MGA: A Gamma-Ray Spectrum Analysis Code for Determining Plutonium Isotopic Abundances

FORTTRAN Listing of the MGA Code

R. Gunnink

September 1, 1991

The logo of the Lawrence Livermore National Laboratory, featuring a stylized 'L' symbol to the left of the text 'Lawrence Livermore National Laboratory' which is arranged in four lines.

Lawrence
Livermore
National
Laboratory

Received by OSTI
APR 03 1992

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of 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 own rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of 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. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

MGA: A Gamma-Ray Spectrum Analysis Code for Determining Plutonium Isotopic Abundances

FORTTRAN Listing of the MGA Code

R. Gunnink

Manuscript date: September 1, 1991

LAWRENCE LIVERMORE NATIONAL LABORATORY
University of California • Livermore, California • 94551

MASTER

CR

MGA: A Gamma-Ray Spectrum Analysis Code for Determining Plutonium Isotopic Abundances

Volume 3 FORTRAN Listing of the MGA Code

Abstract

Nondestructive measurements of x-ray and gamma-ray emissions can be used to determine the abundances of various actinides in a sample. Volume 1 of this report describes the methods and algorithms we have developed to determine the relative isotopic abundances of actinides in a sample, by analyzing gamma-ray spectra obtained using germanium detector systems. Volume 2 is a guide to using the MGA (Multiple Group Aalysis) computer program we have written to perform plutonium isotopic analyses. This report contains a listing of the FORTRAN instructions of the code.

Introduction

MGA calculates relative isotopic abundances of plutonium and other actinides in a sample. The code performs its analysis using data from a gamma-ray spectrum of the sample taken with a germanium detector.

The code requires no calibrations and can be used to measure virtually any size and type of plutonium sample. Measurement times can be as short as a few minutes; measurement accuracies are typically within 1 or 2%.

The FORTRAN listings in this report are for the VAX computer. The listings for the IBM compatible personal computer differ only in the I/O statements of the code and a few other areas of minor consequence.

