VALUES AND CHECK FOR COMPLETION OF ITERATIVE
C PROCESS.
1 L = 0
LFLG = 0
IF (NFLG1) 43, 43, 41
41 L = L + 1
ALPHA = ALPHA + DEL(L)
IF (ABS (DEL(L)) + .01*ALFA) 43, 43, 42
42 LFLG = 1
43 IF (NFLG2) 46, 46, 44
44 L = L + 1
EXP1 = EXP1 + DEL(L)
IF (ABS (DEL(L)) - .05 * EXP1) 46, 46, 45
45 LFLG = 1
46 IF (NFLG3) 49, 49, 47
47 L = L + 1
EXP2 = EXP2 + DEL(L)
IF (ABS (DEL(L)) - .1 * EXP2) 49, 49, 48
48 LFLG = 1
49 K = L
DO 60 J = 1, NP
IF (NYFLG(J)) 52, 52, 50
```

```

50      K = K + 1
      PKHT(J) = PKHT(J) + DEL(K)
      IF (ABS (DEL(K)) - .002 * PKHT(J)) 52,52,51
51      LFLG = 1
52      IF (NXFLG(J)) 60,60,53
53      K = K + 1
      PKPOS(J) = PKPOS(J) + DEL(K)
      IF (ABS (DEL(K)) - .02 * PKPOS(J)) 60,60,54
54      LFLG = 1
60      CONTINUE
      IF(LFLG) 70,79,70
70      ITER = ITER + 1
      IF (ITER - 10)78,78,77
78      CALL CHAIN ('FIT1')
77      WRITE (1,105)
105     FORMAT ('MAXIMUM ITERATIONS EXCEEDED.')
79      CALL MLRE (DM,100,K,BM,WM)
      CALL CHAIN ('OUT1')
      END

```

C OUT1

C THIS SUBPROGRAM IS THE SEVENTH LINK TO THE PROGRAM "GRPANL".

```

      COMMON K, KK, NDPTS, NP, NWRDS, NYFLG, NXFLG, FNAME, EXP1, EXP2,
1      GAIN, ALFA, START, ALPHA, FWHM, ATAIL, BTAIL, R1, R2, C1, C2,
2      AVEBG, ASLP, BWRDS, YNET, PKPOS, PKHT, RES, DEL, RM, DM
      DIMENSION NWRDS(16), NYFLG(10), NXFLG(10), BWRDS(14),
1      YNET(100), PKPOS(10), PKHT(10), RES(100), DEL(10), RM(100),
2      DM(10)
      NOUT = 3
      SUM = 0.0
      DO 4 I = 1, NDPTS
      RES(I) = RM(I) * SQRT (YNET(I) + AVEBG)
4      SUM = SUM + RM(I) **2
      DOF = NDPTS - KK - NP
      QFIT = SUM / DOF
      SG = -2.7726 / ALPHA + ASLP * PKPOS(1)
      FWHM = SQRT (SG)
      BTAIL = EXP2 / GAIN
      CA = -1.109 / SG
      CE = 2.44 / FWHM
      AR = 0.0
      AX = 0.0
      XX = 0.25
      CT = 0.5
      GSQ = GAIN * GAIN
      DO 18 I = 1,100
      XSQ = XX * XX * CA
      DELA = EXP (-EXP2 * XX + XSQ * CT)
      AR = AR + DELA * CT
      DELX = EXP (-CE * XX + XSQ)

```

```

AX = AX + DELX * CT
XX = I
CT = 1.0
IF (DELA * 200. - AR) 19,18,18
19 IF (DELX * 200. - AX) 20,18,18
18 CONTINUE
20 R1 = (SQRT (1.133 * SG)) + EXP1 * (1./EXP2 - AR)
R2 = R1 + (.6 / (GSD * SG)) * (1./CE - AX / 1.05)
R2 = R2 / R1
FWHM = FWHM * GAIN
WRITE (NOUT,100) START,FNAME,QFIT,FWHM,EXP1,BTAIL
100 FORMAT ('ANALYSIS RESULTS OF PEAK GROUP STARTING AT ',F8.0/
1 'SPECTRUM ID IS ',A6,' QFIT =',F5.1,' FWHM =',F8.5/23X,
2 'EXP1 =',F6.4,' BTAIL =',F8.4/3X,
3 'CHANNEL ENERGY COUNTS PCTERR')
K = KK
DO 6 J = 1,NP
K = K + 1
CTS = R1 * PKHT(J)
IF (NYFLG(J) - 2) 8,7,8
7 CTS = CTS * R2
8 ERR = (100 * SQRT (QFIT * DM(K))) / PKHT(J)
CHAN = PKPOS(J) + START
ENRG = CHAN * GAIN
WRITE (NOUT,102) J,CHAN,ENRG,CTS,ERR
102 FORMAT (I2,F9.3,F10.3,F9.0,F6.1)
IF (NXFLG(J)) 9,6,9
9 K = K + 1
6 CONTINUE
WRITE (NOUT,103) (I,YNET(I),RES(I),RM(I), I = 1,NDPTS)
103 FORMAT (/7X,'YNET RESIDUALS STD. DEV. ',/(I3,F10.2,2F9.2))
CALL EXIT
END

```


PROGRAM:
FILES

PURPOSE:

To create or edit disk files containing data or constants used by other programs.

INPUT:

- 1) File name.
- 2) Name of device on which the file is to be found and/or stored.
- 3) Answer teletyped questions regarding file length, listing of values, changes to be made, etc.

```
C      FILES (/I/O)

C      THIS IS A GENERAL PURPOSE ROUTINE FOR THE PREPARATION OR EDITING
C      OF DISK FILES PREPARED IN FLOATING POINT FORMAT

      DIMENSION CON(1000)
      READ (1,100) FNAME
100     FORMAT ('FILE NAME IS ',A6)
      READ (1,101) DEV
101     FORMAT ('DEVICE NAME IS ',A6)
      READ (1,102) NM
102     FORMAT ('IS THIS A NEW FILE? ',A1)
      IF (NM - 1632) 2,10,2
C      FETCH EXISTING FILE
2       READ (1,117) LENGTH
      CALL IOPEN (DEV,FNAME)
      READ (4,104) (CON(I), I = 1,LENGTH)
104     FORMAT (A6)
      READ (1,106) NUM
106     FORMAT ('HOW MANY VALUES DO YOU WANT LISTED? ',I3)
      READ (1,107) NM
107     FORMAT ('STARTING AT ',I4)
      NUM = NUM + NM
      WRITE (1,108) (I,CON(I), I = NM,NUM)
108     FORMAT (I3,E13.5)
      READ (1,110) NM
110     FORMAT ('DO YOU WANT TO CHANGE ANY VALUES? ',A1)
      IF (NM - 1632) 30,6,30
C      CHANGE VALUES IN EXISTING FILE
6       READ (1,112) N
112     FORMAT ('WHICH VALUE? ',I4)
      IF (N) 7,8,8
C      A NEGATIVE VALUE INDICATES ENTRY POSITION IN THE FILE FOR A
C      SEQUENCE OF CHANGES
7       N = -N -1
      GO TO 13
8       READ (1,114) AA
```

```

114     FORMAT ('NEW VALUE = ',E12.5)
        CON(N) = AA
        READ (1,116) NM
116     FORMAT ('ANY MORE CHANGES? ',A1)
        IF (NM - 1632) 20,6,20
C       PREPARE FOR FORMATION OF A NEW FILE
10      READ (1,117) LENGTH
117     FORMAT ('FILE LENGTH = ', I4)
C       ZERO OUT THE ARRAY BEFORE ENTERING VALUES.
        DO 11 I = 1,LENGTH
11      CON(I) = 0.0
        N = 0
13      READ (1,121) NUM
121     FORMAT ('HOW MANY VALUES WILL YOU ENTER? ',I3)
        NUM = NUM + N
        WRITE (1,120)
120     FORMAT ('TYPE VALUES IN SEQUENCE. ')
17      N = N + 1
        READ (1,122) CON(N)
122     FORMAT (E11.5)
        IF (N - NUM) 12,20,12
20      CALL OOPEN (DEV,FNAME)
        WRITE (4,104) (CON(I), I = 1,LENGTH)
        CALL OCLOSE
30      WRITE (1,124)
124     FORMAT ('ALL DONE')
        CALL EXIT
        END

```

PROGRAM:
PKANAL

PURPOSE:

To evaluate a single peak and characterize the parameters which describe it.

INPUT:

- 1) Spectrum ID.
- 2) Device on which spectrum is stored.
- 3) Starting and ending channel position at the peak.
- 4) Number of channels to be used for backgrounds.
- 5) Option Number
 - 0 = simple integration of the peak and removal of background. NO FITTING.
 - 1 = fitting with only position and height parameters free.
 - 2 = width parameter also free.
 - 3 = tailing amplitude parameter also free.
 - 4 = tailing slope parameter also free.
 (This option will not always result in convergence.)

- 6) FWHM of the peak (in keV). If this parameter is to be determined, a best estimate may be entered.
- 7) Amplitude of the peak tailing (fraction of peak height). If this parameter is to be determined, a best estimate or zero may be entered.
- 8) Slope of the peak tailing. If zero is entered, a default value of 3.0 will be used.

CHAIN OF PROGRAMS USED:

COMMENTS:

The peak shape characteristics for two peaks (as determined by this program) may be used in SANAL to compute generalized shape parameters.
Maximum group size is 50 channels.

C PKANAL (FSIO, FLDAT, RDSK, MAXVAL, CHFIT, NCTS, AVE)

```

COMMON NS1, NE1, NDPTS, NWRDS, IFLG, NFLG, NXFLG, NYFLG, AVEBG,
1  PTERR, START, AWRDS, ALPHA, ATAIL, BTAIL, Y, PKPOS, PKHT
DIMENSION NWRDS(6), NFLG(10), NXFLG(10), NYFLG(10), AWRDS(24),
1  Y(50), PKPOS(5), PKHT(5)
READ (1, 110) FNAME
FORMAT ('SPECTRUM ID IS ', A6)
READ (1, 120) DEV
FORMAT ('DEVICE IS ', A6)
IF (DEV = ' ') 3, 5, 3
3  CALL FLDAT (DEV, FNAME, 'RD')
5  READ (1, 100) ST
FORMAT ('PK STARTS AT ', F4.0)
READ (1, 101) EN
FORMAT ('PK ENDS AT ', F4.0)
READ (1, 102) NUMS
FORMAT ('CHANNELS IN FRONT = ', I2)
READ (1, 109) NUME
FORMAT ('CHANNELS IN BACK = ', I2)
READ (1, 107) IFLG
FORMAT ('OPTION NO. = ', I1)
IF (IFLG) 1, 11, 1
1  READ (1, 103) GAIN
FORMAT ('GAIN = ', F7.5)
READ (1, 104) FWHM
FORMAT ('FWHM (IN KEV) = ', F7.5)
READ (1, 105) ATAIL
  
```

```

105     FORMAT ('ATAIL = ',F7.5)
      READ (1,106) BTAIL
106     FORMAT ('BTAIL = ',F7.5)
      IF (FWHM) 6,4,6
      4     FWHM = 0.6
      6     ALPHA = -2.7707 * GAIN * GAIN / (FWHM * FWHM)
      IF (BTAIL) 10,8,10
      8     BTAIL = 3.
      10    BTAIL = BTAIL * GAIN
      11    IF (NUMS) 15,14,15
      14    NUMS = 5
      15    IF (NUME) 17,16,17
      16    NUME = 5
      17    XNUMS = NUMS
      NWRDS(2) = 0
      AWRDS(1) = GAIN
      ST1 = ST - XNUMS + 1.
      START = ST1 - 1.
      NS1 = ST - START
      NE1 = EN - START
      NDPTS = NE1 + NUME
      CALL RDSK ('SYS',ST1,NDPTS,'RAYXXX',Y)
      CALL NCTS (1,NS1,NE1,NDPTS,NDPTS,BGD,YSUM,AVBG1,AVBG2,Y,Y)
      CALL MAXVAL (1,NDPTS,JPK,BIGST,Y)
      CALL CHFIT (JPK,REFPK,Y)
      PKPOS(1) = REFPK
      PKHT(1) = BIGST
      START = START + XNUMS
      AVEBG = AVBG1 + DBG * 0.5
      CHANS = KK - K + 1
      XPTS = NDPTS
      R = CHANS / XPTS
      SUM = YSUM + AVEBG * XPTS
      PCTERR = 100. / YSUM * SORT (SUM + R*R*BGD)
      NFLG(1) = 0
      NFLG(2) = 0
      NFLG(3) = 0
      IF (IFLG - 1) 24,29,25
24     WRITE (3,111) REFPK,YSUM,PCTERR
111    FORMAT (' PEAK POSITION = ',F8.3,' AREA = ',F10.1,
1      ' COUNTS +/- 'F4.1' %',//5X,' NET COUNTS')
      WRITE (3,112) (Y(I), I = 1,NDPTS)
112    FORMAT (10F7.0)
      CALL EXIT
      25    IF (IFLG - 3) 28,27,26
      26    NFLG(3) = 1
      27    NFLG(2) = 1
      28    NFLG(1) = 1
      29    NYFLG(1) = 1
      NXFLG(1) = 1
      NFLG(4) = 1
      NFLG(7) = 1
      CALL CHAIN ('FIT')
      END

```

C FIT

C THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "PKANAL".
C IT PERFORMS A TAYLOR'S EXPANSION ABOUT THE TRIAL VALUES TO
C OBTAIN A SET OF LINEAR EQUATIONS.

```
COMMON NS1, NE1, NDPTS, KK, NWRDS, NFLG1, NFLG2, NFLG3, ITER, K, M, NP,  
1 MWRDS, NXFLG, NYFLG, AVEBG, AWRDS, ALFA, EXP1, EXP2, ALPHA, ATAIL,  
2 BTAIL, YNET, PKPOS, PKHT, DELY, BWRDS, ACOEF  
DIMENSION NWRDS(6), MWRDS(3), NXFLG(10), NYFLG(10), AWRDS(23),  
1 YNET(50), PKPOS(5), PKHT(5), DELY(50), BWRDS(55), ACOEF(50, 5)  
IF (ITER - 1) 2, 1, 2  
1 ALFA = ALPHA  
EXP1 = ATAIL  
EXP2 = BTAIL  
KK = NFLG1 + NFLG2 + NFLG3  
2 DO 30 I = 1, NDPTS  
XI = I  
DY = YNET(I)  
DALFA = 0.0  
DBETA = 0.0  
DGAMA = 0.0  
WT = 1.0 / SQRT (YNET(I) + AVEBG)  
K = KK  
DO 20 J = 1, NP  
DLTAX = XI - PKPOS(J)  
DLXSQ = DLTAX * DLTAX  
EXPNT = 0.0  
PHT = PKHT(J)  
EXPNT = ALFA * DLXSQ  
IF (EXPNT + 14.) 4, 3, 3  
3 EXPN1 = EXP (EXPNT)  
DY = DY - EXPN1 * PHT  
4 IF (NFLG1) 6, 6, 5  
5 DALFA = DALFA + PHT * DLXSQ * EXPN1  
6 IF (DLTAX) 7, 14, 14  
7 EXPN = EXP2 * DLTAX  
IF (EXPN + 10.) 14, 8, 8  
8 EXPN2 = EXP (EXPN)  
EXPNT = 0.0  
IF (EXPN1) 12, 13, 12  
12 EXPN3 = EXP (0.4 * EXPNT)  
13 CA = EXP1  
CB = EXPN2 * (1. - EXPN3)  
CY = CA * PHT  
DY = DY - CY * CB  
IF (NFLG2) 10, 10, 9  
9 DBETA = DBETA + CB * PHT  
10 IF (NFLG3) 14, 14, 11  
11 DGAMA = DGAMA + DLTAX * CY * CB  
14 IF (NYFLG(J)) 16, 16, 15  
15 K = K + 1  
ACOE(I, K) = EXPN1 * WT  
16 IF (NXFLG(J)) 20, 20, 17  
17 K = K + 1  
ACOE(I, K) = -2. *ALFA *PHT* DLTAX *EXPN1 *WT  
18
```

```

23     L = L + 1
        ACDEF(J,L) = DBETA * WT
24     IF (NFLG3) 26,26,25
25     L = L + 1
        ACDEF(1,L) = DGAMA * WT
26     DELY(I) = DY * WT
30     CONTINUE
        WRITE (1,200) PKPOS(1),PKHT(1),ALFA,EXP1,EXP2
200    FORMAT(SE11,3)
        CALL CHAIN ('PMLR')
        END

```

C PMLR (MLR/O)

C THIS SUBPROGRAM IS THE THIRD LINK TO THE PROGRAM "PKANAL".

```

        COMMON NFLG,AWRDS,DELY,DEL,RM,ACDEF,WM,VM
        DIMENSION NFLG(40),AWRDS(90),DELY(50),DEL(5),RM(50),ACDEF(50,5)
1       ,WM(5),VM(5)
        K = NFLG(15)
        LFLG = 1
        NDPTS = NFLG(3)
        CALL OOPEN ('DSK1','AMTRX')
        NM = 50 * K
100      WRITE (4,100) (ACDEF(I), I = 1,NM)
        FORMAT (10A6)
        CALL OCLOSE
        CALL MLR (50,NDPTS,K,DELY,DEL,ACDEF,RM,WM,VM,LFLG)
        NFLG(16) = LFLG
        CALL CHAIN ('PMLRA')
        END

```

C PMLRA (MLRA/I)

C THIS SUBPROGRAM IS THE FOURTH LINK TO THE PROGRAM "PKANAL".

```

        COMMON NFLG,AWRDS,DELY,DEL,RM,BM,WM,VM
        DIMENSION NFLG(40),AWRDS(90),DELY(50),DEL(5),RM(50),
1       BM(250),WM(5),VM(5)
        NMLR = 50
        IF (NFLG(16) - 2)2,4,2
2       K = NFLG(15)
        NDPTS = NFLG(3)
        LFLG = NFLG(16)
        CALL MLRA (NMLR,NDPTS,K,DELY,DEL,BM,RM,WM,VM,LFLG)
        NFLG(16) = LFLG
4       CALL CHAIN ('RETRN')
        END

```

C RETRN

C THIS SUBPROGRAM IS THE FIFTH LINK TO THE PROGRAM "PKANAL".
C IT INCREMENTS THE TRIAL VALUES AND CHECKS FOR COMPLETION OF THE
C ITERATIVE PROCESS

```
COMMON NWRDS, NFLG1, NFLG2, NFLG3, ITER, KL, NERR, NP, MWRDS, NXFLG,
1 NYFLG, AWRDS, ALFA, EXP1, EXP2, BWRDS, PKPOS, PKHT, CWRDS, DEL, RM
DIMENSION NWRDS(10), MWRDS(3), NXFLG(10), NYFLG(10), AWRDS(24),
1 BWRDS(53), PKPOS(5), PKHT(5), CWRDS(50), DEL(5), RM(50)
GO TO (1,2,3) NERR
2 WRITE (1,101)
101 FORMAT ('MATRIX IS SINGULAR')
GO TO 10
3 WRITE (1,102)
102 FORMAT ('SUM OF SQUARES OF R > THAN THOSE OF Y')
10 CALL EXIT
1 L = 0
LFLG = 0
IF (NFLG1) 43,43,41
41 L = L + 1
ALFA = ALFA + DEL(L)
IF (ABS (DEL(L)) + .01*ALFA) 43,43,42
42 LFLG = 1
43 IF (NFLG2) 46,46,44
44 L = L + 1
EXP1 = EXP1 + DEL(L)
IF (ABS (DEL(L)) - .05 * EXP1) 46,46,45
45 LFLG = 1
46 IF (NFLG3) 49,49,47
47 L = L + 1
EXP2 = EXP2 + DEL(L)
IF (ABS (DEL(L)) - .1 * EXP2) 49,49,48
48 LFLG = 1
49 K = L
DO 60 J = 1, NP
IF (NYFLG(J)) 52,52,50
50 K = K + 1
PKHT(J) = PKHT(J) + DEL(K)
IF (ABS (DEL(K)) - .002 * PKHT(J)) 52,52,51
51 LFLG = 1
52 IF (NXFLG(J)) 60,60,53
53 K = K + 1
PKPOS(J) = PKPOS(J) + DEL(K)
IF (ABS (DEL(K)) - .02 * PKPOS(J)) 60,60,54
54 LFLG = 1
60 CONTINUE
IF (LFLG) 70,79,70
70 ITER = ITER + 1
IF (ITER - 10) 78,78,77
77 WRITE (1,105)
105 FORMAT ('MAXIMUM ITERATIONS EXCEEDED.')
78 CALL CHAIN ('FIT')
79 CALL CHAIN ('OUTO')
END
```

C OUTO

C THIS SUBPROGRAM IS THE SIXTH LINK TO THE PROGRAM "PKANAL".

```
COMMON NS1, NE1, NDPTS, KK, IJ, NWRDS, AVEBG, PCTERR, START, GAIN,
1 EN, CH, FWHM, ATAIL, AWRDS, ALFA, EXP1, EXP2, BWRDS, YNET, PKPOS,
2 PKHT, DELY, DEL, RM
DIMENSION NWRDS(35), EN(2), CH(2), FWHM(2), ATAIL(2), AWRDS(12),
1 BWRDS(3), YNET(50), PKPOS(5), PKHT(5), DELY(50), DEL(5), RM(50)
CA = 0.4 * ALFA
AR = 0.0
XX = 0.25
CT = 0.5
DO 80 I = 1, NDPTS
XSQ = XX * XX * CA
DELA = EXP (-EXP2 * XX + XSQ * CT)
AR = AR + DELA * CT
XX = I
CT = 1.0
IF (DELA * 200. - AR) 81, 80, 80
80 CONTINUE
81 RALFA = SQRT (-1. / ALFA)
ARATO = 1.7724 * RALFA + EXP1 * (1. / EXP2 - AR)
AREA = PKHT(1) * ARATO
SUM = 0.0
DO 84 I = 1, NDPTS
DELY(I) = RM(I) * SQRT (YNET(I) + AVEBG)
84 SUM = SUM + RM(I) * RM(I)
DOF = NDPTS - KK - 3
QFIT = SUM / DOF
RA = 1.6645 * RALFA
F12M = GAIN * RA
BTAIL = EXP2 / GAIN
WRITE (3, 106) START, PKPOS(1), PKHT(1), AREA, PCTERR, QFIT, F12M, EXP1,
1 BTAIL
106 FORMAT (5X, 'ANALYSIS RESULTS OF PEAK STARTING AT ', F8.0 /
1 ' PKPOS PKHT NETCTS PCTERR QFIT FWHM ATAIL BTAIL'
2 ' / F8.3, F8.0, F9.0, F8.2, F5.1, F8.4, F9.5, F8.4 /')
WRITE (3, 107) (YNET(I), DELY(I), RM(I), I = 1, NDPTS)
107 FORMAT (5X, 'YNET RESIDUALS STD. DEV. ' / (F10.1, 2F9.2))
IF (IJ - 1) 99, 85, 85
85 FWHM(IJ) = RA
ATAIL(IJ) = EXP1
IF (IJ - 1) 89, 89, 86
86 IJ = IJ - 1
IF (NWRDS(10) - 4) 88, 87, 88
87 NWRDS(10) = 3
88 CALL CHAIN ('NTCTS')
89 GAIN = (EN(2) - EN(1)) / (CH(2) - CH(1))
DO 90 J = 1, 2
90 FWHM(J) = GAIN * FWHM(J)
BWRDS(3) = EXP2
CALL CHAIN ('SHAP')
99 CALL EXIT
END
```


PROGRAM:
SANAL

PURPOSE:

To compute generalized peak shape parameters using the shape characteristics of two clean and statistically good peaks. These peaks should be near the extremes of the energy region of interest.

INPUT:

- GAIN (keV/Ch)
- For each peak:
 - 1) Peak energy (in keV).
 - 2) FWHM (in keV).
 - 3) The tailing amplitude.

- 4) The tailing slope. (This value will be the same for both peaks.)

CHAIN OF PROGRAMS USED:

SANAL→SHAP

COMMENTS:

The program prints out values which may be used by the generalized analysis programs. These values may be stored in a disk file using the FILES routine, or, if such a file already exists, the SHAP program permits an update of that file to be made.

C SANAL

C THIS PROGRAM IS USED TO GENERATE A FILE OF PEAK SHAPE PARAMETERS.

```
COMMON NWRDS, AWRDS, GAIN, EN, CH, FWHM, ATAIL, SHAPC, BWRDS, BTAIL
DIMENSION NWRDS(40), AWRDS(3), EN(2), CH(2), FWHM(2), ATAIL(2),
1 SHAPC(5), BWRDS(12)
1 READ (1,100) GAIN
100 FORMAT ('GAIN = ',F5.3)
I = 0
2 I = I + 1
READ (1,101) EN(I)
101 FORMAT ('PK ENERGY = ',F8.3)
READ (1,102) FWHM(I)
102 FORMAT ('FWHM = ',F5.3)
READ (1,103) ATAIL(I)
103 FORMAT ('ATAIL = ',F5.3)
READ (1,104) BTAIL
104 FORMAT ('BTAIL = ',F5.3)
IF (I - 1) 3,2,3
3 READ (1,105) NREDO
105 FORMAT ('DATA OK? Y OR N ',A1)
IF (NREDO - 1632) 1,4,1
4 BTAIL = BTAIL * GAIN
NWRDS(10) = 0
CALL CHAIN ('SHAP')
END
```

```

C      SHAP (/I/O)

C      THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "SANAL".

      COMMON NWRDS, AWRDS, GAIN, EN, CH, FWHM, ATAIL, SHAPC, BWRDS, BTAIL
      DIMENSION NWRDS(40), AWRDS(3), EN(2), CH(2), FWHM(2), ATAIL(2),
1  SHAPC(5), BWRDS(12), WP(2), TP(2), CON(25)
      GSO = GAIN * GAIN
      DO 10 J = 1,2
      WP(J) = FWHM(J) * FWHM(J) - 0.462 * GSO
      TP(J) = ALOG (ATAIL(J))
10     CONTINUE
C      CALCULATE THE AREA-TO-PEAK HEIGHT RATIO FOR A 100 KEV PEAK.
      DEN = 1. / (EN(1) - EN(2))
      SA = (WP(1) - WP(2)) * DEN
      SHAPC(1) = WP(1) - EN(1) * SA
      SHAPC(2) = SA
      SA = (TP(1) - TP(2)) * DEN
      SHAPC(3) = TP(1) - EN(1) * SA
      SHAPC(4) = SA
      SG = ( 100 * SHAPC(2) + SHAPC(1)) / GSO + .462
      CA = -1.109 / SG
      CE = 2.44 / SQRT (SG)
      AR = 0.0
      AX = 0.0
      XX = 0.25
      CT = 0.5
      DO 20 I = 1,100
      XSO = XX * XX * CA
      DELA = EXP (-BTAIL * XX + XSO * CT)
      AR = AR + DELA * CT
      DELX = EXP (-CE * XX + XSO)
      AX = AX + DELX * CT
      XX = I
      CT = 1.0
      IF (DELA * 200. - AR) 21,20,20
21     IF (DELX * 200. - AX) 22,20,20
20     CONTINUE
22     EXP1 = EXP (SHAPC(3) + 100. * SHAPC(4))
      SHAPC(5) = BTAIL / GAIN
      C100 = (SQRT (1.133 * SG)) + EXP1 * (1. / BTAIL - AR)
C      CALCULATE THE X-RAY TO GAMMA RAY AREA RATIO AT 100 KEV.
      XC100 = C100 + (.6 / (GSO * SG)) * (1.0 / CE - AX / 1.05)
      XC100 = XC100 / C100
      WRITE (1,110) SHAPC(1), SHAPC(3), SHAPC(5), SHAPC(2), SHAPC(4),
1  C100, XC100
110    FORMAT ('SHAPE PARAMETERS ARE :', // 'S(1) =', F6.3, 6X, 'S(3) =',
1  E10.3, 2X, 'S(5) =', F6.3, / 'S(2) =', E10.3, 2X,
2  'S(4) =', E10.3, 2X, 'C100 =', E10.4, / 'XC100 =', F6.4 )
      IF (NWRDS(5)) 99,99,30
30     READ (1,112) M
112    FORMAT ('DO YOU WANT TO STORE ANY VALUES? Y OR N. 'A1)
      IF (M - 1632) 99,31,99
31     READ (1,120) FNAME

```

```

120     FORMAT ('FILE NAME IS ',A6)
      READ (1,121) DEV
121     FORMAT ('DEVICE NAME IS ',A6)
      CALL IOPEN (DEV,FNAME)
      READ (4,114) (CON(I), I = 1,25)
114     FORMAT (A6)
      READ (1,113) N
113     FORMAT ('WHICH VALUE? (TYPE 9 FOR ALL) ',I1)
      IF (N - 9) 32,34,34
32      CON(N) = SHAPC(N)
      READ (1,115) M
115     FORMAT ('ANY MORE CHANGES? ',A1)
      IF (M - 1632) 36,33,36
33      READ (1,116) N
116     FORMAT ('WHICH VALUE? ',I1)
      GO TO 32
34      DO 35 I = 1,7
35      CON(I) = SHAPC(I)
36      CALL OOPEN (DEV,FNAME)
      WRITE (4,114) (CON(I), I = 1,25)
      CALL OCLOSE
99      CALL EXIT
      END

```

PROGRAM:
SHAPE

PURPOSE:

Same as SANAL. The shape parameters are calculated using the two peaks specified. A previous determination or estimate of the shape parameters must already exist in a disk file.

4 = 3 plus redetermination of tailing slope parameters.

5) Name of disk file containing correct shape parameters.

6) Device on which this file is stored.

INPUT:

- 1) Spectrum ID.
- 2) Device on which spectrum is stored.
- 3) Energies and channel positions of the peaks.
- 4) Option Number:
 - 2 = Redetermination of peak width shape parameters.
 - 3 = 2 plus redetermination of tailing amplitude parameters.

CHAIN OF PROGRAMS USED:

COMMENTS:
See SANAL.

C SHAPE (FSIO,FLDAT/I)

```

COMMON NS1, NE1, NDPTS, KK, IJ, NWRDS, IFLG, NFLG, NXFLG, NYFLG, AVEBG,
1  PCTERR, START, GAIN, EN, CH, FWHM, ATAIL, SHAPC, FNAME, BWRDS, ALFA, EXP1,
2  EXP2, Y, PKPOS, PKHT, DUMMY
DIMENSION NWRDS(4), NFLG(10), NXFLG(10), NYFLG(10), EN(2), CH(2),
1  FWHM(2), ATAIL(2), SHAPC(7), BWRDS(7), Y(50), PKPOS(5),
2  PKHT(5), DUMMY(140)
READ (1,100) FNAME
FORMAT ('SPECTRUM ID IS ',A6)
READ (1,120) DEV
FORMAT ('DEVICE IS ',A6)
IF (DEV = ' ') 2,1,2
2  CALL FLDAT (DEV,FNAME,'RD')
1  READ (1,101) EN(1)
101  FORMAT ('PK1 ENERGY = ',F8.3)
READ (1,102) CH(1)
102  FORMAT ('CHAN POSITION = ',F8.3)
READ (1,103) EN(2)
103  FORMAT ('PK2 ENERGY = ',F8.3)
READ (1,104) CH(2)
104  FORMAT ('CHAN POSITION = ',F8.3)
READ (1,107) IFLG
107  FORMAT ('OPTION NO. = ',I1)
READ (1,105) FNAME
105  FORMAT ('FILE NAME OF SHAPE CONSTANTS IS ',A6)
READ (1,110) DEV
110  FORMAT ('DEVICE NAME IS ',A6)
GAIN = (EN(2) - EN(1))/(CH(2) - CH(1))
CALL IOPEN (DEV,FNAME)
READ (4,106) (SHAPC(I), I = 1,7)
106  FORMAT (A6)
IJ = 2
WRITE (1,108) SHAPC(1), SHAPC(3), SHAPC(5), SHAPC(2), SHAPC(4),
1  SHAPC(6), SHAPC(7)
  
```

```

108  FORMAT ('/CURRENT SHAPE PARAMETERS ARE :'/S(1) =',F6.3,
1    6X,S(3) =',E10.3,2X,S(5) =',F6.3,'/S(2) =',E10.3,2X,
2    S(4) =',E10.3,2X,C100 =',E10.4,' XC100 =',F6.4)
    CALL CHAIN ('NTCTS')
    END

```

C NTCTS (FSIO,RDSK,MAXVAL,CHFIT,NCTS,AVE)

```

    COMMON NS1,NE1,NDPTS,FK,IJ,NWRDS,IFLG,NFLG,NXFLG,NYFLG,AVEBG,
1    PCTERR,START,GAIN,EN,CH,FWHM,ATAIL,SHAPL,FNAME,BWRDS,ALFA,EXP1,
2    EXP2,Y,PKPOS,PKHT,DUMMY
    DIMENSION NWRDS(4),NFLG(10),NXFLG(10),NYFLG(10),EN(2),
1    CH(2),FWHM(2),ATAIL(2),SHAPC(7),BWRDS(7),Y(50),PKPOS(5),
2    PKHT(5),DUMMY(140)
    SG = SHAPC(1) / (GAIN * GAIN) + SHAPC(2) * CH(IJ)/GAIN + 462
    F12M = SORT (SG)
    ST1 = CH(IJ) - 4.5 * F12M
    NDPTS = 8. * F12M
    CALL RDSK ('SYS',ST1,NDPTS,'RAYXXX',Y)
    CALL MAXVAL (1,NDPTS,JPK,BIGST,Y)
    ALFA = -2.7726 / SG
    EXP1 = EXP (EN(IJ) * SHAPC(4) + SHAPC(3))
    EXP2 = SHAPC(5) * GAIN
    SQTY1 = 2. * SORT (Y(1))
    SQTY2 = SORT (Y(NDPTS))
    LL = F12M
    L = JPK - LL
    LMT = L - 2
    MS = JPK + L.L
    DO 15 I = 1,LMT
    L = L - 1
    XL = L - JPK
    C1 = BIGST * EXP(ALFA * XL * XL)
    C2 = C1 + BIGST * EXP1 * EXP (EXP2 * XL)
    IF (C1 - SQTY2) 14,13,12
13  MS = MS + 1
14  IF (C2 - SQTY1) 16,15,15
15  CONTINUE
16  IF (L-3) 17,18,18
17  L = 3
18  NE = L
    NS = NE - 2
    ME = MS + 2
    CALL NCTS (NS,NE,MS,ME,NDPTS,BGD,YSUM,AVBG1,AVBG2,Y,Y)
    CALL MAXVAL (1,NDPTS,JPK,BIGST,Y)
    CALL CHFIT (JPK,REFFK,Y)
    START = ST1 + FLOA1 (NE)
    CH(IJ) = REFFK + START
    PKPOS(1) = REFFK
    PKHT(1) = BIGST
    AVEBG = AVBG1 + DBG * 0.5
    CHANS = KK' - K + 1

```

```

XPTS = NDPTS
R = CHANS / XPTS
SUM = YSUM + AVEBG * XPTS
PCTERR = 100. / YSUM * SQRT (SUM + R*R*BGD)
NFLG(1) = 1
NFLG(2) = 0
NFLG(3) = 0
IF (IFLG - 3) 23, 22, 21
21 NFLG(3) = 1
22 NFLG(2) = 1
23 NYFLG(1) = 1
    NXFLG(1) = 1
    NFLG(4) = 1
    NFLG(7) = 1
    CALL CHAIN ('FIT')
    END

```

Programs for Specific Analyses

The following programs are designed to interpret spectra taken with a high-resolution germanium spectrometer and to provide assay results for the isotopes of plutonium. The programs LEPA, LEPA2, and PUB are designed to operate on spectra taken of samples from which the ^{241}Am and ^{237}U have been recently removed. Both LEPA and LEPA2 use spectral information found below 60 keV, whereas PUB does not. Of these, the former two provide more precise determinations and in a much shorter counting period.

AMPUB and AXPUB operate on spectra taken of samples which are at least six to eight weeks old since the time of chemical separation. They provide an ^{241}Am determination in addition to results for the four isotopes of plutonium.

Most of these programs are designed to operate on spectra which have been taken at an approximate gain of 0.075 keV/channel with the zero energy being at or close to channel zero. The sample self-attenuation corrections written into the codes are specifically for a sample area of 10 cm^2 . Furthermore, a volume of 10 ml is assumed, and the final results are normalized to 1.0 ml.

PROGRAM: AMPUB

PURPOSE:

To quantitatively analyse plutonium solution samples containing ^{241}Am and ^{237}U in equilibrium with its ^{241}Pu parent.

INPUT:

- 1) Spectrum ID.
- 2) Device on which the spectrum is stored.

CHAIN OF PROGRAMS:

AMPUB→MATRX4→XMLR4→XMLA4→OUT4

FILES REQUIRED:

CFILEB, AFACTB

COMMENTS:

This is a program for determining the isotopic and total plutonium content of samples which have aged at least eight weeks since chemical processing. Like program PUB, it uses only the spectral region above 59 keV.

The sequence of the analysis and the peaks used is as follows:

- 1) Input shape constants from "CFILEB."
- 2) Analyze the 152 keV peak of ^{238}Pu .
- 3) Analyze the 129 keV peak of ^{239}Pu .
- 4) Analyze the 208 keV peak of $^{241}\text{Pu} - ^{237}\text{U}$.
- 5) Analyze the 148 keV peak of ^{241}Pu .
- 6) Analyze the 59 keV peak of ^{241}Am .
- 7) Isolate the 103 to 104 keV triplet peaks and form a set of equations consisting of the five peaks listed above plus the net counts of the 103.04, 103.684, and 104.243 keV triplet. After correcting for self-absorptions in the samples, the matrix of equations is solved by least-squares for the plutonium isotopes and ^{241}Am .
- 8) Final calculations are performed for atoms, grams, and isotopic percentages, and the results are printed out.

C AMPUB (FSID, FLDAT, RDSK, MAXVAL, NCNTS, NCTS, AVE/I)

```
COMMON NDPTS, NWRDS, FNAME, GAIN, TIME, ALFA, EXP1, EXP2, GSO,
1 AWRDS, SHAPC, EFAC, AREA, ERR, Y, YN
  DIMENSION NWRDS(4), AWRDS(3), SHAPC(10), EFAC(50), AREA(5),
1 ERR(5), Y(50), YN(230), EN(5)
  READ (1,100) FNAME
100 FORMAT ('SPECTRUM ID IS ',A6)
  READ (1,101) DEV
101 FORMAT ('DEVICE IS ',A6)
  IF (DEV - ' ') 2,4,2
2 CALL FLDAT (DEV, FNAME, 'RD')
4 CALL IOPEN ('SYS', 'CFILEB')
  READ(4,102) (SHAPC(I), I = 1,9)
102 FORMAT (A6)
  CALL IOPEN ('SYS', 'AFACTB')
  READ (4,102) (EFAC(I), I = 1,48)
```

CFILEB File Structure. See pg. 3

AFACTB File Structure (48 values)

Entries are in units of counts/disintegration.
No entry equal zero.

	²³⁸ Pu	²³⁹ Pu	²⁴⁰ Pu	²⁴¹ Pu	²⁴¹ Am
103.00	1	2 X	3	4	5 X
103.67	6	7	8	9 X	10
104.23	11	12	13 X	14	15
152.77	16 X	17	18	19	20
129.28	21	22 X	23	24	25
207.98	26	27	28	29 X	30 X
148.60	31	32	33	34 X	35
59.45	36	37	38	39 X	40 X
	41 1.65	42 1.64	43 1.63	44 2.50	45 3.70
	46 1.20	47 2.65	48 7.00		

Entries 41 to 48 are attenuation coefficients for the
gamma rays listed on the left.


```

C      ASSUME THE FOLLOWING APPROXIMATE GAIN
      GAIN = .075
      GSQ = GAIN * GAIN
      EXP2 = SHAPC(5) * GAIN
C      OBTAIN INTENSITIES FOR THE 152, 129, 208, 148 AND 59 KEV PEAKS
      EN(1) = 152.77
      EN(2) = 129.28
      EN(3) = 207.98
      EN(4) = 148.60
      EN(5) = 59.54
      ST1 = 40.
      NDPTS = 68
      NST = 10
      DO 30 J = 1,5
      REFCH = EN(J) / GAIN
      ST = REFCH - ST1
      CALL RDSK ('SYS', ST, NDPTS, 'RAYXXX', Y)
      ND = NDPTS - 10
      CALL MAXVAL (20, ND, JPK, BIGST, Y)
      SG = (SHAPC(1) + EN(J) * SHAPC(2)) / GSQ + 0.462
      ALFA = -2.7726 / SG
      F12M = SQRT (SG)
      EXP1 = EXP (EN(J) * SHAPC(4) + SHAPC(3))
      SQTY1 = SQRT (Y(1))
      SQTY2 = SQRT (Y(NDPTS))
      LL = F12M
      L = JPK - LL
      LMT = L - 2
      MS = JPK + LL
C      FINDING STARTING AND ENDING CHANNELS OF THE PEAK
      DO 20 J = 1, LMT
      L = L - 1
      DX = L - JPK
      X1 = BIGST * EXP (ALFA * DX * DX)
      X2 = X1 + BIGST * EXP1 * EXP (EXP2 * DX)
      IF (X1 - SQTY2) 14, 13, 13
13      MS = MS + 1
14      IF (X2 - SQTY1) 21, 20, 20
20      CONTINUE
21      NE = L
      NS = NE - NST
      IF (NS) 22, 22, 23
22      NS = 1
23      CALL NCNTS (NS, NE, MS, NDPTS, ND, BGD, SUM, ER, AVEBG, Y, Y)
      AREA(J) = SUM
      ERR(J) = ER
      IF (J - 4) 30, 25, 30
25      NST = 3
30      CONTINUE
      NDPTS = 230
      CALL RDSK ('SYS', 1190., NDPTS, 'RAYXXX', YN)
      CALL RDSK ('SYS', 1., 1, 'RAYXXX', TIME)
      TIME = TIME / 60.
      CALL CHAIN ('MATRX4')
      END

```

C MATRX4 (MAXVAL,NCTS,AVE,CHFIT)

C THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "AMPUB".

```
COMMON NDPTS,K,MEQ,NWRDS,FNAME,GAIN,TIME,ALFA,EXP1,EXP2,GSD,
1  AWRDS,SHAPC,EFACT,AREA,ERR,YNET,Y,ANS,RM,VM,
2  ACOEF
DIMENSION NWRDS(2),AWRDS(3),SHAPC(10),EFACT(5,10), AREA(5),
1  ERR(5),YNET(50),Y(60),ANS(5),RM(60),
2  VM(5),ACOE(60,5),DELEN(3),PKPOS(3)
C     THIS PROGRAM PREPARES PARTS OF THE 100 KEV REGION FOR ANALYSIS
CALL MAXVAL (50,90,JPK,BIGST,Y)
SG = (SHAPC(1) + 102. * SHAPC(2)) / GSD + .462
ALFA = -2.7726 / SG
F12M = SQRT (SG)
EXP1 = EXP (SHAPC(3) + 102. * SHAPC(4))
C1 = 1.0 / (GAIN * F12M)
C1 = 0.3 * C1 * C1 / 1.05
C2 = -2.44 / F12M
SQTY1 = SQRT (Y(20))
LL = F12M
MS = JPK
CALL BOUNDS (LL,MS,BIGST,ALFA,EXP1,EXP2,C1,C2,SQTY1,SQTY1)
NS = LL - 10
NA = (JPK - LL) / 3
NE = JPK + NA + 131
CALL NCTS (NS,LL,NE,NDPTS,NDPTS,BGD,SUM,AVBG1,AVBG2,Y,Y)
C     ISOLATE THE 103 - 104 KEV PEAKS BY SUBTRACTING OUT 101 KEV
C     COMPONENT.
CALL MAXVAL (90,120,JPK,BIGST,Y)
CALL CHFIT (JPK,REFPK,Y)
NS = JPK + 8
NE = NS + 12
DO 2 I = NS,NE
XI = I
DX = XI - REFPK
2  Y(I) = Y(I) - BIGST*(EXP(ALFA*DX*DX) + C1*EXP(C2*DX))
L = 0
NS = NS + 12 - NA
DO 4 I = NS,NDPTS
L = L + 1
4  YNET(L) = Y(I)
NDPTS = L
REFPK = REFPK - FLOAT (NS - 1)
DELEN(1) = 1.902
DELEN(2) = 2.613
DELEN(3) = 3.173
DO 5 J = 1,3
5  PKPOS(J) = REFPK + DELEN(J) / GAIN
RATIO = .989 / SHAPC(6)
AMTPU = 3.95E-12 * AREA(2) / (TIME * EFACT(2,5))
L = 40
```

```

C      APPLY SELF-ABSORPTION CORRECTIONS.
      DO 7 J = 1,8
        L = L + 1
        CORR = 1. - AMTPU * EFACT(L)
        DO 6 I = 1,5
          EFACT(I,J) = EFACT(I,J) * CORR
6      CONTINUE
C      COMPUTE MATRIX COEFFICIENTS AND FORM A SET OF LINEAR EQUATIONS.
      DO 20 I = 1,NDPTS
        WTF = 1.0 / SQRT (YNET(I) + AVBG2)
        XI = I
        DO 8 K = 1,5
          ACOEF(I,K) = 0.0
          DO 18 J = 1,3
            DX = XI - PKPOS(J)
            DDX = DX * DX
            ADX = ALFA * DDX
            IF (ADX + 20.) 18,10,10
10         FXX = EXP (ADX)
            EXPN3 = EXP (0.4 * ADX)
            IF (DX) 12,14,14
12         EXPN2 = EXP (EXP2 * DX)
            FXX = FXX + EXP1 * EXPN2 * (1. - EXPN3)
14         DO 17 K = 2,5
            IF (EFACT(K,J)) 16,17,16
16         ACOEF(I,K) = ACOEF(I,K) + EFACT(K,J) * RATIO * FXX
17         CONTINUE
18         CONTINUE
          Y(I) = YNET(I) * WTF
          DO 19 K = 1,5
19         ACOEF(I,K) = ACOEF(I,K) * WTF
20         CONTINUE
        MEQ = NDPTS + 5
        L = NDPTS + 1
        M = 3
        J = 0
        DO 32 I = L,MEQ
          M = M + 1
          J = J + 1
          Y(I) = AREA(J) / ERR(J)
          YNET(I) = AREA(J)
          DO 30 K = 1,5
30         ACOEF(I,K) = EFACT(K,M) / ERR(J)
32         CONTINUE
        K = 5
        CALL CHAIN ('XMLR4')
      END

```

C XMLR4 (MLR/O)

C THIS SUBPROGRAM IS THE THIRD LINK TO THE PROGRAM "AMPUB".