This work was performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

```
C*****
```

```
C      CHARACTER*15 BUFF1
C      EQUIVALENCE (BUFF,BUFF1)
```

```
C-----
```

```
C $NOTRUNCATE
C $INCLUDE: 'MGACOM.CMN'
```

```
C-----
```

```
C      !Initialize value
```

```
C      CDABS = 0.0
```

```
C      PUGPSC = 0.0
C      PUFRAC = 1.0
```

```
C      CALL MGASPC
C      IF (PU242 .GT. 50.) ICMP(16) = 0
```

```
C      IF (ERRFLG(8) .EQ. 1) GOTO 99
C      IF (NOPFLG(8) .EQ. 0) THEN
C        BUFF1='Entering MGAGAN'
C        CALL CLR_STATUS
C        CALL PUT_STATUS (BUFF)
C        CALL MGAGAN
C        IF (ERRFLG(1) .EQ. 3) GOTO 99
C      ENDIF
```

```
C      AREA(13) = 0.0
C      !Analyze for the presence of 235U
C      IF (ICMP(9) .GE. 0) THEN
C        BUFF1 = 'Entering MGA235'
C        CALL PUT_STATUS (BUFF)
C        CALL MGA235
C      ENDIF
```

```
C      !Guess of U content.
C      IF (ICMP(10) .GT. 0) PUFRAC = 0.5
C      !Call routine to work on individual peaks.
C      BUFF1 = 'Entering MGAPKS'
C      CALL PUT_STATUS (BUFF)
C      CALL MGAPKS
```

```
C      !Prepare 100 keV region for analysis
C      BUFF1 = 'Entering MGA94 '
C      CALL PUT_STATUS (BUFF)
C      CALL MGA94
```

```
C      !Calc. Cd and Pu thicknesses; also eff. curve.
C      BUFF1 = 'Entering MGAABS'
C      CALL PUT_STATUS (BUFF)
C      CALL MGAABS
C      IF (ERRFLG(1) .EQ. 7) GOTO 99
C      IF ((NOPFLG(8) .EQ. 3) .OR. (NOPFLG(8) .EQ. 4)) THEN
C        NOPFLG(8) = 0
```

```
C      !Repeat analysis
C      GOTO 41
C      END IF
```

```
C      !Default values for 38 keV peak used as flags.
C      AREA(1) = 1.
C      ERR(1) = 1.
```

```
C      !Low energy region will be analyzed
C      IF (NOPFLG(1) .EQ. 1) CALL MGA45
C      !Call routine to work on 100 keV region.
C      BUFF1 = 'Entering MGA100'
C      CALL PUT_STATUS (BUFF)
```

```
C      CALL MGA100
C      !Add in single peak equations and compress
C      BUFF1 = 'Entering MGAEQS'
C      CALL PUT_STATUS (BUFF)
C      CALL MGAEQS
```

```
C      IF (SPC2ID(1:1) .NE. ' ') THEN
C        BUFF1 = 'Entering MGAHIE'
C        CALL PUT_STATUS (BUFF)
C        CALL MGAHIE
C        IF (SPC2ID(1:1) .EQ. ' ') GOTO 50
```

```
C      BUFF1 = 'Entering MGA300'
C      CALL PUT_STATUS (BUFF)
C      CALL MGA300
```

```
C      BUFF1 = 'Entering MGA600'
C      CALL PUT_STATUS (BUFF)
C      CALL MGA600
C      ENDIF
```

```
C      !Perform final calculations and output results
C      50 BUFF1 = 'Entering MGACAL'
C      CALL PUT_STATUS (BUFF)
C      CALL MGACAL
C      IF (ERRFLG(1) .EQ. 9) GOTO 99
C      IF (NOPFLG(8) .GT. 0) GOTO 40
```

```
C      BUFF1 = 'Entering MGAOUT'
C      CALL PUT_STATUS (BUFF)
C      IF (NOOUT .NE. 0) CALL MGAOUT
```

```
C      99 BUFF1 = 'Exit MGA '
C      IF (NOOUT .EQ. 0) CALL PUT_STATUS(BUFF)
```

```
C      END
```

```
----->
```

```

C
C      SUBROUTINE MGASPC
C
C*****
C
C      MGASPC
C      =====
C*****
C
C      Developed for : Safeguards Technology Program
C                    Nuclear Chemistry Division
C                    Lawrence Livermore National Laboratory
C                    Livermore, CA. 94550
C*****
C
C      Title   : MGASPC
C
C      Author  : Ray Gunnink
C      Date    : 24-May-89
C      Language: Microsoft Fortran Ver. 4.01
C      Revision: 9.0 of MGA (December-1991)
C*****
C
C      Abstract:
C
C      Full call to MGASPC specifies an array for file name, a
C      device #, buffer for header, & # characters for file name.
C      The program also allows entry of an options with the file
C      name.
C
C      This version performs only one read of the entire spectrum
C      into the YDATA array. Subsequent subroutines read
C      data from the YDATA array rather than accessing the data
C      from the original spectrum file.
C
C      This version can be run as an "engine". I.e. if a spectrum
C      file is deposited in the \PUIS\MGA\ directory under the file
C      name of PURAW.BIN (in IAEA format), then the program will
C      automatically analyze this spectrum and deposit all results
C      in a file called OUTMGA.TXT in the same directory. There
C      will be no user I/O in this mode.
C*****
C
C      Arguments:
C*****
C
C      Routines
C      Called : C2JDAT,GETDAT,CLR_STATUS,RDBIN,RDCHN
C
C      Libraries
C      Used   :
C*****
C
C      Revision History ( Include date, revision I.D., and details)

```

```

C*****
C
C      Date of last change to this routine: 13-June-1991
C*****
C      COMMON and variable declarations
C*****
C      $INCLUDE: 'MGACOM.CMN'
C      $INCLUDE: 'MGACL.CMN'
C-----
C
C      REAL*8 DSTIM,DTO
C
C      DIMENSION SHP(16)
C
C      CHARACTER*1 M
C      CHARACTER*3 TYPFIL(2)
C      CHARACTER*6 FLFORM
C      CHARACTER*9 LAB
C      CHARACTER*26 SMPID2
C
C      DATA TYPFIL/'CON','TT:'/
C      DATA FLFORM/'BINARY'/
C      DATA LAB/'123456789'/
C      Spectrum format index
C      1 = LLNL (Not available in this version)
C      2 = Old Canberra (Not available in this version)
C      3 = ASCII (Sequential integer)
C      4 = IAEA (Silena analyzer)
C      5 = Nuclear Data uMCA
C      6 = "\MGA\PUIS\PURAW.BIN" file exists in IAEA format
C      7 = New Canberra S100 format
C      8 = ORTEC format
C      Set NOPFLG(3) in BLOCK DATA MGACMN.FOR to specify data format type.
C
C      DATA NDEV/1/
C-----
C-----
C
C      IF (NOPFLG(8) .GT. 0) RETURN
C      NPLOT = 0
C      IF (NOPFLG(3) .NE. 6) THEN
C          Initialize defaults values
C          LPK1 = 1980
C          LPK2 = 2773
C          ISTRY = 0
C          MONTH = 0
C          NDAY = 0
C      ENDIF
C
C      Check to see if the 'PURAW.BIN' file exists. If not, a spectrum
C      file name will be requested.
C
C      IF (ICMPTR .EQ. 0) THEN
C          ERRFLG(1) = 2
C          OPEN (UNIT=IOCH1,FILE='\PUIS\MGA\PURAW.BIN',STATUS='OLD',
C              +      FORM=FLFORM,ERR=3)
C          File PURAW.BIN exists. Therefore set NOPFLG(3) to 6.
C          NOPFLG(3) = 6
C          PU242 = 0.0
C          ERRFLG(1) = 0