```

COMMON NDPTS, K, MEQ, NERR, NWRDS, AWRDS, Y, ANS, RM, VM, ACDEF, WM
1  DIMENSION AWRDS(130), Y(60), ANS(5), RM(60), VM(5), ACDEF(60, 5),
    WM(5)
    NERR = 1
    CALL OOPEN ('DSK1', 'AMTRX')
    NM = 60 * K
    WRITE (4, 100) (ACDEF(J), I = 1, NM)
100  FORMAT (10A6)
    CALL OCLOSE
    CALL MLR (60, MEQ, K, Y, ANS, ACDEF, RM, WM, VM, NERR)
    CALL CHAIN ('XMLA4')
    END

```

C XMLA4 (MLRA/I)

C THIS SUBPROGRAM IS THE FOURTH LINK TO THE PROGRAM "AMPUB".

```

COMMON NDPTS, K, MEQ, NERR, NWRDS, AWRDS, Y, ANS, RM, VM, BM, WM
    DIMENSION AWRDS(130), Y(60), ANS(5), RM(60), VM(5), BM(300), WM(5)
    IF (NERR -2) 2, 4, 2
2    CALL MLRA (60, MEQ, K, Y, ANS, BM, RM, WM, VM, NERR)
4    GO TO (5, 6, 7) NERR
6    WRITE (1, 101)
101  FORMAT ('MATRIX IS SINGULAR')
    GO TO 10
7    WRITE (1, 102)
102  FORMAT ('SUM OF SQUARES OF R > THAN THOSE OF Y')
10    CALL EXIT
5    CALL CHAIN ('OUT4')
    END

```

C OUT4 (MLRE)

C THIS SUBPROGRAM IS THE FIFTH LINK TO THE PROGRAM "AMPUB".

```

COMMON NDPTS, K, MEQ, NERR, NWRDS, FNAME, GAIN, TIME, ALFA, EXP1,
1  EXP2, GSQ, AWRDS, SHAPC, EFAC, AREA, ERR, YNET, Y, ANS, RM, DM, BM, WM
    DIMENSION AWRDS(3), SHAPC(10), EFAC(50), AREA(5), ERR(5), YNET(50),
1  Y(60), ANS(5), RM(60), DM(5), BM(300), WM(5), PCT(5), SIGSQ(5),
2  FRAC(5), T1HLF(5), ATOMS(5)
    T1HLF(1) = 4.613 E07
    T1HLF(2) = 1.283 E10
    T1HLF(3) = 3.471 E09
    T1HLF(4) = 7.626 E06
    CALL MLRE (DM, 60, K, BM, WM)
    SUM = 0.0
    DO 40 I = 1, MEQ

```

```

40      SUM = SUM + RM(I) **2
      DOF = MEQ - 5
      QFIT = SUM / DOF
      SUM = 0.0
      ATNUM = 237.
      GRMS = 0.0
      SUMSQ = 0.0
      ADPM = 0.0
      DO 50 J = 1,4
      ATNUM = ATNUM + 1
      ERR(J) = SQRT (QFIT * DM(J)) / ANS(J)
      ANS(J) = ANS(J) / TIME
      ADPM = ADPM + ANS(J)
      ATOMS(J) = ANS(J) * T1HLF(J) / .693147
      GRMS = GRMS + 1.6603 E-21 * ATOMS(J) * ATNUM
      SIG = ERR(J) * ATOMS(J)
      SIGSQ(J) = SIG * SIG
      SUMSQ = SUMSQ + SIGSQ(J)
50      SUM = SUM + ATOMS(J)
      SIG = (100./SUM) * SQRT (SUMSQ)
      DO 52 J = 1,4
      FRAC(J) = ATOMS(J) / SUM
      SX = SUM - ATOMS(J)
      PCT(J)=(100.*SX/SUM)*SQRT(ERR(J)*ERR(J)+(SUMSQ-SIGSQ(J))
1      / (SX * SX))
52      ERR(J) = 100. * ERR(J)
      ERR(5) = 100. * SQRT (QFIT * DM(5)) / ANS(5)
      ANS(5) = ANS(5) / TIME
      ATOMS(5) = ANS(5) * 3.286E+08
      ADPM = ADPM + ANS(5) - ANS(4)
      WRITE (1,102) FNAME,TIME,GRMS,SIG,QFIT,ADPM
102     FORMAT('ANALYSIS OF SAMPLE ',A6,F9.1,' MINUTES','MGMS PU/ML =',
1      F7.3,' / +/-',F6.3,'%','QFIT =',F5.2,' ALPHA DPM =',E11.4/
2      10X, 'DPM/ML      ATOMS/ML  PCTERR  ISOTOPIC PCTERR')
      WRITE (1,103) ANS(1),ATOMS(1),ERR(1),FRAC(1),PCT(1),ANS(2),
1      ATOMS(2),ERR(2),FRAC(2),PCT(2),ANS(3),ATOMS(3),ERR(3),FRAC(3),
2      PCT(3),ANS(4),ATOMS(4),ERR(4),FRAC(4),PCT(4),ANS(5),ATOMS(5),
3      ERR(5)
103     FORMAT ('PU238 =',2E11.4,F6.2,F11.6,F7.3/'PU239 =',2E11.4,
1      F6.2,F11.6,F7.3/'PU240 =',2E11.4,F6.2,F11.6,F7.3/'PU241 =',
2      2E11.4,F6.2,F11.6,F7.3/'AM241 =',2E11.4,F6.2)
      IF (QFIT - 1.5) 60,59,59
59      WRITE (1,104) (I,YNET(I),RM(I), I = 1,MEQ)
104     FORMAT (/ 'X',YNET   STD DEV' / (I2,F10.1,F8.2))
60      CALL EXIT
      END

```

PROGRAM:**AXPUB****PURPOSE:**

To quantitatively analyze plutonium solution samples containing ^{241}Am and ^{237}U in equilibrium with its ^{241}Pu parent.

INPUT:

- 1) Spectrum ID.
- 2) Device on which the spectrum is stored.

CHAIN OF PROGRAMS:

AXPUB→MATRX5→XMLR5→XMLA5→OUT5

FILES REQUIRED:

CFILB, AXFCTB, AXCNST

COMMENTS:

This is a program for determining the isotopic and

total plutonium content of samples which have aged at least eight weeks since chemical processing. It is very similar to the AMPUB program. The principal difference is that this program incorporates the entire set of gamma- and x-rays in the 100 keV region.

Steps 1 to 6 in the analysis sequence are the same as in AMPUB. The remaining part of the analysis is as follows:

- 7) Analysis of the 94 keV peak.
- 8) Isolate the 97 to 104 keV set of peaks and form a set of equations consisting of the six individual peaks plus the net counts of the 97.08, 98.446, 98.77, 98.97, 99.87, 101.06, 103.00, 103.66, and 104.23 keV peaks. The matrix set of equations is solved by least-squares for the plutonium isotopes and ^{241}Am .
- 9) Final calculations are performed for atoms, grams, and isotopic percentages, and the results printed out.

AXFCTB File Structure (75 values)

(Entries are in units of counts/disintegration.

No entry equals zero.)

	^{238}Pu	^{239}Pu	^{240}Pu	^{241}Pu	^{241}Am
97.08	1	2	3	4 X	5 X
98.446	6 X	7 X	8 X	9 X	10
98.77	11	12 X	13	14	15
98.97	16	17	18	19	20 X
99.87	21 X	22	23	24	25
101.06	26	27	28	29 X	30 X
102.97	31	32 X	33	34	35 X
103.66	36	37	38	39 X	40
104.23	41	42	43 X	44	45
152.77	46 X	47	48	49	50
129.28	51	52 X	53	54	55
207.98	56	57	58	59 X	60 X
148.60	61	62	63	64 X	65
59.45	66	67	68	69 X	70 X
94.665	71 X	72 X	73 X	74 X	75

AXCNST File on "SYS@@" (30 values)

1	152.77	-	^{238}Pu
2	129.28	-	^{239}Pu
3	207.98	-	^{241}Pu , ^{237}U and ^{241}Am
4	148.60	-	^{241}Pu
5	59.450	-	^{241}Pu and ^{241}Am
6	94.665	-	U-K_{α_1} x-ray
7	97.080	-	Np-K_{α_1} x-ray
8	98.446	-	U-K_{α_2} x-ray
9	98.770	-	^{239}Pu
10	98.970	-	^{241}Am
11	99.870	-	^{238}Pu
12	101.06	-	Np-K_{α_2} x-ray
13	102.97	-	^{239}Pu and ^{241}Am
14	103.66	-	^{241}Pu
15	104.23	-	^{240}Pu
16	1.95	}	Absorption coefficients for the above radiations
17	1.90		
18	1.90		
19	1.87		
20	1.83		
21	1.78		
22	1.65		
23	1.64		
24	1.63		
25	2.50		
26	3.70		
27	1.20		
28	2.65		
29	7.00		
30	2.10		

```

C      AXSUB (FSIO, FLDAT, RDSK, MAXVAL, BOUNDS, NCTS, AVE, CHFIT, STRIP/I)

      COMMON NDPTS, NWRDS, FNAME, GAIN, TIME, ALFA, EXP1, EXP2, GSQ, ST,
1      AVEBG, SG, C1, C2, AWRDS, AREA, ERR, PKPOS, Y, EFAC, SHAPC, XMU, EN
      DIMENSION NWRDS(4), SHAPC(10), EFAC(75), AREA(6), ERR(6),
1      PKPOS(10), Y(140)
      DIMENSION XMU(15), EN(15)
      READ (1,100) FNAME
100     FORMAT ('SPECTRUM ID IS ',A6)
      READ (1,101) DEV
101     FORMAT ('DEVICE IS ',A6)
      IF (DEV - ' ') 2,4,2
2      CALL FLDAT (DEV,FNAME,'RD')
4      CALL IOPEN ('SYS','CFILEB')
      READ(4,102) (SHAPC(I), I = 1,9)
102     FORMAT (A6)
      CALL IOPEN ('SYS','AXCNST')
      READ (4,102) (EN(I), I = 1,15)
      READ (4,102) (XMU(I), I = 1,15)
C      ASSUME THE FOLLOWING APPROXIMATE GAIN
      GAIN = .075
      GSQ = GAIN * GAIN
      EXP2 = SHAPC(5) * GAIN
C      OBTAIN INTENSITIES FOR THE 152,129,208,148 AND 59 KEV PEAKS
      ST1 = 40.
      NDPTS = 68
      NST = 10
      DO 30 J = 1,5
      REFCH = EN(J) / GAIN
      ST = REFCH - ST1
      CALL RDSK ('SYS',ST,NDPTS,'RAYXXX',Y)
      ND = NDPTS - 10
      CALL MAXVAL (20,ND,JPK,BIGST,Y)
      SG = (SHAPC(1) + EN(J) * SHAPC(2)) / GSQ + 0.462
      ALFA = -2.7726 / SG
      F12M = SQRT (SG)
      EXP1 = EXP (EN(J) * SHAPC(4) + SHAPC(3))
      SQTY1 = SQRT (Y(1))
      SQTY2 = SQRT (Y(NDPTS))
      LL = F12M
      MS = JPK
      CALL BOUNDS (LL,MS,BIGST,ALFA,EXP1,EXP2,0.,0.,SQTY1,SQTY2)
      NE = LL
      NS = NE - NST
      IF (NS) 22,22,23
22     NS = 1
23     CALL NCTS (NS,NE,MS,NDPTS,ND,BGD,SUM,AVBG1,AVBG2,Y,Y)
      AVEBG = 0.5 * (AVBG1 + AVBG2)
      CHANS = MS - NE + 1
      XPTS = NDPTS + NE - MS - NS + 1
      IF (SUM) 26,27,27
26     SUM = 0.0
27     AREA(J) = SUM
      ERR(J) = SQRT (SUM + AVEBG * CHANS * (1. + CHANS / XPTS))
      IF (J - 4) 30,25,30

```



```

25     NST = 3
30     CONTINUE
C     READ IN 100 KEV PEAK REGION AND PREPARE FOR ANALYSIS
      NDPTS = 215
      CALL RDSK ('SYS', 1205, NDPTS, 'RAYXXX', Y)
      CALL RDSK ('SYS', 1, 1, 'RAYXXX', TIME)
      TIME = TIME / 60
      CALL MAXVAL (35, 75, JFK, BIGST, Y)
      CALL CHFIT (JFK, PK94, Y)
      SG = (SHAPC(1) + 100. * SHAPC(2)) / GSD + 0.462
      ALFA = -2.7726 / SG
      F12M = SQRT (SG)
      EXP1 = EXP (100. * SHAPC(4) + SHAPC(3))
      BGD = Y(40)
      SQTY1 = SQRT (BGD)
      LL = F12M
      C1 = 1.0 / (GAIN * F12M)
      C1 = 0.3 * C1 * C1 / 1.05
      C2 = -2.44 / F12M
      MS = JFK
      CALL BOUNDS (LL, MS, BIGST, ALFA, EXP1, EXP2, C1, C2, SQTY1, SQTY1)
      CALL MAXVAL (85, 95, JP, HT97, Y)
      CALL CHFIT (JP, PK97, Y)
      CALL BOUNDS (ME, JP, HT97, ALFA, EXP1, EXP2, C1, C2, SQTY1, SQTY1)
      IF (ME - MS) 31, 31, 32
31     MS = (ME + MS) / 2 - 1
      ME = MS + 1
32     PKHT = BIGST - BGD
C     ISOLATE THE 94 KEV PEAK.
      CALL STRIP (MS, ME, PK94, PKHT, ALFA, EXP1, EXP2, C1, C2, Y)
      PKHT = HT97 - BGD
      CALL STRIP (MS, ME, PK97, PKHT, ALFA, EXP1, EXP2, C1, C2, Y)
C     OBTAIN THE AREA OF THE 94 KEV PEAK.
      CALL NCTS (NS, LL, MS, ME, ND, BGD, SUM, AVBG1, AVBG2, Y, Y)
      AREA(6) = SUM
      CHANS = MS - LL + 1
      XPTS = ME + LL - MS - NS + 1
      ERR(6) = SQRT (SUM + AVBG1 * CHANS * (1. + CHANS / XPTS))
C     OBTAIN THE NET CHANNEL COUNTS FOR THE 97-104 KEV PEAK REGION.
      NA = (JFK - LL) / 3
      NE = JFK + NA + 131
      CALL NCTS (MS, ME, NE, NDPTS, NDPTS, BGD, SUM, AVBG1, AVBG2, Y, Y)
      ST = ME + 1204
      CALL MAXVAL (60, 80, JFK, BIGST, Y)
      CALL CHFIT (JFK, PK101, Y)
      GAIN = 0.5 * (94.665 / (1204. + PK94) + 101.07 / (ST + PK101))
      CALL IOPEN ('SYS', 'AXFCTB')
      READ (4, 102) (EFACT(I), I = 1, 75)
      CALL CHAIN ('MATRX5')
      END

```

```

C      MATRX5

C      THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "AXPUB".

      COMMON NDPTS, K, MEQ, NWRDS, FNAME, GAIN, TIME, ALFA, EXP1, EXP2, GSG,
1  ST, AVEBG, SG, C1, C2, AWRDS, AREA, ERR, PKPOS, Y, EFACF,
2  SHAPC, XMU, EN, BWRDS, YNET, ACOEF
      DIMENSION NWRDS(2), SHAPC(10), EFACF(5, 15), AREA(6), ERR(6),
1  PKPOS(10), Y(140), YNET(140), XMU(15), EN(15), BWRDS(35), ACOEF(140, 5)
      DIMENSION NFLG(9)

C      DETERMINE THE EXACT POSITIONS OF THE PEAKS IN THE 97 TO 104 KEV
C      REGION
      L = 6
      DO 35 J = 1, 9
        L = L + 1
35    PKPOS(J) = EN(L) / GAIN - ST
C      APPLY SELF-ABSORPTION CORRECTIONS.
      AMTPU = 3.95E-12 * AREA(2) / (TIME * EFACF(2, 11))
      DO 7 J = 1, 15
        CORR = 1. - AMTPU * XMU(J)
        DO 6 I = 1, 5
          EFACF(I, J) = EFACF(I, J) * CORR
6      CONTINUE
7      ASLP = SHAPC(2) / GAIN
      S100 = 100. / GAIN - ST
      RATIO = 0.989 / SHAPC(6)

C      SET FLAGS. 0 = GAMMA RAYS; 1 = X-RAYS
      NFLG(1) = 1
      NFLG(2) = 1
      NFLG(3) = 0
      NFLG(4) = 0
      NFLG(5) = 0
      NFLG(6) = 1
      NFLG(7) = 0
      NFLG(8) = 0
      NFLG(9) = 0

C      COMPUTE MATRIX COEFFICIENTS AND FORM A SET OF LINEAR EQUATIONS.
      DO 20 I = 1, NDPTS
        YNET(I) = Y(I)
        WTF = 1.0 / SQRT (YNET(I) + AVEBG)
        XI = I
        DO 8 K = 1, 5
          ACOEF(I, K) = 0.0
          DO 18 J = 1, 9
            DX = XI - PKPOS(J)
            DDX = DX * DX
            ALPHA = ALFA * SG / (ASLP * (PKPOS(J) - S100) + SG)
            ADX = ALPHA * DDX
            IF (ADX + 20.) 18, 10, 10
10         FXX = EXP (ADX)
            EXPN3 = EXP (0.4 * ADX)
            RT = SQRT (ALPHA / ALFA)
            IF (DX) 12, 14, 14

```

```

12     EXPN2 = EXP (EXP2 * DX)
      FXX = FXX + EXP1 * EXPN2 * (1 - EXPN3)
      DX = -DX
14     RAT = RATIO * RT
      IF (NFLG(J)) 13,15,13
13     FXX = FXX + C1 * (1.05 - EXPN3) * EXP(C2 * DX)
      RAT = RAT / SHAPC(7)
15     DO 17 K = 1,5
      IF (EFACT(K,J)) 16,17,16
16     ACOEF(I,K) = ACOEF(I,K) + EFACT(K,J) * RAT * FXX
17     CONTINUE
18     CONTINUE
      Y(I) = YNET(I) * WTF
      DO 19 K = 1,5
19     ACOEF(I,K) = ACOEF(I,K) * WTF
20     CONTINUE
      MEQ = NDPTS + 6
      L = NDPTS + 1
      M = 9
      J = 0
      DO 32 I = L,MEQ
      M = M + 1
      J = J + 1
      Y(I) = AREA(J) / ERR(J)
      YNET(I) = AREA(J)
      DO 30 K = 1,5
30     ACOEF(I,K) = EFACT(K,M) / ERR(J)
32     CONTINUE
      K = 5
      CALL CHAIN ('XMLR5')
      END

```

C XMLR5 (MLR/D)

C THIS SUBPROGRAM IS THE THIRD LINK TO THE PROGRAM "AXPUB"

```

      COMMON NDPTS,K,MEQ,NERR,NWRDS,AWRDS,Y,ANS,RM,VM,YNET,ACOE,WM
      DIMENSION AWRDS(35),Y(140),YNET(140),ANS(5),RM(140),VM(5),
1     ACOEF(140,5),WM(5)
      NERR = 1
      WRITE (1,200) (AWRDS(I), I = 14,20)
200    FORMAT (5F12.1)
      CALL QOPEN ('DSK1','AMTRX')
      NM = 140 * K
      WRITE (4,100) (ACOE(I), I = 1,NM)
100    FORMAT (10A6)
      CALL OCLOSE
      CALL MLR (140,MEQ,K,Y,ANS,ACOE,RM,WM,VM,NERR)
      CALL CHAIN ('XMLR5')
      END

```

C XMLA5 (MLRA/1)

C THIS SUBPROGRAM IS THE FOURTH LINK TO THE PROGRAM "AXPUB".

```
COMMON NDPTS, K, MEQ, NERR, NWRDS, AWRDS, Y, ANS, RM, VM, YNET, BM, WM
DIMENSION AWRDS(35), Y(140), YNET(140), ANS(5), RM(140), VM(5),
1 BM(700), WM(5)
IF (NERR -2) 2,6,3
2 CALL MLRA (140, MEQ, K, Y, ANS, BM, RM, WM, VM, NERR)
4 GO TO (5,6,7) NERR
6 WRITE (1,101)
101 FORMAT ('MATRIX IS SINGULAR')
GO TO 10
7 WRITE (1,102)
102 FORMAT ('SUM OF SQUARES OF R > THAN THOSE OF Y')
10 CALL EXIT
5 CALL CHAIN ('OUT5')
END
```

C OUT5 (MLRE)

C THIS SUBPROGRAM IS THE FIFTH LINK TO THE PROGRAM "AXPUB".