```

```

      IF (NOOUT .NE. 0) THEN
        NOOUT = 0
        OPEN (UNIT=NDEV,FILE='\\PUIS\MGA\INMGA.TXT',STATUS='OLD',
+          ACCESS='SEQUENTIAL',ERR=903)
      1003 READ (NDEV,1003) LPK1,LPK2,SHPTM1
          FORMAT (214,F3.1)
          GOTO 903
      ELSE
        GOTO 20
      ENDIF
    C
    3  + OPEN (UNIT=NDEV,FILE='INMGA.TXT',STATUS='OLD',ACCESS=
+       'SEQUENTIAL',ERR=903)
      READ (NDEV,1003) LPK1,LPK2,SHPTM1
    C
    903 CLOSE (NDEV)
      GAIN = 59.5/(LPK2-LPK1)
      ZERO = LPK1 - 148.567/GAIN
      IF (NOPFLG(3) .EQ. 6) GOTO 20
    C
    2  CALL SETUP
    C
      IF (ICMPTR .EQ. 0) THEN
        CALL GETDAT (ISTYR,MONTH,NDAY)
        CALL CHKDAT (NDEV2,NDAY,MONTH,ISTYR)
        ISTRY = ISTRY-1900
        CALL CLR STATUS
        WRITE (BUFF,1002) NDAY,CMONTH(MONTH),ISTYR
    1002 +   FORMAT (20X,'MGAPC (PC Version 9.0 23-Jan-92) Date: ',
+               12,'-',A,'-',12)
      ENDIF
    C
      LPK1 = 148.5/GAIN + ZERO
      LPK2 = 208. /GAIN + ZERO
      WRITE (NDEV2,1010)
    1010 FORMAT('$Enter Spectrum ID: ', $)
      NOPFLG(5) = 0
    1020 READ (NDEV1,1020) SPC1ID
      FORMAT(A)
      IF (SPC1ID(1:1) .EQ. ' ') GOTO 2
    C
      L = 1
      IEND = 40
      IEND2 = 39
    4  J = INDEX (SPC1ID(L:), '/')
      IF (J .EQ. 0) GOTO 18
      IF (IEND .EQ. 40) IEND = J-1
      L = L+J
      IF (SPC1ID(L:L) .NE. 'H'.AND. spc1id(l:l).ne.'h') GOTO 10
      WRITE (NDEV2,1025)
    1025 +   FORMAT (' The Following Options are Available:/'
+   ' /A = Test homogeneity of 241Am content./'
    C   +   ' /C = Calibration mode./'
+   ' /D = Input and fix Cd absorber thickness./'
+   ' /F = Analysis of Freshly separated sample. /'
+   ' (Default is Aged sample.)/'
+   ' /G = Type or Generate file of spectrum data regions. /'
+   ' /H = Help listing /'
+   ' /L = Output to Line printer /'
+   ' /P = Long Printout, useful for diagnostics. /'
+   ' /P/P=Long printout + PKFIT output + Correlation Matrix. /'
+   ' /R = Reverse format of report /'
+   ' /S = Solution Analysis /'
+   ' /T = Input and fix Pu sample Thickness. /'
+   ' /U = Uranium is present. /'