```
COMMON NDPTS, K, MEQ, NERR, NWRDS, FNAME, GAIN, TIME, ALFA, EXP1,
1 EXP2, GSQ, ST, AVEBG, SG, C1, C2, AWRDS, AREA, ERR, PKPOS,
2 Y, ANS, RM, DM, YNET, BM, WM
DIMENSION AREA(6), ERR(6), PKPOS(10),
1 Y(140), YNET(140), ANS(5), RM(140), DM(5), BM(700), WM(5)
DIMENSION PCT(5), SIGSQ(5), FRAC(5), T1HLF(5), ATOMS(5)
T1HLF(1) = 4.613 E07
T1HLF(2) = 1.283 E10
T1HLF(3) = 3.471 E09
T1HLF(4) = 7.626 E06
CALL MLRE (DM, 140, K, BM, WM)
SUM = 0.0
DO 40 I = 1, MEQ
40 SUM = SUM + RM(I) **2
DOF = MEQ - 5
QFIT = SUM / DOF
SUM = 0.0
ATNUM = 237
GRMS = 0.0
SUMSQ = 0.0
ADPM = 0.0
DO 50 J = 1, 4
ATNUM = ATNUM + 1
ERR(J) = SQRT (QFIT * DM(J)) / ANS(J)
ANS(J) = ANS(J) / TIME
ADPM = ADPM + ANS(J)
ATOMS(J) = ANS(J) * T1HLF(J) / .693147
GRMS = GRMS + 1.6603 E-21 * ATOMS(J) * ATNUM
SIG = ERR(J) * ATOMS(J)
SIGSQ(J) = SIG * SIG
SUMSG = SUMSG + SIGSQ(J)
```

```

50      SUM = SUM + ATOMS(J)
      SIG = (100./SUM) * SQRT (SUMSG)
      DO 52 J = 1,4
      FRAC(J) = ATOMS(J) / SUM
      SX = SUM - ATOMS(J)
      PCT(J)=(100.*SX/SUM)*SQRT(ERR(J)*ERR(J)+(SUMSG-SIGSQ(J))
1      / (SX * SX))
52      ERR(J) = 100. * ERR(J)
      ERR(5) = 100. * SQRT (QFIT * DM(5)) / ANS(5)
      ANS(5) = ANS(5) / TIME
      ATOMS(5) = ANS(5) * 3.286E+08
      ADPM = ADPM + ANS(5) - ANS(4)
      WRITE (3,102) FNAME, TIME, GRMS, SIG, QFIT, ADPM
102     FORMAT ('ANALYSIS OF SAMPLE ', A6, F9.1, ' MINUTES'///MGMS PU/ML =',
1      F7.3, ' +/-', F6.3, '%', 'QFIT =', F5.2, ' ALPHA DPM =', E11.4/10X,
2      'DPM/ML   ATOMS/ML  PCTERR  ISOTOPIC PCTERR')
      WRITE (3,103) ANS(1), ATOMS(1), ERR(1), FRAC(1), PCT(1), ANS(2),
1      ATOMS(2), ERR(2), FRAC(2), PCT(2), ANS(3), ATOMS(3), ERR(3), FRAC(3),
2      PCT(3), ANS(4), ATOMS(4), ERR(4), FRAC(4), PCT(4), ANS(5), ATOMS(5),
3      ERR(5)
103     FORMAT ('PU238 =', 2E11.4, F6.2, F11.6, F7.3/'PU239 =', 2E11.4,
1      F6.2, F11.6, F7.3/'PU240 =', 2E11.4, F6.2, F11.6, F7.3/'PU241 =',
2      2E11.4, F6.2, F11.6, F7.3,/'AM241 =', 2E11.4, F6.2)
      IF (QFIT - 1.5) 60,59,59
59      WRITE (3,104) (I, YNET(I), RM(I), I = 1, MEQ)
104     FORMAT (/ 9X, 'YNET   STD DEV' / (I4, F10.1, F8.2))
60      CALL EXIT

```

PROGRAM:
LEPA

PURPOSE:

To quantitatively analyze a recently processed sample of plutonium.

INPUT:

- 1) Spectrum ID.
- 2) Device on which the spectrum is stored.

CHAIN OF SUBPROGRAMS:

LEPA → PK45 → REPORT

FILES REQUIRED:

CFILEB, LEFILB, PUBKGB

COMMENTS:

This is the fastest of the current plutonium analysis programs because no least-squares fitting is performed. It also provides better results than does the PUB program. It assumes that ^{241}Am has been removed and that the ^{237}U has not grown back in to an appreciable extent. The program assumes an approximate gain of 0.075 keV/channel.

The sequence of the analysis and the peaks use is as follows:

- 1) Input of shape constants from "CFILEB" disk file.
- 2) Analysis of 148 keV peak of ^{241}Pu .
- 3) Analysis of 51 keV peak of ^{239}Pu .
- 4) Subtract a background continuum from the 41 to 47 keV region due to ^{239}Pu (see PUBKGD file).
- 5) Strip out 44.9 keV peak (due to ^{241}Pu) and 46.23 keV peak (due to ^{239}Pu) from this region.
- 6) Analysis of 43 keV peak of ^{238}Pu .
- 7) Analysis of 45 keV peak of ^{240}Pu .
- 8) Correct constants for gamma ray attenuation in the source.
- 9) Make final calculations for atoms, grams, and isotopic percentages and printout results.

CFILEB File Structure (See pg. 3)

LEFILB File Structure (7 values)

Entry

1. Counts/disintegration for the 43.492 keV peak of ^{238}Pu .
2. Counts/disintegration for the 51.628 keV peak of ^{239}Pu .
3. Counts/disintegration for the 45.235 keV peak of ^{240}Pu .
4. Counts/disintegration for the 148.60 keV peak of ^{241}Pu .
5. This entry equals the reciprocal of the number of counts in the 51 keV ^{239}Pu peak of the spectrum from which background data was obtained for file "PUBKGD."
6. This entry is the observed intensity ratio of the 44.19 to 148.60 keV peaks of ^{241}Pu .
7. This entry is the observed intensity ratio of the 46.23 to 51.628 keV peaks of ^{239}Pu .

PUBKGD File Structure (100 values)

This file of numbers describes the background continuum in the energy range from 42 to 49.5 keV due to the decay of ^{239}Pu . This background, arising principally from the scattering of the 51 keV gamma ray, is quite nonlinear. A direct measurement of this shape is, therefore, used by LEPA and LEPA2. An isotopically pure ^{239}Pu sample should first be counted in the same configuration as the normal samples. The data from channels 560 to 659 are then printed out using the DATARD routine. (A gain of 0.075 keV/channel is assumed.) The 46.23 keV peak in this region is then located and the background under it is interpolated by hand. The 100 data points, including the modified values, are then stored in the PUBKGD file.

The 51 keV peak should also be integrated using PKANAL. The reciprocal of the net count must be used as entry 5 in file LEFILB and as entry 4 in file LSFILB.

C LEPA (FSIO, FLDAT, RDSK, MAXVAL, NCTS, AVE/I)

```
COMMON FNAME, GAIN, EXPZ, AMTPU, TIME, SHAPC, AREA, ERR, ECONST, Y
DIMENSION SHAPC(9), AREA(4), ERR(4), ECONST(10), Y(100)
READ (1, 100) FNAME
```

```

100  FORMAT ('SPECTRUM ID IS ',A6)
    READ (1,101) DEV
101  FORMAT ('DEVICE IS ',A6)
    IF (DEV - ' ') 2,4,2
2    CALL FLDAT (DEV,FNAME,'RD')
4    CALL IOPEN ('SYS','CFILEB')
    READ (4,102) (SHAPC(I), I = 1,9)
102  FORMAT (A6)
C    ASSUME FOLLOWING APPROXIMATE GAIN
    GAIN = .075
C    OBTAIN 148 AND 51 KEV PEAK INTENSITIES
    EN = 148.
    ST = 1958.
    NDPTS = 62
    GSQ = GAIN * GAIN
    DO 30 J = 1,2
    CALL RDSK ('SYS',ST,NDPTS,'RAYXXX',Y)
    CALL MAXVAL (10,NDPTS,JPK,BIGST,Y)
    SG = (SHAPC(1) + EN * SHAPC(2)) / GSQ + 0.462
    ALFA = -2.7726 / SG
    F12M = SQRT (SG)
    EXP1 = EXP (EN * SHAPC(4) + SHAPC(3))
    EXP2 = SHAPC(5) * GAIN
    SQTY1 = SQRT (Y(1))
    SQTY2 = SQRT (Y(NDPTS))
    LL = F12M
    L = JPK - LL
    LMT = L - 2
    MS = JPK + LL
C    FIND STARTING AND ENDING CHANNELS OF THE PEAK
    DO 20 I = 1,LMT
    L = L - 1
    DX = L - JPK
    X1 = BIGST * EXP (ALFA * DX * DX)
    X2 = X1 + BIGST * EXP1 * EXP (EXP2 * DX)
    IF (X1 - SQTY2) 14,13,13
13    MS = MS + 1
14    IF (X2 - SQTY1) 21,20,20
20    CONTINUE
21    NE = L
    NS = NE - 7
    IF (NS) 22,22,23
22    NS = 1
23    CALL NCTS (NS,NE,MS,NDPTS,ND,BGD,SUM,AVBG1,AVBG2,Y,Y)
    AVEBG = 0.5 * (AVBG1 + AVBG2)
    CHANS = MS - NE + 1
    XPTS = NDPTS + NE - MS - NS + 1
    AREA(J) = SUM
    ERR(J) = SQRT (SUM + AVEBG * CHANS * (1. + CHANS / XPTS))
    EN = 51.
    ST = 670.
    NDPTS = 44
30    CONTINUE
    AREA(4) = AREA(1)
    ERR(4) = ERR(1)
    CALL CHAIN ('PK45')
42    END

```

```

C      PK45 (FSIO, RDSK, MAXVAL, AVE, CHFIT, NCTS/I)

C      THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "LEPA".

      COMMON FNAME, GAIN, EXP2, AMTPU, TIME, SHAPC, AREA, ERR, ECONST,
1      Y, BUFF
      DIMENSION SHAPC(9), AREA(4), ERR(4), ECONST(10), Y(100), BUFF(100)
C      CALL IN 41-47 KEV REGION
      CALL RDSK ('SYS', 560., 100, 'RAYXXX', Y)
      CALL MAXVAL (15, 30, JPK, BIGST, Y)
      GSD = GAIN * GAIN
      SG = (SHAPC(1) + 45. * SHAPC(2)) / GSD + 0.462
      ALFA = -2.7726 / SG
      F12M = SQRT (SG)
      EXP1 = EXP (SHAPC(3) + 45. * SHAPC(4))
      C45 = SQRT (1.133 * SG)
      BG1 = Y(15)
      BG2 = Y(32)
      SQTY1 = SQRT (BG1)
      SQTY2 = SQRT (BG2)
      BIGST = BIGST - BG2
C      DETERMINE THE INTEGRATION LIMITS FOR THE 43 AND 45 KEV PEAKS
      DO 8 J = 1, 2
        LL = F12M
        L = JPK - LL
        LMT = L - 2
        MS = JPK + LL
        DO 5 I = 1, LMT
          L = L - 1
          DX = L - JPK
          X1 = BIGST * EXP (ALFA * DX * DX)
          X2 = X1 + BIGST * EXP1 * EXP (EXP2 * DX)
          IF (X1 - SQTY2) 4, 3, 3
3          MS = MS + 1
4          IF (X2 - SQTY1) 6, 5, 5
5          CONTINUE
6          IF (J-1) 7, 7, 9
7          NE1 = L
          MS1 = MS
          SQTY1 = SQTY2
          CALL MAXVAL (38, 52, JPK, BIGST, Y)
          BIGST = BIGST - BG2
8          CONTINUE
9          NE = L
          CALL AVE (MS, 80, SUM, BG3, Y)
          CALL IQPEN ('SYS', 'LEFILB')
          READ (4, 102) (ECONST(I), I = 1, 7)
102      FORMAT (A6)
C      OBTAIN AN ESTIMATE OF THE AMOUNT OF PU PRESENT.
      AMTPU = 7.38E-12 * AREA(2) / ECONST(2) + 4.38E-15 * AREA(4) /
1      ECONST(4) + 2.00E-12 * C45 * BIGST / ECONST(3)
      CALL RDSK ('SYS', 1., 1, 'RAYXXX', TIME)
      TIME = TIME / 60.
      AMTPU = AMTPU / TIME
      PK51 = (5.1 * AMTPU + 1.) * AREA(2)
      BGFACT = ECONST(5) * PK51

```



```

C      READ IN A FILE DELINEATING THE SHAPE OF THE BACKGROUND CONTINUUM
C      IN THE 45 KEV REGION DUE TO 239PU.
      CALL IOPEN ('SYS', 'PUBKGB')
      READ (4,102) (BUFF(I), I = 1,100)
C      SUBTRACT BACKGROUND DUE TO 239PU
      DO 10 I = 1,100
10      Y(I) = Y(I) - BGFACT * BUFF(I)
      CALL CHFIT (JPK, REFPK, Y)
C      SUBTRACT OUT THE 44.19 AND 46.23 KEV PEAKS
      PHT = ECONST(6) * AREA(4) / C45
      PPOS = REFPK - 13.9
      NS2 = 26
      NE2 = 38
      DO 30 J = 1,2
      DO 20 I = NS2, NE2
      XI = I
      DX = XI - PPOS
      DDX = DX * DX
      ADX = ALFA * DDX
      FXX = EXP (ADX)
      IF (DX) 22,24,24
22      FXX = FXX + EXP1 * EXP(EXP2 * DX) * (1. - EXP (.4 * ADX))
24      Y(I) = Y(I) - PHT * FXX
20      CONTINUE
      PHT = ECONST(7) * PK51 / (C45 * (7.5 * AMTPU + 1.))
      PPOS = REFPK + 12.8
      NS2 = 50
      NE2 = 65
30      CONTINUE
C      INTEGRATE THE 43 KEV PEAK
      NS1 = NE1 - 6
      ME1 = MS1 + 5
      CALL NCTS (NS1, NE1, MS1, ME1, ND, BGD, SUM, A1, A2, Y, BUFF)
      AREA(1) = SUM
      AVEBG = 0.5 * (BG1 + BG2)
      CHANS = MS1 - NE1 - 1
      XPTS = ME1 + NE1 - MS1 - NS1 + 1
      ERR(1) = SQRT (SUM + AVEBG * CHANS * (1. + CHANS / XPTS))
C      INTEGRATE THE 45 KEV PEAK
      NS = NE - 5
      ME = MS + 25
      CALL NCTS (NS, NE, MS, ME, ND, BGD, SUM, A1, A2, Y, BUFF)
      AREA(3) = SUM
      CHANS = MS - NE - 1
      XPTS = ME + NE - MS - NS + 1
      ERR(3) = SQRT (SUM + AVEBG * CHANS * (1. + CHANS / XPTS))
      CALL CHAIN ('REPORT')
      END

```

C REPORT

C THIS SUBPROGRAM IS THE THIRD LINK TO THE PROGRAM "LEPA".

```

COMMON FNAME, GAIN, EXP2, AMTPU, TIME, SHAPC, AREA, ERR, ECONST
DIMENSION SHAPC(9), AREA(4), ERR(4), ECONST(10), XMU(4),
1 DCONST(4), DPM(4), ATOMS(4), SIGSQ(4), FRAC(4), PCT(4)
DCONST(1) = 1.503E-08
DCONST(2) = 5.403E-11
DCONST(3) = 1.977E-10
DCONST(4) = 9.089E-08
C DETERMINE APPROXIMATE AMOUNT OF PU FOR FINAL ATTENUATION
C CORRECTIONS
XMU(1) = 8.0
XMU(2) = 7.2
XMU(3) = 4.9
XMU(4) = 1.32
GMSPU = 0.0
ATNUM = 237.
DO 40 J = 1,4
ATNUM = ATNUM + 1.
GMSPU = GMSPU + AREA(J) *(XMU(J) *AMTPU + 1.)/DCONST(J) *ATNUM
1 * 1.6603E-24 / (ECONST(J) * TIME)
40 CONTINUE
SUM = 0.0
SUMSQ = 0.0
ATNUM = 237.
AMTPU = 0.0
ADPM = 0.0
DO 50 J = 1,4
ATNUM = ATNUM + 1.
ERR(J) = ERR(J) / AREA(J)
DPM(J) = AREA(J) *(XMU(J) *GMSPU + 1.) / (ECONST(J) * TIME)
ADPM = ADPM + DPM(J)
ATOMS(J) = DPM(J) / DCONST(J)
AMTPU = AMTPU + 1.6603E-21 * ATOMS(J) * ATNUM
SIG = ERR(J) * ATOMS(J)
SIGSQ(J) = SIG * SIG
SUMSQ = SUMSQ + SIGSQ(J)
50 SUM = SUM + ATOMS(J)
ADPM = ADPM - DPM(4)
SIG = (100. / SUM) * SQRT (SUMSQ)
DO 60 J = 1,4
FRAC(J) = ATOMS(J) / SUM
SX = SUM - ATOMS(J)
PCT(J) = (100. * SX/SUM)* SQRT(ERR(J) * ERR(J)+ (SUMSQ -
1 SIGSQ(J)) / (SX * SX))
60 ERR(J) = 100. * ERR(J)
WRITE (1,102) FNAME, TIME, AMTPU, SIG, ADPM
102 FORMAT('ANALYSIS OF SAMPLE ', A6, F9.1, ' MINUTES'/'MGMS PU/ML =',
1 F7.3, ' +/-', F6.3, '%/'/'ALPHA DPM =', E11.4/10X,
2 'DPM/ML ATOMS/ML PCTERR ISOTOPIC PCTERR')
WRITE (1,103) DPM(1), ATOMS(1), ERR(1), FRAC(1), PCT(1), DPM(2),
1 ATOMS(2), ERR(2), FRAC(2), PCT(2), DPM(3), ATOMS(3), ERR(3), FRAC(3),
2 PCT(3), DPM(4), ATOMS(4), ERR(4), FRAC(4), PCT(4)
103 FORMAT ('PU238 =', 2E11.4, F6.2, F11.6, F7.3/'PU239 =', 2E11.4,
1 F6.2, F11.6, F7.3/'PU240 =', 2E11.4, F6.2, F11.6, F7.3/'PU241 =',
2 2E11.4, F6.2, F11.6, F7.3)
CALL EXIT
END

```

PROGRAM:

LEPA2

PURPOSE:

To quantitatively analyze a recently processed sample of plutonium.

INPUT:

- 1) Spectrum ID.
- 2) Device on which the spectrum is stored.

CHAIN OF SUBPROGRAMS:

LEPA2→PKS45→PK100→LSFIT→REPR

FILES REQUIRED:

CFILB, LSFILB, PUBKGB

COMMENTS:

This program is similar to LEPA but, in contrast,

uses some of the spectral information available in the 100 keV region. This is particularly useful for obtaining a more precise answer for the ^{241}Pu content, which is quite low in weapons-grade plutonium.

The criterion and initial analysis steps (1 to 7) are the same as those of LEPA. The latter part of the analysis is as follows:

- 8) Analysis of 94 keV x-ray peak.
- 9) A set of equations is formed consisting of the area of the 43, 45, 51, 94, 129, and 148 keV peaks, plus the net counts of the 103.04, 103.684, and 104.243 keV triplet. The matrix coefficients are corrected for self-attenuation effects, and the set of equations is then solved by the method of least-squares, resulting in disintegration rates for the four plutonium isotopes.
- 10) Final calculations are made for atoms, grams, and isotopic abundances, and the results are printed out.

```

C      LEPA2 (FSIO,FLDAT,RDSK,MAXVAL,NDPTS,NCTS,AVE/1)

      COMMON NDPTS,MEQ,FNAME,GAIN,EXP2,AMTPU,TIME,GSQ,SHAPC,AREA,ERR,
1      Y
      DIMENSION SHAPC(9),AREA(6),ERR(6),Y(100)
      READ (1,100) FNAME
100     FORMAT ('SPECTRUM ID IS ',A6)
      READ (1,101) DEV
101     FORMAT ('DEVICE IS ',A6)
      IF (DEV - ' ') 2,4,2
2       CALL FLDAT (DEV,FNAME,'RD')
4       CALL IOPEN ('SYS','CFILB')
      READ (4,102) (SHAPC(I), I = 1,9)
102     FORMAT (A6)
C      ASSUME FOLLOWING APPROXIMATE GAIN
      GAIN = .075
C      OBTAIN 51,129, AND 148 KEV PEAK INTENSITIES
      EN = 51.
      ST = 670.
      NDPTS = 44
      GSQ = GAIN * GAIN
      NJ = 3
      DO 30 J = 1,3
      CALL RDSK ('SYS',ST,NDPTS,'RAYXXX',Y)
      CALL MAXVAL (10,NDPTS,JPK,BIGST,Y)
      SG = (SHAPC(1) + EN * SHAPC(2)) / GSQ + 0.462
      ALFA = -2.7726 / SG
      F12M = SQRT (SG)
      EXP1 = EXP (EN * SHAPC(4) + SHAPC(3))
      EXP2 = SHAPC(5) * GAIN
      SQTY1 = SQRT (Y(1))
      SQTY2 = SQRT (Y(NDPTS))

```

LSFILB File Structure (51 values)

1.	103.03		
2.	103.67	Gamma ray energies	
3.	104.24		
4.			
5.	Same as entries 5, 6, and 7 in LEFILB file		
6.			
7.	0.82		103.03
8.	0.82		103.67
9.	0.81	Attenuation	104.24
10.	8.0	coefficients for	43.492
11.	7.2	gamma rays	45.235
12.	4.9	listed on	51.628
13.	1.05	the right	94.665
14.	1.85		129.28
15.	1.32		148.60

 ^{238}Pu ^{239}Pu ^{240}Pu ^{241}Pu

103.03	16	17 X	18	19
103.67	20	21	22	23 X
104.24	24	25	26 X	27
43.492	28 X	29	30	31
45.235	32	33	34 X	35
51.628	36	37 X	38	39
94.665	40 X	41 X	42 X	43 X
129.28	44	45 X	46	47
148.60	48	49	50	51 X

47

```

      LL = F12M
      L = JPK - LL
      LMT = L - 2
      MS = JPK + LL
C     FIND STARTING AND ENDING CHANNELS OF THE PEAK
      DO 20 I = 1, LMT
      L = L - 1
      DX = L - JPK
      X1 = BIGST * EXP (ALFA * DX * DX)
      X2 = X1 + BIGST * EXP1 * EXP (EXP2 * DX)
      IF (X1 - SQTY2) 14, 13, 13
13     MS = MS + 1
14     IF (X2 - SQTY1) 21, 20, 20
20     CONTINUE
21     NE = L
      NS = NE - 7
      IF (NS) 22, 22, 23
22     NS = 1
23     CALL NCONTS (NS, NE, MS, NDPTS, ND, BGD, SUM, ER, AVEBG, Y, Y)
      ERR(NJ) = ER
      IF (J - 2) 24, 25, 25
24     NJ = 5
      EN = 129.
      ST = 1680.
      NDPTS = 80
      GO TO 30
25     NJ = 5
      EN = 148.
      ST = 1958.
      NDPTS = 62
30     CONTINUE
      CALL CHAIN ('PKS45')
      END

```

```

C     PKS45 (FSIO, RDSK, MAXVAL, AVE, CHFIT, NCTS/I)

```

```

C     THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "LEPA2".

```

```

      COMMON NDPTS, MEQ, FNAME, GAIN, EXP2, AMTPU, TIME, GSQ, SHAPC, AREA, ERR,
1     EN, CONST, ECONST, Y, BUFF
      DIMENSION SHAPC(9), AREA(6), ERR(6), EN(3), CONST(3), ECONST(36),
1     Y(100), BUFF(100)
C     CALL IN 41-47 KEV REGION
      CALL RDSK ('SYS', 560., 100, 'RAYXXX', Y)
      CALL MAXVAL (15, 30, JPK, BIGST, Y)
      GSQ = GAIN * GAIN
      SG = (SHAPC(1) + 45. * SHAPC(2)) / GSQ + 0.462
      ALFA = -2.7726 / SG
      F12M = SQRT (SG)
      EXP1 = EXP (SHAPC(3) + 45. * SHAPC(4))

```

```

C45 = SQRT (1.133 * SG)
BG1 = Y(15)
BG2 = Y(32)
SQTY1 = SQRT (BG1)
SQTY2 = SQRT (BG2)
BIGST = BIGST - BG2
C DETERMINE THE INTEGRATION LIMITS FOR THE 43 AND 45 KEV PEAKS
DO 8 J = 1,2
LL = F12M
L = JPK - LL
LMT = L - 2
MS = JPK + LL
DO 5 I = 1,LMT
L = L - 1
DX = L - JPK
X1 = BIGST * EXP (ALFA * DX * DX)
X2 = X1 + BIGST * EXP1 * EXP (EXP2 * DX)
IF (X1 - SQTY2) 4,3,3
3 MS = MS + 1
4 IF (X2 - SQTY1) 6,5,5
5 CONTINUE
6 IF (J-1) 7,7,9
7 NE1 = L
MS1 = MS
SQTY1 = SQTY2
CALL MAXVAL (38,52,JPK,BIGST,Y)
BIGST = BIGST - BG2
8 CONTINUE
9 NE = L
CALL AVE (MS,80,SUM,BG3,Y)
C OBTAIN ESTIMATE OF AMOUNT PU PRESENT
CALL IOPEN ('SYS','LSFILE')
READ (4,102) (EN(I), I = 1,3)
READ (4,102) (CONST(I), I = 1,3)
READ (4,102) (BUFF(I), I = 1,9)
READ (4,102) (ECONST(I), I = 1,36)
102 FORMAT (A6)
AMTPU = 7.38E-12 * AREA(2)/ECONST(22) + 4.38E-15 * AREA(6)/
1 ECONST(36) + 2.00E-12 * C45 * BIGST / ECONST(19)
CALL RDSK ('SYS',1,1,'RAYXXX',TIME)
TIME = TIME / 60.
AMTPU = AMTPU / TIME
C APPLY SELF-ABSORPTION CORRECTIONS.
DO 12 J = 1,9
CORR = 1. - AMTPU * BUFF(J)
DO 13 I = 1,4
13 ECONST(I,J) = ECONST(I,J) * CORR
12 CONTINUE
PK51 = (5.1 * AMTPU + 1.) * AREA(2)
BGFACT = CONST(1) * PK51
C READ IN FILE DELINEATING THE SHAPE OF THE BACKGROUND CONTINUUM IN
C THE 45 KEV REGION DUE TO 239PU.
CALL IOPEN ('SYS','PUBKGB')
READ (4,102) (BUFF(I), I = 1,100)

```

```

C      SUBTRACT BACKGROUND DUE TO 239PU
      DO 10 I = 1,100
10      Y(I) = Y(I) - BGFACT * BUFF(I)
      CALL CHFIT (JPK, REFFK, Y)
C      SUBTRACT OUT THE 44.19 AND 46.23 KEV PEAKS
      PHT = CONST(2) * AREA(4) / C45
      PPOS = REFFK - 13.9
      NS2 = 26
      NE2 = 38
      DO 30 J = 1,2
      DO 20 I = NS2, NE2
      XI = I
      DX = XI - PPOS
      DOX = DX * DX
      ADX = ALFA * DOX
      FXX = EXP (ADX)
      IF (DX) 22, 24, 24
22      FXX = FXX + EXP1 * EXP(EXP2 * DX) * (1. - EXP (.4 * ADX))
24      Y(I) = Y(I) - PHT * FXX
20      CONTINUE
      PHT = CONST(3) * PK51 / (C45 * (7.5 * AMTPU + 1.))
      PPOS = REFFK + 12.8
      NS2 = 50
      NE2 = 65
30      CONTINUE
C      INTEGRATE THE 43 KEV PEAK
      NS1 = NE1 - 6
      ME1 = MS1 + 5
      CALL NCTS (NS1, NE1, MS1, ME1, ND, BGD, SUM, A1, A2, Y, BUFF)
      AREA(1) = SUM
      AVEBG = 0.5 * (BG1 + BG2)
      CHANS = MS1 - NE1 - 1
      XPTS = ME1 + NE1 - MS1 - NS1 + 1
      ERR(1) = SQRT (SUM + AVEBG * CHANS * (1. + CHANS / XPTS))
C      INTEGRATE THE 45 KEV PEAK
      NS = NE - 5
      ME = MS + 25
      CALL NCTS (NS, NE, MS, ME, ND, BGD, SUM, A1, A2, Y, BUFF)
      AREA(2) = SUM
      CHANS = MS - NE - 1
      XPTS = ME + NE - MS - NS + 1
      ERR(2) = SQRT (SUM + AVEBG * CHANS * (1. + CHANS / XPTS))
      CALL CHAIN ('PK100')
      END

```

```

C      PK100 (FSIO, RDSK, MAXVAL, CHFIT, BOUNDS, STRIP, NCNTS, NCTS, AVE)

C      THIS SUBPROGRAM IS THE THIRD LINK TO THE PROGRAM "LEPA2"

      COMMON NDPTS, MEQ, FNAME, GAIN, EXP2, AMTPU, TIME, GSO, SHAPC, AREA, ERR,
1      EN, CONST, ECONST, Y, YNET, ACOEF, BCOEF
      DIMENSION SHAPC(9), AREA(6), ERR(6), EN(3), CONST(3), ECONST(4, 9),
1      Y(100), YNET(50), ACOEF(50, 4), BCOEF(50, 4), PKPOS(3)

```

```

C   ANALYZE 94 KEV PEAK
    SG = (SHAPC(1) + 103. * SHAPC(2)) / GSO + 0.462
    ALFA = -2.7726 / SG
    F12M = SQRT (SG)
    EXP1 = EXP (SHAPC(3) + 103. * SHAPC(4))
    C1 = 1. / (GAIN * F12M)
    C1 = 0.3 * C1 * C1 / 1.05
    C2 = -2.44 / F12M
    NDPTS = 100
    CALL RDSK ('SYS', 1200., NDPTS, 'RAYXXX', Y)
    CALL MAXVAL (40, 80, JPK, BIGST, Y)
    CALL CHFIT (JPK, REFCH, Y)
    GAIN = 94.665 / (REFCH + 1199.)
    LL = F12M
    MS = JPK
    BIGST = BIGST - Y(20)
    SQTY = SQRT (Y(20) * 1.5)
    CALL BOUNDS (LL, MS, BIGST, ALFA, EXP1, EXP2, C1, C2, SQTY, SQTY)
    NS = LL - 5

C   STRIP OUT 97 KEV LEADING EDGE FROM BKGRD.
    JPK97 = JPK + 32
    PKPS = JPK97
    JS = JPK - 30
    PKHT = Y(JPK97) - Y(JS)
    ME = MS + 5
    CALL STRIP (MS, ME, PKPS, PKHT, ALFA, EXP1, EXP2, C1, C2, Y)
    CALL NCONTS (NS, LL, MS, ME, NDPTS, BGD, SUM, ER, AVEBG, Y, Y)
    AREA(4) = SUM
    ERR(4) = ER

C   ISOLATE THE 103-104 KEV TRIPLET
    START = 1280 + JPK
    NDPTS = 90
    CALL RDSK ('SYS', START, NDPTS, 'RAYXXX', Y)

C   STRIP OUT 101 HIGH SIDE TAILING FROM BKGRD.
    JP101 = 5
    NE = JP101 + 20
    PKPS = JP101
    PKHT = Y(JP101) - Y(NE)
    NS = NE - 10
    CALL STRIP (NS, NE, PKPS, PKHT, ALFA, EXP1, EXP2, C1, C2, Y)
    CALL NCTS (NS, NE, 58, 80, NDPTS, BGD, SUM, AVBG1, AVBG2, Y, YNET)
    AVEBG = 0.5 * (AVBG1 + AVBG2)
    START = START + FLOAT (NE - 2)
    RATIO = 0.989 / SHAPC(6)
    DO 6 J = 1, 3
6    PKPOS(J) = EN(J) / GAIN - START
    DO 20 I = 1, NDPTS
    WTF = 1. / SQRT (YNET(I) + AVEBG)
    XI = I
    DO 8 K = 1, 4
8    ACOEF(I, K) = 0.0
    DO 18 J = 1, 3
    DX = XI - PKPOS(J)
    DDX = DX * DX
    ADX = ALFA * DDX
    IF (ADX + 20.) 18, 10, 10

```



```

10      FXX = EXP (ADX)
      EXPN3 = EXP (0.4 * ADX)
      IF (DX) 12,14,14
12      EXPN2 = EXP (EXP2 * DX)
      FXX = FXX + EXP1 * EXPN2 * (1. - EXPN3)
14      DO 17 K = 2,4
      IF (ECONST(K,J)) 16,17,16
16      ACOEF(I,K) = ACOEF(I,K) + ECONST(K,J) * RATIO * FXX
17      CONTINUE
18      CONTINUE
      Y(I) = YNET(I) * WTF
      DO 19 K = 1,4
      ACOEF(I,K) = ACOEF(I,K) * WTF
19      BCOEF(I,K) = ACOEF(I,K)
20      CONTINUE
      MEQ = NDPTS + 6
      L = NDPTS + 1
      M = 3
      J = 0
      DO 32 I = L,MEQ
      M = M + 1
      J = J + 1
      Y (I) = AREA(J) / ERR (J)
      YNET(I) = AREA(J)
      DO 30 K = 1,4
      ACOEF(I,K) = ECONST(K,M) / ERR(J)
30      BCOEF(I,K) = ACOEF(I,K)
32      CONTINUE
      CALL CHAIN ('LSFIT')
      END

```

C LSFIT (MLR,MLRS)

C THIS SUBPROGRAM IS THE FOURTH LINK TO THE PROGRAM "LEPA2".

```

      COMMON NDPTS,MEQ,AWRDS,Y,YNET,ACOEf,BCOEf,ANS,RM,VM,WM
      DIMENSION AWRDS(69),Y(100),YNET(50),ACOEf(50,4),BCOEf(200),
1  ANS(4),RM(50),VM(4),WM(4)
      NERR = 1
      CALL MLR (50,MEQ,4,Y,ANS,BCOEf,RM,WM,VM,NERR)
      IF (NERR - 2) 2,4,2
2      CALL MLRS (50,MEQ,4,ACOEf,Y,ANS,BCOEf,RM,WM,VM,NERR)
4      GO TO (5,6,7) NERR
6      WRITE (1,101)
101     FORMAT ('MATRIX IS SINGULAR')
      GO TO 10
7      WRITE (1,102)
102     FORMAT ('SUM OF SQUARES OF R > THAN THOSE OF Y')
10      CALL EXIT
5      CALL CHAIN ('REPRT')
      END

```

C REPRT (MLRE)

C THIS SUBPROGRAM IS THE FIFTH LINK TO THE PROGRAM "LEPA2".

```
COMMON NDPTS, MEQ, FNAME, GAIN, EXP2, AMTPU, TIME, GSO, SHAPC, AREA, ERR,
1  AWRDS, Y, YNET, ACOEF, BCOEF, ANS, RM, VM, WM, DM
  DIMENSION SHAPC(9), AREA(6), ERR(6), AWRDS(42), Y(100), YNET(50),
1  ACOEF(200), BCOEF(200), ANS(4), RM(50), VM(4), WM(4), DM(4),
2  DCONST(4), ATOMS(4), SIGSQ(4), FRAC(4), PCT(4)
  DCONST(1) = 1.503E-08
  DCONST(2) = 5.403E-11
  DCONST(3) = 1.997E-10
  DCONST(4) = 9.089E-08
  CALL MLRE (DM, 50, 4, BCOEF, WM)
  SUM = 0.0
  DO 30 I = 1, MEQ
30  SUM = SUM + RM(I) **2
  DOF = MEQ - 5
  QFIT = SUM / DOF
  SUM = 0.0
  SUMSG = 0.0
  ATNUM = 237.
  AMTPU = 0.0
  ADPM = 0.0
  DO 50 J = 1, 4
  ATNUM = ATNUM + 1.
  ERR(J) = SQRT (QFIT * DM(J) ) / ANS(J)
  ANS(J) = ANS(J) / TIME
  ADPM = ADPM + ANS(J)
  ATOMS(J) = ANS(J) / DCONST(J)
  AMTPU = AMTPU + 1.6603E-21 * ATOMS(J) * ATNUM
  SIG = ERR(J) * ATOMS(J)
  SIGSQ(J) = SIG * SIG
  SUMSG = SUMSG + SIGSQ(J)
50  SUM = SUM + ATOMS(J)
  ADPM = ADPM - ANS(4)
  SIG = (100. / SUM) * SQRT (SUMSG)
  DO 60 J = 1, 4
  FRAC(J) = ATOMS(J) / SUM
  SX = SUM - ATOMS(J)
  PCT(J) = (100. * SX / SUM) * SQRT (ERR(J) * ERR(J) + (SUMSG -
1  SIGSQ(J)) / (SX * SX))
60  ERR(J) = 100. * ERR(J)
  WRITE (1, 102) FNAME, TIME, AMTPU, SIG, QFIT, ADPM
102  FORMAT('ANALYSIS OF SAMPLE ', A6, F9.1, ' MINUTES'// 'MGMS PU/ML =',
1  F7.3, ' +/-', F6.3, '%', 'QFIT =', F5.2, 'ALPHA DPM =', E11.4/10X,
2  'DPM/ML    ATOMS/ML    PCTERR    ISOTOPIC PCTERR')
  WRITE (1, 103) ANS(1), ATOMS(1), ERR(1), FRAC(1), PCT(1), ANS(2),
1  ATOMS(2), ERR(2), FRAC(2), PCT(2), ANS(3), ATOMS(3), ERR(3), FRAC(3),
2  PCT(3), ANS(4), ATOMS(4), ERR(4), FRAC(4), PCT(4)
103  FORMAT ('PU238 =', 2E11.4, F6.2, F11.6, F7.3/'PU239 =', 2E11.4,
1  F6.2, F11.6, F7.3/'PU240 =', 2E11.4, F6.2, F11.6, F7.3/'PU241 =',
2  2E11.4, F6.2, F11.6, F7.3)
  IF (QFIT - 1.5) 64, 62, 62
```

C REPRT (MLRE)

C THIS SUBPROGRAM IS THE FIFTH LINK TO THE PROGRAM "LEPA2".

```
COMMON NDPTS, MEQ, FNAME, GAIN, EXP2, AMTPU, TIME, GSG, SHAPC, AREA, ERR,
1 AWRDS, Y, YNET, ACOEF, BCOEF, ANS, RM, VM, WM, DM
  DIMENSION SHAPC(9), AREA(6), ERR(6), AWRDS(42), Y(100), YNET(50),
1 ACOEF(200), BCOEF(200), ANS(4), RM(50), VM(4), WM(4), DM(4),
2 DCONST(4), ATOMS(4), SIGSQ(4), FRAC(4), PCT(4)
  DCONST(1) = 1.503E-08
  DCONST(2) = 5.403E-11
  DCONST(3) = 1.997E-10
  DCONST(4) = 9.089E-08
  CALL MLRE (DM, 50, 4, BM, WM)
  SUM = 0.0
  DO 30 I = 1, MEQ
30 SUM = SUM + RM(I) **2
  DOF = MEQ - 5
  QFIT = SUM / DOF
  SUM = 0.0
  SUMSG = 0.0
  ATNUM = 237.
  AMTPU = 0.0
  ADPM = 0.0
  DO 50 J = 1, 4
  ATNUM = ATNUM + 1.
  ERR(J) = SQRT (QFIT * DM(J) ) / ANS(J)
  ANS(J) = ANS(J) / TIME
  ADPM = ADPM + ANS(J)
  ATOMS(J) = ANS(J) / DCONST(J)
  AMTPU = AMTPU + 1.6603E-21 * ATOMS(J) * ATNUM
  SIG = ERR(J) * ATOMS(J)
  SIGSQ(J) = SIG * SIG
  SUMSG = SUMSG + SIGSQ(J)
50 SUM = SUM + ATOMS(J)
  ADPM = ADPM - ANS(4)
  SIG = (100. / SUM) * SQRT (SUMSG)
  DO 60 J = 1, 4
  FRAC(J) = ATOMS(J) / SUM
  SX = SUM - ATOMS(J)
  PCT(J) = (100. * SX / SUM) * SQRT (ERR(J) * ERR(J) + (SUMSG -
1 SIGSQ(J)) / (SX * SX))
60 ERR(J) = 100. * ERR(J)
  WRITE (1, 102) FNAME, TIME, AMTPU, SIG, QFIT, ADPM
102 FORMAT('ANALYSIS OF SAMPLE ', A6, F9.1, ' MINUTES'// 'MGMS PU/ML =',
1 F7.3, ' +/-', F6.3, '% '// 'QFIT =', F5.2 // 'ALPHA DPM =', E11.4 / 10X,
2 'DPM/ML ATOMS/ML PCTERR ISOTOPIC PCTERR')
  WRITE (1, 103) ANS(1), ATOMS(1), ERR(1), FRAC(1), PCT(1), ANS(2),
1 ATOMS(2), ERR(2), FRAC(2), PCT(2), ANS(3), ATOMS(3), ERR(3), FRAC(3),
2 PCT(3), ANS(4), ATOMS(4), ERR(4), FRAC(4), PCT(4)
103 FORMAT ('PU238 =', 2E11.4, F6.2, F11.6, F7.3 // 'PU239 =', 2E11.4,
1 F6.2, F11.6, F7.3 // 'PU240 =', 2E11.4, F6.2, F11.6, F7.3 // 'PU241 =',
2 2E11.4, F6.2, F11.6, F7.3)
  IF (QFIT - 1.5) 64, 62, 62
62 WRITE (1, 104) (I, YNET(I), RM(I), I = 1, MEQ)
104 FORMAT (/ 7X, 'YNET STD DEV' / (I2, F10.1, F8.2))
64 CALL EXIT
  END
```

PROGRAM:

PUB

PURPOSE:

To quantitatively analyze a recently processed sample of plutonium.

INPUT:

- 1) Spectrum ID.
- 2) Device on which the spectrum is stored.

CHAIN OF PROGRAMS:

PUB→FIT3→PMLR3

PMLA3→MATRX3→XMLR3

XMLA3→OUT3

FILES REQUIRED:

CFILEB, EFILEB, EFACTB

COMMENTS:

This is one of the first programs used for measuring the isotopic and total analysis of plutonium solutions in a reprocessing plant. It assumes that the ^{241}Am has

been removed and the ^{237}U has not grown back into an appreciable extent. The sequence of the analysis and the peaks used is as follows:

- 1) Input of shape constants, gain, etc. from "CFILEB" disk file. (See pg. 3)
- 2) Analysis of 152.77 keV peak of ^{238}Pu .
- 3) Analysis of 129.28 keV peak of ^{239}Pu .
- 4) Analysis of 148.60 keV peak of ^{241}Pu .
- 5) Analysis of 94.665 keV x-ray peak.
- 6) Analysis of a multiplet consisting of 97.03, 98.446, 98.770, 99.87, and 101.07 keV peaks. The combined intensities of the 98.446 and 98.770 peaks and the intensity of the 99.87 keV peak are preserved.
- 7) A matrix of equations is formed consisting of areas of the six peaks listed above plus the net counts of the 103.04, 103.684, and 104.243 triplet. The least-squares solution of this set of equations yields the disintegration rates of the isotopes of interest, i.e., 238, 239, 240, 241Pu.
- 8) Final calculations are performed to yield atoms, grams, and isotopic percentages, etc., and the results are printed out.

C PUB (FSID, FLDAT, RISK, MAXVAL, CHFIT, NCNTS, NCTS, BOUNDS, AVE/I)

```

COMMON K, KK, NDPTS, NP, ITER, NERR, NWRD, NFLG1, NFLG2, NFLG3,
1 NYFLG, NXFLG, FNAME, GAIN, ZERO, START, REFFK, ALFA, EXP1, EXP2,
2 C100, XC100, C1, C2, AVEBG, SHAPC, EN, AREA, ERR, Y,
3 PKPOS, PKHT
DIMENSION NYFLG(10), NXFLG(10), SHAPC(5), EN(13),
1 AREA(6), ERR(6), Y(100), PKPOS(10), PKHT(10)
READ (1, 100) FNAME
100 FORMAT ('SPECTRUM ID IS ', A6)
READ (1, 101) DEV
101 FORMAT ('DEVICE IS ', A6)
IF (DEV - ' ') 5, 9, 5
5 CALL FLDAT (DEV, FNAME, 'RD')
9 CALL IOPEN ('SYS', 'CFILEB')
READ (4, 105) (SHAPC(I), I = 1, 9)
105 FORMAT (A6)
C100 = SHAPC(6)
XC100 = SHAPC(7)
GAIN = SHAPC(8)
ZERO = SHAPC(9)
CALL IOPEN ('SYS', 'EFILEB')
READ (4, 105) (EN(I), I = 1, 13)
C1 = 0.0
C ANALYZE THE 152, 129, 148 AND 94 KEV PEAKS. THEN GET 97-101 KEV
C PEAK REGION INTO CORE.
DO 30 J = 1, 5
IF (J-5) 1, 4, 1
1 IF (J-3) 50, 3, 2

```

EFIEB File Structure (13 values)

1. 152.77
2. 129.28
3. 148.60
4. 94.665
5. 98.440
6. 97.10
7. 98.446
8. 98.770
9. 99.87
10. 101.07
11. 103.03
12. 103.67
13. 104.24

EFACTB File Structure (45 values)

Entries are in units of counts/disintegration. No entry equals zero.

	²³⁸ Pu	²³⁹ Pu	²⁴⁰ Pu	²⁴¹ Pu
103.03	1	2 X	3	4
103.67	5	6	7	8 X
104.23	9	10	11 X	12
152.77	13 X	14	15	16
129.28	17	18 X	19	20
148.60	21	22	23	24 X
94.665	25 X	26 X	27 X	28 X
98.446	29 X	30 X	31 X	32 X
98.77	33 X	34	35	36
99.87	37 1.65	38 1.64	39 1.63	40 2.50
	41 3.70	42 2.65	43 2.10	44 1.9
	45 1.80			

Entries 37 to 45 are attenuation coefficients for the gamma rays listed at the left.

```

3      ST1 = 1.8
      XDT = 4.65
      GO TO 4
50     XDT = 5.1
2      ST1 = 3.
4      REFCR = EN(J) / GAIN + ZERO
      ST = REFCR - ST1 / GAIN
      NDPTS = XDT / GAIN
      CALL RDSK ('SYS', ST, NDPTS, 'RAYXXX', Y)
      CALL MAXVAL (1, NDPTS, JPK, BIGST, Y)
      SG = SHAPC(1) / (GAIN * GAIN) + SHAPC(2) * REFCR / GAIN + 462
      ALFA = -2.7726 / SG
      F12M = SQRT (SG)
      EXP1 = EXP (EN(J) * SHAPC(4) + SHAPC(3))
      EXP2 = SHAPC(5) * GAIN
      SQTY1 = SQRT (Y(1))
      SQTY2 = SQRT (Y(NDPTS))
      LL = F12M
      MS = JPK
      IF (J - 4) 8, 6, 7
6      ST1 = 2.7
      XDT = 6.9
      CALL CHFIT (JPK, REFCR, Y)
      NX = ST
      ST = FLOAT (NX)
      PK94 = REFCR + ST * -1.
      GAIN = 94.665 / (PK94 - ZERO)
7      C1 = 1.0 / (GAIN * F12M)
      C1 = 0.3 * C1 * C1
      C2 = -2.44 / F12M
      IF (J - 5) 8, 40, 8
8      CALL BOUNDS (LL, MS, BIGST, ALFA, EXP1, EXP2, C1, C2, SQTY1, SQTY2)
      NS = NE - 7
      IF (NS) 17, 17, 18
17     NS = 1
18     CALL NCNTS (NS, NE, MS, NDPTS, ND, BGD, SUM, ER, AVEBG, Y, Y)
      AREA (J) = SUM
      ERR(J) = ER
30     CONTINUE
40     NS1 = 10
      NE1 = NDPTS - 9
      START = ST + FLOAT (NDPTS)
      CALL NCNTS (1, NS1, NE1, NDPTS, NDPTS, BGD, SUM, AVEBG1, AVEBG2, Y, Y)
      AVEBG = 0.5 * (AVEBG1 + AVEBG2)
      NX = ST
      ST = NX + NS1
      DO 42 I = 1, 5
      L = I + 5
      NXFLG(I) = 0
      NYFLG(I) = 1
      PKPOS(I) = EN(L) / GAIN - ST + ZERO + 1.
42     CONTINUE
      NYFLG(1) = 2
      NYFLG(2) = 2
      NYFLG(5) = 2
      NP = 5
      CALL CHAIN ('FIT3')
      END

```

C FIT3

C THIS SUBPROGRAM IS THE SECOND LINK TO THE PROGRAM "PUB".

```
COMMON NWRD1, NWRD2, NDPTS, NP, NWRDS, NYFLG, NXFLG, FNAME, GAIN,
1 ZERO, START, REFPK, ALFA, EXP1, EXP2, C100, XC100, C1, C2, AVEBG, SHAFC,
2 EN, AREA, ERR, YNET, PKPOS, PKHT, CPOS, Y, RM, VM, ACDEF
DIMENSION NWRDS(6), NYFLG(10), NXFLG(10), SHAFC(5), EN(13),
1 AREA(6), ERR(6), YNET(100), PKPOS(10), PKHT(10), CPOS(10),
2 Y(100), RM(100), VM(10), ACDEF(100, 8)
DO 30 I = 1, NDPTS
  WT = 1. / SQRT (YNET(I) + AVEBG)
  XI = I
  DO 20 J = 1, NP
    FXX = 0.0
    DX = XI - PKPOS(J)
    DDX = DX * DX
    ADX = ALFA * DDX
    EXPN3 = 0.0
    IF (ADX + 12.) 5, 4, 4
4    FXX = EXP (ADX)
    EXPN3 = EXP (.4 * ADX)
    IF (DX) 6, 9, 9
    5    EXPN = EXP2 * DX
    IF (EXPN + 9.) 7, 8, 8
    6    EXPN2 = EXP (EXPN)
    FXX = FXX + EXP1 * EXPN2 * (1. - EXPN3)
    7    DX = -DX
    IF (NYFLG(J) - 2) 14, 10, 14
    8    CXPN = C2 * DX
    IF (CXPN + 12.) 14, 12, 12
    9    FXX = FXX + C1 * (1.05 - EXPN3) * EXP (CXPN)
    10    IF (J - 5) 18, 16, 18
    11    FXX = FXX * 1.72
    ACDEF(I, 1) = ACDEF(I, 1) + FXX * WT
    GO TO 20
    12    ACDEF(I, J) = FXX * WT
    13    CONTINUE
    Y(I) = YNET(I) * WT
    14    CONTINUE
    15    NP = 4
    CALL CHAIN ('PMLR3')
  END
```

C PMLR3 (MLR/O)

C THIS SUBPROGRAM IS THE THIRD LINK TO THE PROGRAM "PUB".

```
COMMON NWR1, NWR2, NDPTS, NP, NWRD3, NERR, NWRDS, AWRDS, YNET, PKPOS,
1 PKHT, CPOS, Y, RM, VM, ACDEF, WM
DIMENSION NWRDS(24), AWRDS(43), YNET(100), PKPOS(10), PKHT(10),
```

```

1  CPOS(10), Y(100), RM(100), VM(10), ACOEF(100, 8), WM(10)
  NERR = 1
  CALL OOPEN ('DSK1', 'AMTRX')
  NM = 100 * NP
  WRITE (4, 100) (ACOEF(I), I = 1, NM)
100 FORMAT (10A6)
  CALL OCLOSE
  CALL MLR (100, NDPTS, NP, Y, PKHT, ACOEF, RM, WM, VM, NERR)
  CALL CHAIN ('PMLA3')
  END

```

C PMLA3 (MLRA, MLRE/I)

C THIS SUBPROGRAM IS THE FOURTH LINK TO THE PROGRAM "PUB".

```

COMMON NWR1, NWR2, NDPTS, NP, NWR3, NERR, NWRDS, AWRDS, C100, XC100,
1  BWRDS, AREA, ERR, YNET, PKPOS, PKHT, CPOS, Y, RM, DM, BM, WM
  DIMENSION NWRDS(24), AWRDS(8), BWRDS(21), AREA(6), ERR(6), YNET(100),
1  PKPOS(10), PKHT(10), CPOS(10), Y(100), RM(100), DM(10), BM(800), WM(10)
  IF (NERR - 2) 2, 4, 2
2  CALL MLRA (100, NDPTS, NP, Y, PKHT, BM, RM, WM, DM, NERR)
4  GO TO (5, 6, 7) NERR
6  WRITE (1, 101)
101 FORMAT ('MATRIX IS SINGULAR')
  GO TO 10
7  WRITE (1, 102)
102 FORMAT ('SUM OF SQUARES OF R > THAN THOSE OF Y')
10  CALL EXIT
5  K = NP
  CALL MLRE (DM, 100, K, BM, WM)
  AREA(5) = C100 * (PKHT(2) * XC100 + PKHT(3))
  AREA(6) = C100 * C100
  ERR(5) = 2. * C100 * SQRT (DM(2))
  ERR(6) = 2. * C100 * SQRT (DM(4))
  CALL CHAIN ('MATRX3')
  END

```

C MATRX3 (FSIO, RDSK, MAXVAL, NCTS, AVE/I)

C THIS SUBPROGRAM IS THE FIFTH LINK TO THE PROGRAM "PUB".

```

COMMON K, MEQ, NDPTS, NP, ITER, NERR, NWRDS, FNAME, GAIN, ZERO,
1  START, TIME, ALFA, EXP1, EXP2, C100, XC100, C1, C2, AVEBG, SHAPC,

```



```

2  EN, AREA, ERR, YNET, PKPOS, PKHT, WRDS, Y, ANS, EFACT, DM, ACOEF
   DIMENSION NWRDS(24), SHAPC(5), EN(13), AREA(4), ERR(6), YNET(100),
1  PKPOS(10), PKHT(10), WRDS(10), Y(100), ANS(10), EFACT(4,25),
2  DM(10), ACOEF(100,7)
   CALL RDSK ('SYS', 1, 1, 'RAYXXX', TIME)
   REFCH = 51.628 / GAIN + ZERO
   ST = REFCH - 10.
   CALL RDSK ('SYS', ST, 20, 'RAYXXX', Y)
   CALL MAXVAL (1, 20, JPK, BIGST, Y)
C  CALCULATE CHANCE COINCIDENCE PILEUP USING A RESOLVING TIME OF
C  5.5 MICROSECONDS.
   CHANC = 1.1 E-05 * BIGST * BIGST / TIME
   NS1 = 4
   ND = START
   XS = ND - 1
   NDPTS = 4.0 / GAIN
   START = START - FLOAT (NS1)
   CALL RDSK ('SYS', START, NDPTS, 'RAYXXX', Y)
   NE1 = NDPTS - 19 + NS1
   CALL NCTS (1, NS1, NE1, NDPTS, NDPTS, BGD, SUM, AVBG1, AVBG2, Y, YNET)
   AVEBG = 0.5 * (AVBG1 + AVBG2)
   DO 4 J = 1,3
   L = J + 10
4  PKPOS(J) = EN(L) / GAIN - XS + ZERO
   NP = 3
   GSQ = 1.0 / (GAIN * GAIN)
   SG = SHAPC(1) * GSQ + SHAPC(2) * 104. * GSQ
   ALFA = -2.7726 / SG
   EXP1 = EXP (SHAPC(3) + 104. * SHAPC(4))
   EXP2 = SHAPC(5) * GAIN
   RATIO = 0.989 / C100
   CHNHT = CHANC * RATIO
   CHPOS = 103.256 / GAIN - XS + ZERO
   TIME = TIME / 60.
   CALL IOFEN ('SYS', 'EFACTB')
101 READ (4,101) (EFACT(I), I = 1,36)
   FORMAT (A6)
   READ (4,101) (DM(I), I = 1,9)
   AMTPU = 3.95E-12 * AREA(2) / (TIME * EFACT(2,5))
   DO 7 J = 1,9
   CORR = 1 - AMTPU * DM(J)
   DO 6 I = 1,4
6  EFACT(I,J) = EFACT(I,J) * CORR
7  CONTINUE
C  CALCULATE MATRIX COEFFICIENTS AND FORM SET OF LINEAR EQUATIONS.
   DO 20 I = 1,NDPTS
   WTF = 1.0 / SQRT (YNET(I) + AVEBG)
   XI = I
   DO 8 K = 1,4
8  ACOEF(I,K) = 0.0
   DO 18 J = 1,NP
   DX = XI - PKPOS(J)
   DDX = DX * DX
   ADX = ALFA * DDX
   IF (ADX + 20.) 18,10,10
10  FXX = EXP (ADX)

```

```

EXP3 = EXP (0.4 * ADX)
IF (DX) 12,14,14
12 EXP2 = EXP (EXP2 * DX)
   FXX = FXX + EXP1 * EXP2 * (1. - EXP3)
14 DO 17 K = 2,4
   IF (EFACT(K,J)) 16,17,16
16 ACOEF(I,K) = ACOEF(I,K) + EFACT(K,J) * RAT10 * FXX
17 CONTINUE
18 CONTINUE
   DX = XI - CHPOS
   ADX = ALFA * DX * DX
   IF (ADX + 10.) 22,21,21
21 YNET(I) = YNET(I) - CHNHT * EXP (ADX)
22 Y(I) = YNET(I) * WTF
   DO 19 K = 1,4
19 ACOEF(I,K) = ACOEF(I,K) * WTF
20 CONTINUE
   MEQ = NDPTS + 6
   L = NDPTS + 1
   M = 3
   J = 0
   DO 32 I = L,MEQ
   M = M + 1
   J = J + 1
   Y(I) = AREA(J) / ERR(J)
   YNET(I) = AREA(J)
   DO 30 K = 1,4
30 ACOEF(I,K) = EFACT(K,M) / ERR(J)
32 CONTINUE
   K = 4
   CALL CHAIN('XMLR3')
END

```

C XMLR3 (MLR/O)

C THIS SUBPROGRAM IS THE SIXTH LINK TO THE PROGRAM "PUB".

```

COMMON K, MEQ, NDPTS, NP, ITER, NERR, NWRDS, AWRDS, YNET, BWRDS, Y,
1  ANS, RM, VM, ACOEF, WM
   DIMENSION NWRDS(24), AWRDS(43), YNET(100), BWRDS(30), Y(100),
1  ANS(10), RM(100), VM(10), ACOEF(100,7), WM(10)
   NERR = 1
   CALL OPEN ('DSK1', 'AMTRX')
   NM = 100 * K
   WRITE (4,100) (ACOE(I), I = 1,NM)
100 FORMAT (10A6)
   CALL OCLOSE
   CALL MLR (100, MEQ, K, Y, ANS, ACOEF, RM, WM, VM, NERR)
   CALL CHAIN ('XMLA3')
END

```

C XMLA3 (MLRA/I)

C THIS SUBPROGRAM IS THE SEVENTH LINK TO THE PROGRAM "PUB".

```
COMMON K, MEQ, NDPTS, NP, ITER, NERR, NWRDS, AWRDS, YNET, BWRDS, Y,
1  ANS, RM, VM, BM, WM
  DIMENSION NWRDS(24), AWRDS(43), YNET(100), BWRDS(30), Y(100),
1  ANS(10), RM(100), VM(10), BM(700), WM(10)
  IF (NERR -2) 2,4,2
2  CALL MLRA (100, MEQ, K, Y, ANS, BM, RM, WM, VM, NERR)
4  GO TO (5,6,7) NERR
6  WRITE (1,101)
101 FORMAT ('MATRIX IS SINGULAR')
  GO TO 10
7  WRITE (1,102)
102 FORMAT ('SUM OF SQUARES OF R > THAN THOSE OF Y')
10  CALL EXIT
5  CALL CHAIN ('OUT3')
  END
```

C OUT3 (MLRE)

C THIS SUBPROGRAM IS THE EIGHTH LINK TO THE PROGRAM "PUB".

```
COMMON K, MEQ, NDPTS, MWRDS, FNAME, GAIN, ZERO, START, TIME, AWRDS,
1  ERR, YNET, BWRDS, ANS, RM, DM, BM, WM
  DIMENSION MWRDS(27), AWRDS(32), ERR(6), YNET(100), BWRDS(130),
1  ANS(10), RM(100), DM(10), BM(700), WM(10), PCT(5), SIGSQ(5), FRAC(5)
  DIMENSION TIHLF(5), ATOMS(5)
  TIHLF(1) = 4.613 E07
  TIHLF(2) = 1.283 E10
  TIHLF(3) = 3.471 E09
  TIHLF(4) = 7.626 E06
  CALL MLRE (DM, 100, K, BM, WM)
  SUM = 0.0
  DO 40 I = 1, MEQ
40  SUM = SUM + RM(I) **2
  DOF = MEQ - 5
  QFIT = SUM / DOF
  SUM = 0.0
  ATNUM = 237.
  GRMS = 0.0
  SUMSQ = 0.0
  ADPM = 0.0
  DO 50 J = 1, 4
  ATNUM = ATNUM + 1.
  ERR(J) = SQRT (QFIT * DM(J)) / ANS(J)
  ANS(J) = ANS(J) / TIME
  ADPM = ADPM + ANS(J)
```

```

      ATOMS(J) = ANS(J) * T1HLF(J) / .693147
      GRMS = GRMS + 1.6603 E-21 * ATOMS(J) * ATNUM
      SIG = ERR(J) * ATOMS(J)
      SIGSQ(J) = SIG * SIG
      SUMSQ = SUMSQ + SIGSQ(J)
50    SUM = SUM + ATOMS(J)
      ADPM = ADPM - ANS(4)
      SIG = (100./SUM) * SQRT (SUMSQ)
      DO 52 J = 1,4
      FRAC(J) = ATOMS(J) / SUM
      SX = SUM - ATOMS(J)
      PCT(J)=(100.*SX/SUM)*SQRT(ERR(J)*ERR(J)+(SUMSQ-SIGSQ(J))
1    / (SX * SX))
52    ERR(J) = 100. * ERR(J)
      WRITE (1,102) FNAME, TIME, GRMS, SIG, QFIT, ADPM
102   FORMAT('ANALYSIS OF SAMPLE ',A6,F9.1,' MINUTES'///'MGMS PU/ML =',
1     F7.3,' +/-',F6.3,'%''/QFIT =',F5.2,' ALPHA DPM =',E10.4/
2     10X,'DPM/ML   ATOMS/ML  PCTERR  ISOTOPIC PCTERR')
      WRITE (1,103) ANS(1), ATOMS(1), ERR(1), FRAC(1), PCT(1), ANS(2),
1     ATOMS(2), ERR(2), FRAC(2), PCT(2), ANS(3), ATOMS(3), ERR(3), FRAC(3),
2     PCT(3), ANS(4), ATOMS(4), ERR(4), FRAC(4), PCT(4)
103   FORMAT ('PU238 =',2E11.4,F6.2,F11.6,F7.3/'PU239 =',2E11.4,
1     F6.2,F11.6,F7.3/'PU240 =',2E11.4,F6.2,F11.6,F7.3/'PU241 =',
2     2E11.4,F6.2,F11.6,F7.3)
      IF (QFIT - 1.5) 60,59,59
59    WRITE (1,104) (I,YNET(I),RM(I), I = 1,MEQ)
104   FORMAT (/ 7X,'YNET   STD DEV' / (I2,F10.1,F8.2))
60    CALL EXIT
      END

```

Subroutine Programs

The following programs are frequently used subroutines. RDSK and WDSK require the FSIO package, which contains a set of I/O programs that use a special format to provide much faster disk transfer rates.

Subroutine AVE

This subroutine determines the average value of a group of data points.

Description of arguments

NS = starting channel for the averaging.
NE = ending channel for the averaging.
SUM = sum of the values between NS and NE inclusive.
AV = average value.
Y = name of data array.

C AVE

C THIS ROUTINE DETERMINES THE AVERAGE COUNT OF A GROUP OF DATA
C VALUES FROM NS TO NE INCLUSIVE. IT ALSO RETURNS THE SUM.

```
      SUBROUTINE AVE (NS, NE, SUM, AV, Y)
      DIMENSION Y(2)
      SUM = 0.0
      SM = NE - NS + 1
      DO 2 I = NS, NE
2     SUM = SUM + Y(I)
      AV = SUM / SM
      RETURN
      END
```

Subroutine BKGD

This routine removes the background continuum from a peak grouping and returns the net channel counts and the net total counts. This routine assumes the continuum slope to be zero. If this is not true, use subroutine BKGRD. See subroutines NCTS and NCNTS for variations of BKGD.

Description of arguments

- NS = starting channel of peak grouping found in array Y.
- NE = ending channel of peak grouping found in array Y.
- AVBG1 = background level at the starting channel.
- AVBG2 = background level at the ending channel.
- YSUM = as input, the total gross counts between NS and NE; as output, the total net counts between NS and NE.
- L = length of new YNET array.
- Y = input array name containing gross channel counts.
- YNET = name of array containing newly generated net channel counts (Y and YNET may be the same array).

```
C      BKGD
C      THIS ROUTINE REMOVES THE BACKGROUND FROM A PEAK GROUPING AND
C      RETURNS THE NET CHANNEL COUNTS AND THE TOTAL COUNTS
```

```
      SUBROUTINE BKGD (NS, NE, AVBG1, AVBG2, YSUM, L, Y, YNET)
      DIMENSION Y(2), YNET(2)
      DBG = AVBG2 - AVBG1
      L = 0
      YS = 0.0
      XS = 0.0
      AVEBG = AVBG1 + DBG * 0.5
      XPTS = NE - NS + 1
      YSUM = YSUM - AVEBG * XPTS
      DO 2 I = NS, NE
      L = L + 1
      YS = YS + Y(I) - AVEBG
      BG = AVBG1 + DBG * YS / YSUM
      CNET = Y(I) - BG
      YNET(L) = CNET
2     XS = XS + CNET
      YSUM = XS
      RETURN
      END
```

Subroutine BKGRD

This routine is similar to BKGD. However, in removing the background continuum, it includes the slope of the continuum in its calculations. If the slope in front of the peak groupings is different from in back of it, a smooth transition is made from the one slope to the other.

Description of arguments

All arguments the same as for BKGD except:

SLP1 = slope of the continuum on the low-energy side of the peak grouping in units of counts/channel.

SLP2 = slope of the continuum on the high-energy side of the peak groupings in units of counts/channel.

C BKGRD

C THIS ROUTINE IS USED TO SUBTRACT A BACKGROUND CONTINUUM
C IT ACKNOWLEDGES THE SLOPE ON EITHER END OF THE PEAK GROUPING IN
C ITS CALCULATIONS AND RETURNS THE NET CHANNEL COUNTS AND THE NET
C TOTAL COUNTS.

SUBROUTINE BKGRD (NS, NE, AVBG1, AVBG2, SLP1, SLP2, SUM, L, Y, YNET)
DIMENSION Y(2), YNET(2)

XPTS = NE - NS
DBG = (AVBG2 - AVBG1) - 0.5 * (SLP1 + SLP2) * XPTS
XPTS = XPTS - 1.0
NS = NS + 1
NE = NE - 1
BG2 = AVBG2
SM = 0.0
DO 2 I = NS, NE
2 SM = SM + Y(I)
SM = SM - BG2 * XPTS
XS = 0.0
YS = 0.0
RT2 = 0.0
FDY = 0.0
L = 0
XBG = AVBG1
DSLPL = SLP2 - SLP1
S1 = SLP1

DO 10 I = NS, NE
L = L + 1
DY = Y(I) - BG2
HDY = 0.5 * DY
YS = YS + HDY + FDY
RT1 = YS / SM
XBG = XBG + DBG * DY / SM + S1 + RT1 * DSLPL
FDY = HDY
10 YNET(L) = Y(I) - XBG
XS = XS + YNET(L)

SUM = XS
RETURN
END

Subroutine BOUNDS

This subroutine is useful for determining the starting and ending channel positions of a peak. It does so by comparing the computed net channel count with an uncertainty value associated with the background continuum counts.

Description of arguments

LL	= a value approximately equal to the peak full width at half maximum.	C1	= amplitude parameter used for x-ray analysis. This value should be zero if this is not an x-ray.
MS	= the channel position containing the highest count.	C2	= slope parameter used for x-ray analyses.
YMAX	= the net count peak height.	SQ1	= statistical uncertainty associated with the channel counts in front of the peak.
ALFA	= the peak width parameter.	SQ2	= statistical uncertainty associated with the channel counts in back of the peak.
EXP1	= the tailing amplitude parameter.		
EXP2	= the tailing slope parameter.		

C BOUNDS

C THIS ROUTINE IS USED TO DETERMINE THE PEAK STARTING AND ENDING
C POINTS BY COMPARING THE COMPUTED NET CHANNEL COUNT WITH THE
C STATISTICAL UNCERTAINTY OF THE BACKGROUND CONTINUUM COUNTS.

```
      SUBROUTINE BOUNDS (LL, MS, YMAX, ALFA, EXP1, EXP2, C1, C2, SQ1, SQ2)
      JPK = MS
      L = JPK - LL
      LMT = L - 2
      MS = JPK + LL
      DO 20 I = 1, LMT
      L = L - 1
      DX = L - JPK
      X1 = YMAX * EXP (ALFA * DX * DX)
      X2 = X1 + YMAX * EXP1 * EXP (EXP2 * DX)
      IF (C1) 10, 12, 10
10      XC = C1 * YMAX * EXP (C2 * -DX)
      X1 = C1 + XC
      X2 = C2 + XC
12      IF (X1 - SQ2) 14, 13, 13
13      MS = MS + 1
14      IF (X2 - SQ1) 21, 20, 20
20      CONTINUE
21      LL = L
      RETURN
      END
```


Subroutine CHFIT

This subroutine locates the centroid of a peak using a seven-point parabola fit to the data values surrounding the highest counting channel in the peak. For more exact fits, GFIT should be used.

Description of arguments

L = channel corresponding to highest count
in the peak.
PKPOS = position of the peak.
Y = name of data array.

C CHFIT

C THIS ROUTINE LOCATES THE CENTROID OF A PEAK BASED ON A 7 POINT
C PARABOLA FIT.

```
      SUBROUTINE CHFIT (L,PKPOS,Y)
      DIMENSION Y(2)
      A1 = Y(L-3)
      A2 = Y(L-2)
      A3 = Y(L-1)
      A4 = Y(L)
      A5 = Y(L+1)
      A6 = Y(L+2)
      A7 = Y(L+3)
      PKPOS = L
      PKPOS = PKPOS+(4.5*(A7-A1)+3.*(A6-A2)+1.5*(A5-A3))/
1      (4.*A4 + 3.*(A3+A5) - 5.*(A1 + A7))
      RETURN
      END
```

Subroutine GFIT

This subroutine performs a Gaussian fit using five, seven, or nine data values surrounding the highest counting channel in the peak. A corresponding peak height will also be returned. For a simpler seven-point parabola fit, use CHFIT.

Description of arguments

L = channel corresponding to highest count in the peak.
PKPOS = position of the peak.
PKHT = peak height.
NC = number of data channels to be used in the fit (5, 7, or 9).
Y = name of data array.

C GFIT

C THIS ROUTINE PERFORMS A GAUSSIAN FIT TO 5, 7, OR 9 POINTS
C SURROUNDING THE L TH CHANNEL AND RETURNS THE EXACT PEAK POSITION
C AND THE PEAK HEIGHT.

```
SUBROUTINE GFIT (L, PKPOS, PKHT, NC, Y)
  DIMENSION Y(2)
```

```
  SY = 0.0
  SXY = 0.0
  SXXY = 0.0
  LX = - NC / 2
```

```
  DO 2 I = 1, NC
    K = L + LX
    XL = LX
    YLOG = ALOG (Y(K))
    YX = XL * YLOG
    SXY = SXY + YX
    SY = SY + YLOG
    SXXY = SXXY + XL * YX
    LX = LX + 1
```

2 CONTINUE

```
  PKPOS = L
  IF (NC - 7) 4, 6, 8
4  PKPOS = PKPOS + SXY / (2.8571 * SY - 1.4286 * SXXY)
  PKHT = 0.4857 * SY - 0.14286 * SXXY
  GO TO 10
6  PKPOS = PKPOS + SXY / (2.6666 * SY - 0.6666 * SXXY)
  PKHT = 0.3333 * SY - 0.04762 * SXXY
  GO TO 10
8  PKPOS = PKPOS + SXY / (2.597 * SY - 0.3896 * SXXY)
  PKHT = 0.2554 * SY - 0.021645 * SXXY
10  PKHT = EXP (PKHT)
  RETURN
  END
```

Subroutine MAXVAL

This subroutine is used to determine the channel position and value of the maximum count in a group of data points.

Description of arguments

NS = starting channel of the search.
NE = ending channel of the search.
JMAX = channel position of maximum value.
YMAX = count in channel JMAX.
Y = name of data array.

C MAXVAL

C THIS ROUTINE RETURNS THE POSITION AND VALUE OF THE MAXIMUM VALUE
C IN A GROUP OF DATA POINTS FROM NS TO NE INCLUSIVE.

```
      SUBROUTINE MAXVAL (NS, NE, JMAX, YMAX, Y)
      DIMENSION Y(2)
      YMAX = 0.0
      DO 2 I = NS, NE
      IF (Y(I) - YMAX) 2, 2, 4
4     YMAX = Y(I)
      JMAX = I
2     CONTINUE
      RETURN
      END
```

Subroutine MINVAL

This subroutine is used to determine the channel position and value of the minimum count in a group of data points.

Description of arguments

NS = starting channel of the search.
NE = ending channel of the search.
JMIN = channel position of minimum value.
YMIN = count in channel JMIN.
Y = name of data array.

C MINVAL

C THIS ROUTINE RETURNS THE POSITION AND VALUE OF THE MINIMUM VALUE
C IN A GROUP OF DATA POINTS FROM NS TO NE INCLUSIVE.

```
      SUBROUTINE MINVAL (NS, NE, JMIN, YMIN, Y)
      DIMENSION Y(2)
      YMIN=1 0E38
      DO 4 I=NS, NE
      IF (Y(I)-YMIN)2, 4, 4
2      JMIN=I
      YMIN=Y(I)
4      CONTINUE
      RETURN
      END
```

Subroutines MLR, MLRA, and MLRS

MLR subroutine is the first part of a program which minimizes the sum of squares of the elements of vector R , where $R = Y - BX$. The sequel is MLRA or MLRS. If MLRA is used, the two-dimensional array of coefficients (B) must be written on "DSK1" using "AMTRX" as the file label. The maximum number of equations currently allowed is 140, and the number dimensioned in any given program should be rounded upwards to the nearest unit of ten. These restrictions do not apply if MLRS is used. For an example of this usage, see PMLR on page 17.

Description of arguments

LB = the number of rows specified in the dimension statement for $B(I,J)$.
M = the number of equations.
N = the number of unknowns.
B = the two-dimensional array of coefficients.
Y = the one-dimensional array of constants (data values).

X = a one-dimensional array for the unknowns (answers).
R = a one-dimensional array for the residuals.
W and V = temporary storage arrays of N words each.
NFLG = a flag number which is transmitted to indicate an ill-conditioned matrix.

C MLR

C THIS IS THE FIRST PART OF A LEAST-SQUARES PROGRAM WHICH WAS
C ADAPTED FROM THE "MLR" PROGRAM WRITTEN BY R. E. VON HOLDT, LLL.
C THE SEQUEL IS MLRA OR MLRS.

```
SUBROUTINE MLR (LB, M, N, Y, X, B, R, W, V, NFLG)
  DIMENSION Y(2), B(3), R(2), X(2), W(2), V(2)
  IB = LB
  DO 70 I = 1, N
    T = 0.
    IK = (I-1) * IB
    DO 20 K = 1, M
      LL = IK + K
20    T = T + B(LL) * B(LL)
      IF (T) 30, 29, 30
29    NFLG = 2
      GO TO 75
30    ST = SQRT (T)
      LL = IK + I
      W(I) = ST
      IF (B(LL)) 31, 32, 32
31    W(I) = -ST
32    B(LL) = B(LL) + W(I)
      V(I) = -B(LL) * W(I)
      IF (I-N) 35, 75, 35
```

```

35  IP = I + 1
    DO 60 J = IP, N
      T = 0.
      JK = (J-1) * IB
      DO 40 K = I, M
        LL = IK + K
        LLL = JK + K
40   T = T + B(LL) * B(LLL)
      T = T / V(J)
      DO 50 L = I, M
        LL = JK + L
        LLL = IK + L
50   B(LL) = B(LL) + T * B(LLL)
60   CONTINUE
70   CONTINUE
75   RETURN
    END

```

C MLRA

C THIS IS A SEQUEL TO THE MLR LEAST-SQUARES ROUTINE ADAPTED FROM THE
C "MLR" PROGRAM WRITTEN BY R. H. VOM HOLT, LLL.
C THIS SUBPROGRAM REQUIRES LESS CORE THAN DOES "MLRS" BECAUSE THE
C "A" MATRIX IS STORED AS A FILE ON DSK1. (SEE ROUTINE PMLR)

```

      SUBROUTINE MLRA (LB, M, N, Y, X, B, R, W, V, NPLG)
      DIMENSION Y(2), B(2), R(2), X(2), W(2), V(2), A(140)
      IB = LB
      DO 80 I = 1, M
80    R(I) = Y(I)
      DO 90 I = 1, N
90    X(I) = 0.
      S = 0.
      DO 100 L = 1, M
100   S = S + R(L) * R(L)
      YE = S
      DO 130 I = 1, N
105   T = 0.
      LK = (I-1) * IB
      DO 110 K = I, M
        LL = LK + K
110   T = T + B(LL) * R(K)
      T = T / V(I)
      DO 120 L = I, M
        LL = LK + L
120   R(L) = R(L) + T * B(LL)
130   CONTINUE
      IN = N
140   R(IN) = R(IN) / W(IN)
150   J = IN

```

```

155 J = J - 1
    IF (J) 190,190,160
160 LL = (IN-1) * IB + J
    R(J) = R(J) + R(IN) * B(LL)
    GO TO 155
190 IN = IN - 1
    IF (IN) 200,200,140
200 DO 210 I = 1,N
    SC = X(I)
    SD = R(I)
    XD = SC - SD
210 X(I) = XD
    DO 215 I = 1,M
215 R(I) = Y(I)

```

C READ SECTIONS OF "A" MATRIX FROM PREVIOUSLY STORED "AMTRX" FILES
C STORED ON DSK1.

```

    CALL IOPEN ('DSK1','AMTRX')
    DO 220 J = 1,N
    READ (4,300) (A(I), I = 1,IB)
300 FORMAT (10A6)
    XD = X(J)
    DO 230 I = 1,M
230 R(I) = R(I) - A(I) * XD
220 CONTINUE
    RE = 0.
    DO 240 I = 1,M
240 RE = RE + R(I) * R(I)
    IF (RE - S) 250,270,270
250 S = RE
    GO TO 105
270 IF (RE - YE) 280,290,290
290 NFLG = 3
280 RETURN
    END

```

C MLRS

C THIS IS A SEQUEL TO THE MLR PROGRAM. IT REQUIRES MORE CORE THAN
C DOES "MLRA" BECAUSE THE "A" MATRIX IS RESIDENT AT ALL TIMES.
C (ADAPTED FROM THE MLR PROGRAM WRITTEN BY R. H. VON HOLDT, LLL)

```

SUBROUTINE MLRS (LB,M,N,A,Y,X,B,R,W,V,NFLG)
DIMENSION Y(2),A(2),B(2),R(2),X(2),W(2),V(2)
IB = LB
DO 80 I = 1,M

```

```

80  R(I) = Y(I)
    DO 90 I = 1,N
90  X(I) = 0.
    S = 0.
    DO 100 L = 1,M
100  S = S + R(L) * R(L)
    YE = S
105  DO 130 I = 1,N
    T = 0.
    LK = (I-1) * IB
    DO 110 K = 1,M
    LL = LK + K
110  T = T + B(LL) * R(K)
    T = T/V(I)
    DO 120 L = 1,M
    LL = LK + L
120  R(L) = R(L) + T * B(LL)
130  CONTINUE
    IN = N
140  R(IN) = R(IN)/W(IN)
150  J = IN
155  J = J - 1
    IF (J) 190,190,160
160  LL = (IN-1) * IB + J
    R(J) = R(J) + R(IN) * B(LL)
    GO TO 155
190  IN = IN - 1
    IF (IN) 200,200,140
200  DO 210 I = 1,N
    SC = X(I)
    SD = R(I)
    XD = SC - SD
210  X(I) = XD
    DO 230 I = 1,M
    RD = Y(I)
    DO 220 J = 1,N
    LL = (J - 1) * IB + I
    AD = A(LL)
    XD = X(J)
    RD = RD - AD * XD
220  CONTINUE
    R(I) = RD
230  CONTINUE
    RE = 0.
    DO 240 I = 1,M
240  RE = RE + R(I) * R(I)
    IF (RE - S) 250,270,270
250  S = RE
    GO TO 105
270  IF (RE - YE) 280,290,290
290  NFLG = 3
280  RETURN

```


Subroutine MLRE

This subroutine is used to obtain the diagonal elements of the inverse matrix of the normal equation following calls to MLR and MLRA or MLRS. These values can be used for error analyses.

Description of arguments

- D = the one-dimensional array of diagonal elements.
- IA = the number of rows specified in the dimension statement for B(I,J).
- NN = the number of unknowns
- B = the two-dimensional array generated by MLRA or MLRS.
- W = a temporary storage array of N words.

C MLRE

C THIS ROUTINE OBTAINS THE DIAGONAL MATRIX ELEMENTS FOLLOWING A
C LEAST-SQUARES CALCULATION. THESE CAN BE USED FOR ERROR ANALYSIS.

```
      SUBROUTINE MLRE (D, IA, NN, B, W)
      DIMENSION D(2), B(2), W(2)
      N = NN
      DO 6 I = 1, N
      D(I) = -1. / W(I)
      IF (N - 1) 4, 4, 1
1      JJ = I + 1
      DO 3 J = JJ, N
      S = 0.
      KJ = J - 1
      DO 2 K = I, KJ
      LL = (J - 1) * IA + K
      S = S + D(K) * B(LL)
2      CONTINUE
      D(J) = S / W(J)
3      CONTINUE
4      S = 0.
      DO 5 L = I, N
      S = S + D(L) **2
5      CONTINUE
      D(I) = S
6      CONTINUE
      RETURN
      END
```

Subroutine NCTS

This subroutine is a variation of the BKGD routine for removing the background continuum from a peak grouping. The routine returns the net channel counts,

the total net counts, the average background in front and in back of the peak grouping, and the length of the net count array.

Description of Arguments

N1 and N2 = channel range for obtaining background in front of peak.

M1 and M2 = channel range for obtaining background in back of peak.

ND = number of values in final net count array.

S1 = sum of channel counts used for determining the two background levels.

AV1 and AV2 = average background counts in front and in back of the peak grouping.

Y1 = input array name containing gross channel counts.

Y2 = name of array containing newly generated net channel counts.

```
C      NCTS
C      THIS ROUTINE RETURNS THE NET CHANNEL COUNTS IN A PEAK, THE SUM OF
C      THE NET COUNTS, THE AVERAGE BACKGROUND LEVELS IN FRONT AND IN
C      BACK OF A PEAK GROUPING, AND THE NUMBER OF DATA POINTS IN THE NET
C      COUNT ARRAY.
      SUBROUTINE NCTS (N1, N2, M1, M2, ND, S1, S2, AV1, AV2, Y1, Y2)
      DIMENSION Y1(2), Y2(2)
      CALL AVE (N1, N2, S1, AV1, Y1)
      CALL AVE (M1, M2, S2, AV2, Y1)
      S1 = S1 + S2
      K = N2 + 1
      KK = M1 - 1
      CALL AVE (K, KK, S2, DUM, Y1)
      DBG = AV2 - AV1
      L = 0
      YS = 0.0
      XS = 0.0
      CHANS = KK - K + 1
      S2 = S2 - AV2 * CHANS
      IF (DBG) 1, 4, 4
1     DO 2 I = K, KK
        L = L + 1
        YS = YS + Y1(I) - AV2
        BG = AV1 + DBG * YS / S2
        CNET = Y1(I) - BG
        Y2(L) = CNET
2     XS = XS + CNET
      GO TO 8
4     XK = KK - K + 1
      YS = AV1
      DBG = DBG / XK
      DO 6 I = K, KK
        L = L + 1
        YS = YS + DBG
        CNET = Y1(I) - YS
        Y2(L) = CNET
6     XS = XS + CNET
8     S2 = XS
      ND = L
      RETURN
      END
```

Subroutine NCNTS

This routine is very similar to the NCTS routine, except that it returns an average background value and also an error associated with the total net counts.

Description of arguments

For N1, N2, M1, M2, ND, S1, S2, Y1, and Y2, see definitions given under NCTS subroutine.

ER = the standard deviation to be associated with the total net count returned by S2.

AVEBG = value of the average channel background.

C NCNTS

C THIS ROUTINE IS SIMILAR TO "NCTS" ROUTINE EXCEPT THAT IT RETURNS
C AN AVERAGE BACKGROUND AND THE ERROR ASSOCIATED WITH THE TOTAL NET
C COUNT

```
      SUBROUTINE NCNTS (N1, N2, M1, M2, ND, S1, S2, ER, AVEBG, Y1, Y2)
      DIMENSION Y1(2), Y2(2)
      CALL NCTS (N1, N2, M1, M2, ND, S1, S2, AV1, AV2, Y1, Y2)
      AVEBG = 0.5 * (AV1 + AV2)
      CHANS = M1 - N2 + 1
      XPTS = M2 + N2 - M1 - N1 + 1
      IF (SUM) 1, 2, 2
1      SUM = 0.0
2      ER = SQRT (SUM + AVEBG * CHANS * (1. + CHANS / XPTS))
      RETURN
      END
```

Subroutine RDSK

This subroutine is used to read any portion of a floating-point data file which has been stored using the special FSIO file structured I/O routines. These routines are more restrictive but much faster than the PS-8 system I/O routines. See RDDSK.

Description of arguments

DEV = device name.
ST = starting channel position of read.
NDPTS = number of values to be read.
FNAME = name of file to be read. A .DA extension is assumed.
Y = array name into which the data is to be transferred.

```
C      RDSK
C      USE TO READ SPECIALLY FORMATED FILES

      SUBROUTINE RDSK(DEV, SCHN, NDP, FNAM, ARRAY)
      DIMENSION FBUF(86), ARRAY(1)
      IF (DEV-ODEV) 40, 5, 40
      IF (FNAM-FOLD) 40, 10, 40
      I=1
      NUM2=(SCHN-1.)/64.
      NUMS=NUM2+NBS
      N1=SCHN-64. *FLOAT(NUM2)
      CALL DKRD(1, 1, NUMS, FBUF)
      ARRAY(I)=FBUF(N1)
      N1=N1+1
      I=I+1
      IF(N1-64) 20, 20, 30
      IF(I-NDP) 18, 18, 25
      RETURN
      30    NUMS=NUMS+1
           N1=1
           GO TO 15
      40    CONTINUE
           CALL LUKUP(DEV, 1, FNAM, 'DA', NBS, NBLK)
           FOLD=FNAM
           ODEV=DEV
           GO TO 10
      END
```

Subroutine RDDSK

This subroutine is used to read any portion of a floating-point data file which has been stored in the normal PS-8 system format. Because of field boundary restrictions, the array normally cannot be greater than 1280 data values. This system of file structuring and reading is very general in design and utility but is quite slow compared with the more restrictive filing system used by the file structured I/O routines (FSIO subroutine package). Also see RDSK.

Description of arguments

DEV = device name.
ST = starting channel position of read.
NDPTS = number of values to be read.
FNAME = name of file to be read. A .DA extension is assumed.
Y = array name into which the data is to be transferred.

```
C      RDDSK
C      USE TO READ DATA FILES WRITTEN ON DISKS IN PS-8 SYSTEM FORMAT
C
      SUBROUTINE RDDSK (DEV,ST,NDPTS,FNAME,Y)
      DIMENSION Y(2)
      XN = 0.0
      CALL IOFEN (DEV,FNAME)
      IF (ST - 2048.) 21,20,20
20      READ (4,105) (DUM, I = 1,2047)
      XN = 2047.
21      NST = ST - XN - 1.
      READ (4,105) (DUM, I = 1,NST)
      READ (4,105) (Y(I), I = 1,NDPTS)
105      FORMAT (A6)
      RETURN
      END
```

Subroutine STRIP

This subroutine can be used to strip or "peel" a peak (or a portion of a peak) from a data array. It is useful for removing small interferences.

Description of arguments

NS = starting channel in Y array for computation.
NE = ending channel in Y array for computation.
PKPOS = position of peak.
PKHT = net count height of peak.
ALFA = width parameter of peak.
EXP1 = tailing amplitude parameter.
EXP2 = tailing slope parameter.
C1 = amplitude parameter used for x-ray analyses. This value should be zero if the peak is not an x-ray.
C2 = slope parameter used for x-ray analysis.
Y = array name.

```
C  STRIP
C  THIS SUBROUTINE IS USED TO STRIP A PEAK OR PART OF A PEAK FROM
C  A CONTINUUM
```

```
      SUBROUTINE STRIP (NS, NE, PKPOS, PKHT, ALFA, EXP1, EXP2, C1, C2, Y)
      DIMENSION Y(2)
      DO 10 I = NS, NE
        XI = I
        DX = XI - PKPOS
        DDX = DX * DX
        ADX = ALFA * DDX
        FXX = EXP (ADX)
        EXPN3 = EXP (0.4 * ADX)
        IF (DX) 2, 4, 4
2       EXPN2 = EXP (EXP2 * DX)
        FXX = FXX + EXP1 * EXPN2 * (1. - EXPN3)
        DX = -DX
4       IF (C1) 6, 8, 6
6       FXX = FXX + C1 * (1.05 - EXPN3) * EXP (C2 * DX)
8       Y(I) = Y(I) - PKHT * FXX
10      CONTINUE
      RETURN
      END
```

Subroutine WDSK

This subroutine is used to write onto disk a floating-point data file in the special file structural format. It uses the FSIO package of routines, which are much faster than the PS-8 I/O routines.

Description of arguments

DEV = device name.
SCHN = starting channel for writing. The value of "SCHN" must be a multiple of $(n * 64 + 1)$ where $n = 0, 1, 2, \dots$
NDP = number of values to be written.
FNAM = file name. A .DA extension is assumed. If the file does not already exist on "DEV," a call to "CREAT" must first be made.
ARRAY = array name from which the data is to be transcribed.

```
C      WDSK

C      THIS A FAST SPECIAL PURPOSE DISK WRITING ROUTINE.
C      IT WRITES BLOCKS OF 64 FLOATING POINT VALUES WHICH CAN BE READ
C      BY "RDSK".

      SUBROUTINE WDSK (DEV, SCHN, NDP, FNAM, ARRAY)

C      THE VALUE OF "SCHN" MUST BE A MULTIPLE OF  $(N * 64 + 1)$ .  $N=0, 1, 2, \dots$ 
C      IF "FNAM" DOES NOT EXIST ON "DEV", USE SUBROUTINE "CREAT" TO FORM
C      IT BEFORE CALLING THIS SUBROUTINE.

      DIMENSION FBUF(64), ARRAY(2)

      IF (DEV - ODEV) 4, 2, 4
2      IF (FNAM - ONAM) 4, 10, 4
4      CALL LUKUP (DEV, 1, FNAM, 'DA', NBS, NBLKS)
      ONAM = FNAM
      ODEV = DEV
10     NUM = (SCHN - 1.) / 64.
      NUM = NUM + NBS
      I = 0
15     DO 20 L = 1, 64
      I = I + 1
      IF (I - NDP) 18, 18, 16
16     FBUF(L) = 0.0
      GO TO 20
18     FBUF(L) = ARRAY(I)
20     CONTINUE
      CALL DKWT (1, 1, NUM, FBUF)
      NUM = NUM + 1
      IF (I - NDP) 15, 30, 30
30     RETURN
      END
```

```

/   THIS SUBROUTINE CONVERTS DOUBLE PRECISION
/   FIXED-POINT DATA TO PS-8 FORTRAN FLOATING
/   POINT FORMAT.  INPUT DATA IS FROM A SPECIFIABLE
/   INPUT DEVICE AND FILE NAME.  THE INPUT FILE
/   MAY BE ANY LENGTH BETWEEN 1 AND 32 BLOCKS.
/
/   OUTPUT IS DIRECTED TO A FILE CALLED RAYXXX.DA
/   ON SYS.  THIS FILE MUST EXIST (IE. FLDAT
/   CANNOT CREATE IT) AND MUST BE AT LEAST 64 BLOCKS
/   LONG.
/
/   CALLING SEQUENCE
/   CALL FLDAT ( DEV, FNAM, FEXT )
/
/   DEV      INPUT DEVICE NAME IN A6 FORMAT
/   FNAM     INPUT FILE NAME IN A6
/   FEXT     INPUT FILE EXTENSION IN A6
/
/   AN IFSZ ERROR INDICATES THAT THE INPUT FILE
/   IS TOO LARGE (MORE THAN 32 BLOCKS).

```

1400	OPDEF TADI	1400
3400	OPDEF DCAI	3400
5400	OPDEF JMPI	5400
4400	OPDEF JMSI	4400
7421	OPDEF MQL	7421
7411	OPDEF NMI	7411
7441	OPDEF SCA	7441
7413	OPDEF SHL	7413
7501	OPDEF MQA	7501

/ FLOATING DATA AND RAW DATA BUFFERS

LAP

0200	0000	FDAT,	BLOCK 2
0201	0000		PAGE
0400	0000		BLOCK 2
0401	0000		PAGE
0600	0000	NRAW,	BLOCK 2
0601	0000		PAGE
1000	0000		BLOCK 2
1001	0000		PAGE
			ENTRY FLDAT

1200	0000	GARG,	0
1201	1377		TAD (-1
1202	3016		DCA 16
1203	7000	FCARG,	NGP
1204	1630		TADI FLDAT#
1205	2230		INC FLDAT#

/ SUBROUTINE FLDAT

MAR 27 1974 9:13

1206	3215		DCA FC1	
1207	7240		CLA CMA	
1210	1630		TADI FLDAT#	
1211	2230		INC FLDAT#	
1212	3017		DCA 17	
1213	7346		CLA CLL CMA RTL	
1214	3362		DCA OCH	
1215	7000	FC1,	NGP	
1216	1417		TADI 17	
1217	4045		JMS 45	
1220	7000		NGP	
1221	3416		DCAI 16	
1222	2362		ISZ OCH	
1223	5215		JMP FC1	
1224	5600		JMPI GARG	
1225	1106	SER,	1106	
1226	2332		2332	/IFSZ
1227	0000	FLDAT,	BLOCK 2	
1230	0000			
1231	1227		TAD FLDAT	/CDF TO ARG LIST
1232	3203		DCA FCARG	
1233	1244		TAD PDEV#	/POINTER TO DEV ARG STORAGE
1234	4200		JMS GARG	/GET DEV ARG
1235	1250		TAD PFNAM#	
1236	4200		JMS GARG	/GET FNAM
1237	1252		TAD PFEXT#	
1240	4200		JMS GARG	/GET FEXT
1241	4033		CALL 6, LUKUP	
1242	0602 06			
1243	6201 05	PDEV,	ARG DEV	
1244	1464 01			
1245	6201 05		ARG NO	
1246	1457 01			
1247	6201 05	PFNAM,	ARG FNAM	
1250	1475 01			
1251	6201 05	PFEXT,	ARG FEXT	
1252	1472 01			
1253	6201 05		ARG ISTRT	
1254	1356 01			
1255	6201 05		ARG ISIZE	
1256	1460 01			
1257	4033		CALL 6, LUKUP	
1260	0602 06			
1261	6201 05		ARG SYS	
1262	1467 01			

1263	6201	05	ARG N1
1264	1461	01	
1265	6201	05	ARG RAYX
1266	1500	01	
1267	6201	05	ARG DAX
1270	1463	01	
1271	6201	05	ARG NSTRT
1272	1357	01	
1273	6201	05	ARG NSIZE
1274	1462	01	
1275	7100		CLL
1276	1376		TAD (40

/ SUBROUTINE FLDAT

MAR 27 1974 9:13

1277	1775		TAD ISIZE	
1300	7620		SNL CLA	
1301	5774		JMP XER2	
1302	1775		TAD ISIZE	
1303	3360		DCA INCNT	/INPUT BLOCK COUNT
1304	4033	FL1,	CALL 4, DKRD	
1305	0403	06		
1306	6201	05	ARG NO	
1307	1457	01		
1310	6201	05	ARG N1	
1311	1461	01		
1312	6201	05	ARG ISTRT	
1313	1356	01		
1314	6201	05	PNRAW, ARG NRAW	
1315	0600	01		
1316	2356		INC ISTRT	/BUMP INPUT BLOCK
1317	1373		TAD (-2	
1320	3361		DCA OCNT	/OUTPUT COUNT
1321	7240		CLA CMA	
1322	1315		TAD PNRAW#	
1323	3016		DCA 16	
1324	1372	FL3,	TAD (-100	
1325	3362		DCA OCH	/OUTPUT CHANNEL COUNT
1326	7240		CLA CMA	
1327	1346		TAD PFDAT#	
1330	3017		DCA 17	
1331	1416	FL5,	TADI 16	/GET LOW ORDER RAW
1332	5771		JMP FL6	
1333	2362	FL7,	ISZ OCH	
1334	5331		JMP FL5	
1335	4033		CALL 4, DKWT	
1336	0404	06		
1337	6201	05	ARG N1	
1340	1461	01		
1341	6201	05	ARG N1	
1342	1461	01		
1343	6201	05	ARG NSTRT	
1344	1357	01		
1345	6201	05	PFDAT, ARG FDAT	

```

1346 0200 01
1347 2357          INC NSTRT
1350 2361          ISZ OCNT
1351 5324          JMP FL3
1352 2360          ISZ INCNT
1353 5304          JMP FL1
1354 4040          RETRN FLDAT
1355 0001 06

```

```

1356 0000          ISTRT, 0
1357 0000          NSTRT, 0
1360 0000          INCNT, 0
1361 0000          OCNT, 0
1362 0000          OCH, 0

```

```

1371 1400 01
1372 7700
1373 7776

```

/ SUBROUTINE FLDAT

MAR 27 1974 9:13

```

1374 1445 01
1375 1460 01
1376 0040
1377 7777

```

PAGE

```

1400 7421          FL6,   MQL           /PUT IN MQ
1401 1416          TADI 16          /GET HIGH RAW
1402 7510          SPA             /NORMALIZE?
1403 5252          JMP FL9          /NO
1404 7411          NMI             /NORMALIZE
1405 7450          SNA
1406 5243          JMP ALZER
1407 3260          DCA ISIZE
1410 7441          SCA             /GET SHIFT COUNT
1411 7040          CMA
1412 1377          TAD (230
1413 3262          DCA NSIZE        /EXPONENT PART
1414 1260          TAD ISIZE
1415 7413          SHL
1416 0000          0
1417 3260          DCA ISIZE
1420 7501          FL8,   MQA
1421 3256          DCA TL
1422 1260          TAD ISIZE
1423 7421          MQL
1424 1262          TAD NSIZE
1425 7413          SHL
1426 0002          2
1427 3417          DCAI 17          /STORE EXPONENT PART
1430 7501          MQA
1431 3260          DCA ISIZE
1432 1256          TAD TL

```

1433	7421		MQL	
1434	7413		SHL	
1435	0002		2	
1436	1260		TAD ISIZE	
1437	3417	ALZ,	DCAI 17	/STORE MS
1440	7501		MQA	
1441	3417		DCAI 17	/STORE LS
1442	5776		JMP FL7	
1443	3417	ALZER,	DCAI 17	/ZERO
1444	5237		JMP ALZ	
1445	4033	XER2,	CALL 1, ERROR	
1446	0105 06			
1447	6201 05		ARG SER	
1450	1225 01			
1451	7402		HLT	
1452	3260	FL9,	DCA ISIZE	/SAVE MS
1453	1377		TAD (230	
1454	3262		DCA NSIZE	/SET EXPONENT
1455	5220		JMP FL8	
1456	0000	TL,	0	
1457	0000	NO,	0	
1460	0000	ISIZE,	0	
1461	0001	N1,	1	

/ SUBROUTINE FLDAT

MAR 27 1974

9: 13

1462	0000	NSIZE,	0
1463	0401	DAX,	0401
1464	0000	DEV,	BLOCK 3
1465	0000		
1466	0000		
1467	2331	SYS,	2331
1470	2300		2300
1471	0000		0000
1472	0000	FEXT,	BLOCK 3
1473	0000		
1474	0000		
1475	0000	FNAM,	BLOCK 3
1476	0000		
1477	0000		
1500	2201	RAYX,	2201
1501	3130		3130
1502	3030		3030
1576	1333 01		
1577	0230		
		END	

ALZ	1437
ALZER	1443
DAX	1463
DCAI	34000P
DEV	1464
DKRD	0000EXT
DKWT	0000EXT
ERROR	0000EXT
FCARG	1203
FC1	1215
FDAT	0200
FEXT	1472
FLDAT	1227EXT
FL1	1304
FL3	1324
FL5	1331
FL6	1400
FL7	1333
FL8	1420
FL9	1452
FNAM	1475
GARG	1200
INCNT	1360
ISIZE	1460
ISTR	1356
JMPI	54000P
JMSI	44000P
LUKUP	0000EXT
MQA	75010P
MQL	74210P
NMI	74110P
NRAW	0600
NSIZE	1462
NSTR	1357
NO	1457
N1	1461
OCH	1362
OCNT	1361
PDEV	1243
PFDAT	1345
PFEXT	1251
PFNAM	1247
PNRAW	1314
RAYX	1500
SCA	74410P
SER	1225
SHL	74130P
SYS	1467
TADI	14000P
TL	1456
XER2	1445

```
/ PHIL SIEMENS      1/4/73
/ FILE STRUCTURED DEVICE LOOKUP, READ, WRITE, CREATE

/ THESE SUBROUTINES PROVIDE FAST AND SIMPLE ACCESS
/ TO PS/8 FILE STRUCTURED DEVICES.  THEY USE
/ 3 PAGES OF MEMORY INCLUDING A 1 PAGE HANDLER
/ (NOTE: 2 PAGE HANDLERS ARE NOT ALLOWED).
/ THE USER MUST USE CARE AS THE ROUTINES
/ GIVE RANDOM ACCESS TO THE MASS STORAGE DEVICE.

/      LOOKUP A FILE

/      CALL LUKUP(DEV,LDV,FNAM,FEXTN,IBLKN,NUM)

/ THIS PERFORMS A FILE LOOKUP ON DEVICE 'DEV'
/ OF THE FILE 'FNAM', 'FEXTN'.  THE CALL RETURNS
/ WITH THE STARTING BLOCK NUMBER OF THE FILE
/ IN 'IBLKN' AND THE NUMBER OF BLOCKS (NEGATIVE)
/ IN THE FILE IN 'NUM'.  AN ASSIGNMENT IS MADE
/ BETWEEN THE PHYSICAL DEVICE 'DEV' AND A
/ LOGICAL UNIT NUMBER 'LDV' (0-3).  READS
/ AND WRITES TO THE PROPER UNIT ARE MADE THROUGH
/ THE LOGICAL UNIT NUMBER.

/      CREATE A FILE

/      CALL CREAT(DEV,LDV,FNAM,FEXTN,IBLKN,NUM)

/ THIS CREATES A FILE NAMED 'FNAM', 'FEXTN' ON
/ 'DEV'.  'NUM' (POSITIVE) SPECIFIES THE MAX-
/ IMUM LENGTH OF THE FILE.  IT IS THE USERS
/ RESPONSIBILITY TO GUARANTEE THAT HE WILL NEVER
/ WRITE BEYOND THE END OF THE FILE.  THE CALL
/ RETURNS THE STARTING BLOCK NUMBER OF THE CREATED
/ FILE IN 'IBLKN'.  AS IN LUKUP, AN ASSIGNMENT
/ IS MADE BETWEEN 'DEV' AND THE LOGICAL DEVICE
/ NUMBER 'LDV'.

/      WRITE

/      CALL DKWT(LDV,NBLKS,IBLKN,ARRAY)

/ WRITES 'NBLKS' BLOCKS (256 WORDS) OF DATA
/ STARTING AT BLOCK 'IBLKN' ONTO 'LDV' FROM
/ 'ARRAY'.

/      READ

/      CALL DKRD(LDV,NBLKS,IBLKN,ARRAY)

/ READS 'NBLKS' BLOCKS (256 WORDS) OF DATA
/ STARTING AT BLOCK 'IBLKN' ONTO 'LDV' FROM
/ 'ARRAY'.
```

/ DEVICE ASSIGNMENT AL VANLEHN 1/3/74

/ CALL DEVAS(DEV,LDV)

/ THIS FETCHES THE HANDLER FOR DEVICE 'DEV'
 /AND ASSIGNS LOGICAL UNIT # 'LDF' TO IT.
 /READS AND WRITES CAN THEN BE MADE ON THIS
 /DEVICE.

/ ERRORS

/ LUER -- LOOKUP ERROR
 / CRER -- CREATE ERROR
 / RWER -- READ, WRITE ERROR
 / FLER -- SUBROUTINES IN WRONG FIELD
 / FSIO MUST LOAD IN FIELD 0.

ENTRY DEVAS
 ENTRY CREAT
 ENTRY DKRD
 ENTRY DKWT
 ENTRY LUKUP

1400 OPDEF TADI 1400
 3400 OPDEF DCAI 3400
 4400 OPDEF JMSI 4400
 5400 OPDEF JMPI 5400

LAP
 LUKUP, BLOCK 2
 0200 0000
 0201 0000
 0202 5777 JMP LUKB
 0203 6212 LUKC, 6212
 0204 4776 JMSI (200
 0205 0001 ASD, 1 /FETCH HANDLER
 0206 0000 ASDEV, 0
 0207 0000 0
 0210 0000 HANP, 0 /ENTRY POINT
 0211 5775 JMP LERR /ERROR
 0212 1232 TAD FLUNAM
 0213 1316 TAD HNBLKI

0214	3232		DCA FLUNAM	
0215	1210		TAD HANP	
0216	3632		DCAI FLUNAM	
0217	6201	05	TAD BSW	
0220	1774			
0221	7450		SNA	/ASSIGN?
0222	5241		JMP EXTIT	/YES
0223	7710		SPA CLA	/CREATE?
0224	5773		JMP CRT	/YES
0225	3232		DCA FLUNAM	/POINTER TO FILE NAME
0226	1207		TAD ASDEV#	/GET DEVICE #
0227	6212		6212	
0230	4776		JMSI (200	/LOOKUP
0231	0002	FLUN,	2	
0232	0000	FLUNAM,	0	/BLOCK #
0233	0000	FLUCNT,	0	/LENGTH
0234	5775		JMP LERR	
0235	1232		TAD FLUNAM	
0236	4275		JMS RBACK	/PASS BACK BLOCK #
0237	1233		TAD FLUCNT	
0240	4275		JMS RBACK	/PASS BACK LENGTH
0241	6212	EXTIT,	6212	/DUMP THE USR OUT
0242	4776		JMSI (200	
0243	0011		11	
0244	4040	EXTIU,	RETRN LUKUP	
0245	0005	06		
0246	0231	01	FLNAMI, FLUN	
0247	0205	01	ASDEVI, ASD	

/ FSIO SUBROUTINES

MAR 27 1974 9:30

0250	0000	SETUP,	0	
0251	3010		DCA 10	/POINTER
0252	1200		TAD LUKUP	
0253	3254		DCA SETDF	
0254	0000	SETDF,	0	/SET CALLING DF
0255	1601		TADI LUKUP#	
0256	3267		DCA GETWD#	/FIELD OF ARGUMENT
0257	2201		INC LUKUP#	
0260	1601		TADI LUKUP#	
0261	3254		DCA SETDF	/ADDRESS OF ARG.
0262	2201		INC LUKUP#	
0263	4266		JMS GETWD	
0264	4266		JMS GETWD	
0265	5650		JMPI SETUP	
0266	0000	GETWD,	0	
0267	7000		NOP	/SET ARG FIELD
0270	1654		TADI SETDF	
0271	2254		INC SETDF	
0272	6201		6201	
0273	3410		DCAI 10	
0274	5666		JMPI GETWD	

0275	0000	RBACK,	0	
0276	3266		DCA GETWD	
0277	1200		TAD LUKUP	
0300	3301		DCA RBSDF	
0301	0000	RBSDF,	0	/SET CALLING FIELD
0302	1601		TADI LUKUP#	
0303	3311		DCA RBF	
0304	2201		INC LUKUP#	
0305	1601		TADI LUKUP#	
0306	3254		DCA SETDF	
0307	2201		INC LUKUP#	
0310	1266		TAD GETWD	
0311	0000	RBF,	0	
0312	3654		DCAI SETDF	
0313	6201		6201	
0314	5675		JMPI RBACK	
0315	0000	RW,	0	
0316	0553 01	HNBLKI,	HANBLK	

/ FSIO SUBROUTINES

MAR 27 1974 9:30

0317	0000	DKWT,	BLOCK 2	
0320	0000			
0321	1372		TAD (4000	
0322	3315		DCA RW	
0323	1317		TAD DKWT	
0324	3200		DCA LUKUP	
0325	1320		TAD DKWT#	
0326	3201		DCA LUKUP#	
0327	4771	DKWA,	JMS GETDK	/GET ARG.
0330	1316		TAD HNBLKI	
0331	3317		DCA DKWT	
0332	1717		TADI DKWT	
0333	3317		DCA DKWT	/HANDLER ENTRY
0334	4771		JMS GETDK	
0335	7106		CLL RTL	
0336	7006		RTL	
0337	7006		RTL	
0340	7004		RAL	
0341	3362		DCA NBLK	/BLOCK COUNT
0342	4771		JMS GETDK	
0343	3364		DCA WBNUM	/BLOCK #
0344	1200		TAD LUKUP	
0345	3346		DCA DKSDf	
0346	0000	DKSDF,	0	
0347	1601		TADI LUKUP#	
0350	0370		AND (70	
0351	1315		TAD RW	
0352	1362		TAD NBLK	
0353	3362		DCA NBLK	/WRITE NBLKS AT FIELD X
0354	2201		INC LUKUP#	
0355	1601		TADI LUKUP#	

0356	3363		DCA DKCA	/ADDRESS
0357	2201		INC LUKUP#	
0360	6201		6201	
0361	4717		JMSI DKWT	
0362	0000	NBLK,	0	
0363	0000	DKCA,	0	
0364	0000	WBNUM,	0	
0365	5767		JMP DERR	
0366	5244		JMP EXTIU	

0367 0550 01
0370 0070
0371 0446 01
0372 4000
0373 0500 01
0374 0547 01
0375 0536 01
0376 0200
0377 0410 01

PAGE

/ FSIO SUBROUTINES

MAR 27 1974 9:30

0400	0000	DEVAS,	BLOCK 2	
0401	0000			
0402	1200		TAD DEVAS	
0403	6201 05		DCA LUKUP	
0404	3777			
0405	1201		TAD DEVAS#	
0406	3776		DCA LUKUP#	
0407	5211		JMP LUKB#	
0410	7201	LUKB,	CLA IAC	
0411	3347		DCA BSW	
0412	6201 05	LUKD,	TAD ASDEVI	
0413	1775			
0414	4774		JMS SETUP	/PUT 2 WORDS OF 1ST ARG INTO ASDEV
0415	1773		TAD FLNAMI	
0416	4774		JMS SETUP	/LOGICAL DEV #
0417	4220		JMS LUKA	
0420	0000	LUKA,	0	
0421	1220		TAD LUKA	
0422	0372		AND (7600	/GET THIS PAGE
0423	1371		TAD (200	/ADD OFFSET
0424	6201 05		DCA HANP	/SET HANDLER PAGE
0425	3770			
0426	1347		TAD BSW	
0427	7650		SNA CLA	
0430	5241		JMP LUKE	
0431	7240		CLA CMA	
0432	4774		JMS SETUP	/MOVE FNAM TO 0-2
0433	4767		JMS GETWD	
0434	1010		TAD 10	
0435	4774		JMS SETUP	/MOVE EXTENSION TO 3

0436	6224		6224	/GET FIELD
0437	7640		SZA CLA	
0440	5350		JMP DERR	/NOT PROPER FIELD
0441	6212	LUKE,	6212	
0442	4645		JMSI XUSR	/CALL IN THE USR
0443	0010		10	
0444	5766		JMP LUKC	
0445	7700	XUSR,	7700	
0446	0000	GETDK,	0	
0447	1253		TAD DARGI	
0450	4774		JMS SETUP	
0451	1254		TAD DARG	
0452	5646		JMPI GETDK	
0453	0453 01	DARGI,	DARGI	
0454	0000	DARG,	0	
0455	0000		0	

/ FSIO SUBROUTINES

MAR 27 1974 9:30

0456	0000	DKRD,	BLOCK 2	
0457	0000			
0460	6201 05		DCA RW	
0461	3765			
0462	1256		TAD DKRD	
0463	3777		DCA LUKUP	
0464	1257		TAD DKRD#	
0465	3776		DCA LUKUP#	
0466	5764		JMP DKWA	
0467	0000	CREAT,	BLOCK 2	
0470	0000			
0471	1267		TAD CREAT	
0472	6201 05		DCA LUKUP	
0473	3777			
0474	1270		TAD CREAT#	
0475	3776		DCA LUKUP#	
0476	7240		CLA CMA	
0477	5211		JMP LUKB#	
0500	3306	CRT,	DCA FENAM	
0501	6201 05		TAD ASDEV#	
0502	1763			
0503	6212		6212	
0504	4771		JMS I (200	
0505	0003		3	
0506	0000	FENAM,	0	/STARTING BLOCK
0507	0000	ELNGT,	0	/- BLOCKS AVAILABLE
0510	5334		JMP CERR	
0511	1306		TAD FENAM	
0512	4762		JMS RBACK	/GIVE STARTING BLOCK # BACK
0513	1253		TAD DARGI	
0514	4774		JMS SETUP	/GET FILE LENGTH

0515	7100		CLL	
0516	1254		TAD DARG	
0517	1307		TAD ELNGT	
0520	7630		SZL CLA	/IS IT LARGE ENOUGH
0521	5334		JMP CERR	
0522	1254		TAD DARG	/YES
0523	3331		DCA CLBLK	
0524	1763		TAD ASDEV#	
0525	6212		6212	
0526	4771		JMS I (200	
0527	0004		4	
0530	0000		0	
0531	0000	CLBLK,	0	
0532	5334		JMP CERR	
0533	5761		JMP EXIT	

/ FS10 SUBROUTINES

MAR 27 1974 9:30

0534	1360	CERR,	TAD (0322	
0535	3345		DCA LER1	
0536	6212	LERR,	6212	/DUMP THE USR
0537	4771		JMSI (200	
0540	0011		11	
0541	4033	XLER,	CALL 1, ERROR	
0542	0106 06			
0543	6201 05		ARG LER1	
0544	0545 01			

0545	1425	LER1,	1425; 0522	
0546	0522			
0547	0000	BSW,	0	
0550	1357	DERR,	TAD (2227	/RWER
0551	3345		DCA LER1	
0552	5341		JMP XLER	

0553	0000	HANBLK,	0; 0; 0; 0	
0554	0000			
0555	0000			
0556	0000			
0557	2227			
0560	0322			
0561	0241 01			
0562	0275 01			
0563	0207 01			
0564	0327 01			
0565	0315 01			
0566	0203 01			
0567	0266 01			
0570	0210 01			
0571	0200			

0572 7600
 0573 0246 01
 0574 0250 01
 0575 0247 01
 0576 0201 01
 0577 0200 01

PAGE

0600 0000 BLOCK 2 /RESERVED FOR HANDLER
 0601 0000

END

/ FSIO SUBROUTINES

MAR 27 1974 9:30

ASD 0205
 ASDEV 0206
 ASDEVI 0247
 BSW 0547
 CERR 0534
 CLBLK 0531
 CREAT 0467EXT
 CRT 0500
 DARG 0454
 DARGI 0453
 DCAI 34000P
 DERR 0550
 DEVAS 0400EXT
 DKCA 0363
 DKRD 0456EXT
 DKSDF 0346
 DKWA 0327
 DKWT 0317EXT
 ELNGT 0507
 ERROR 0000EXT
 EXTIT 0241
 EXTIU 0244
 FENAM 0506
 FLNAMI 0246
 FLUCNT 0233
 FLUN 0231
 FLUNAM 0232

GETDK 0446
 GETWD 0266
 HANBLK 0553
 HANP 0210
 HNBLKI 0316
 JMPI 54000P
 JMSI 44000P
 LERR 0536
 LER1 0545
 LUKA 0420
 LUKB 0410
 LUKC 0203
 LUKD 0412
 LUKE 0441
 LUKUP 0200EXT
 NBLK 0362
 RBACK 0275
 RBF 0311
 RBSDF 0301
 RW 0315
 SETDF 0254
 SETUP 0250
 TADI 14000P
 WBNUM 0364
 XLER 0541
 XUSR 0445