```

```

+   ' (Code will usually find U automatically.) /'
+   ' /1to9 = Lab. spectrum format index. /'
+   ' 3=ASCII, 4=IAEA binary, 5=ND uMCA, 6=IAEA "PUIS" /'
+   ' 7=Canberra S100, 8=Ortec Spectran /'
    C
      GOTO 2
    C
    10 IF (SPC1ID(L:L).EQ.'F'.or.spc1id(l:l).eq.'f') THEN
    C      !Fresh sample
      NOPFLG(1) = -1
      ICMP(7) = 1
    C      WRITE (NDEV2,1022)
    1022 +   FORMAT (' Use 38-59 keV region in calculation? (Y/N): ', $)
    C      READ (NDEV1,1023) M
    1023 +   FORMAT (A)
    C      !Use low energy region
      IF (M.EQ.'Y'.or.m.eq.'y') NOPFLG(1) = 1
    C      END IF
    C
      IF (SPC1ID(L:L).EQ.'U'.or.spc1id(l:l).eq.'u') THEN
    C      !Uranium present in sample.
      NOPFLG(2) = 1
    C      !Free up U alpha induced fluorescence component.
      ICMP(10) = 1
    C      !Deter. U238 from 1001 keV pk.
      ICMP(11) = 1
    C      END IF
    C
      !241Am heterogeneity
      IF (SPC1ID(L:L).EQ.'A'.or.spc1id(l:l).eq.'a') ICMP(15) = 1
    C      !Long printout.
      IF (SPC1ID(L:L).EQ.'P'.or.spc1id(l:l).eq.'p')
+      NOPFLG(5) = NOPFLG(5)+1
    C      !Y100.DAT file for plotting.
      IF (SPC1ID(L:L).EQ.'G'.or.spc1id(l:l).eq.'g') NPLT = 1
      IF (SPC1ID(L:L).EQ.'L'.or.spc1id(l:l).eq.'l') THEN
        IF (IPRINT .EQ. 'Y') THEN
          IPRINT = 'N'
        ELSE
          IPRINT = 'Y'
        ENDIF
      ENDIF
    C      !Calibration mode
    C      IF (SPC1ID(L:L).EQ.'C'.or.spc1id(l:l).eq.'c') NOPFLG(6) = 1
    C      IF (SPC1ID(L:L).EQ.'S'.or.spc1id(l:l).eq.'s') NTYPE=NTYPE+1
    C      IF (SPC1ID(L:L).EQ.'R'.or.spc1id(l:l).eq.'r') THEN
        IF (IREPRT .EQ. 0) THEN
          IREPRT = 1
        ELSE
          IREPRT = 0
        ENDIF
      ENDIF
    C      IF (SPC1ID(L:L).EQ.'T'.or.spc1id(l:l).eq.'t') THEN
        !Set flag to fix Pu sample thickness
        NOPFLG(10) = 1
      ENDIF
    C      IF (SPC1ID(L:L).EQ.'D'.or.spc1id(l:l).eq.'d') THEN
        WRITE (NDEV2,1028)
    1028 +   FORMAT (' Cd absorber thickness (g/cm2) = ', $)
        READ (NDEV1,1029) CDABS
        NOPFLG(9) = 1
      ENDIF
    C
      IF (SPC1ID(L:L).EQ.'R' .OR. SPC1ID(L:L).EQ.'r') ICMP(16)=1

```



```

DO 16 I = 1,9
IF (SPC1ID(L:L) .EQ. LAB(I:I)) NOPFLG(3) = I
16 CONTINUE
C
GOTO 4
C
18 IF (NOPFLG(10) .EQ. 1) THEN
WRITE (NDEV2,1027)
1027 FORMAT (' Pu sample thickness (g/cm2) = ', $)
READ (NDEV1,1029) PUGPSC
1029 FORMAT (F7.3)
IF (PUGPSC .EQ. 0.0) PUGPSC = .001
ENDIF
IF (NTYPE .GT. 0) NUMDET = 1
IF (NTYPE .GE. 2) THEN
C
NTYPE = 2
Get calibration information
WRITE (NDEV2,1030)
1030 FORMAT (' Sample Concentration (Gms/L): ', $)
READ (NDEV1,1031) CONC
1031 FORMAT (F10.7)
CONC = .001*CONC
C
WRITE (NDEV2,1032)
1032 FORMAT (' Sample Area (cm2): ', $)
READ (NDEV1,1031) SURFC
C
WRITE (NDEV2,1034)
1034 FORMAT (' Sample Depth (cm): ', $)
READ (NDEV1,1031) DEPTH
VOLUME = SURFC*DEPTH
WRITE (NDEV2,1035)
1035 FORMAT (' Approximate cell-to-detector distance (cm): ', $)
READ (NDEV1,1031) XDIST
C
DD = (XDIST + 1. + .5*DEPTH)**2
ATFCT = SQRT(DD/(DD+.159*SURFC))
PUGPSC = CONC * VOLUME/(SURFC*ATFCT)
C
NOPFLG(10) = 1
Fix g/cm2 at input value
ENDIF
C
IF (IPRINT .EQ. 'Y' .OR. IPRINT .EQ. 'y') THEN
IPRINT = 'Y'
open(unit=ndev3,file='PRN',STATUS='UNKNOWN')
WRITE (NDEV3,1002) NDAY,CMONTH(MONTH),ISTYR
WRITE (NDEV3,1013)
1013 FORMAT ( )
ELSE
OPEN (UNIT=NDEV3,FILE=TYPEFIL(ICMPTR+1),STATUS='UNKNOWN')
ENDIF
IF (NPLT .GT. 0) THEN
WRITE (NDEV2,1050)
1050 FORMAT (' Choose type of output: ', $)
+ ' 1 = "YDATA.DAT" ASCII file of spectrum data', $
+ ' 2 = "Y100.DAT" ASCII file of 100 keV profiles', $
+ ' 3 = Type/print regions of interest: ', $)
READ (NDEV1,1052) NPLT
1052 FORMAT (I1)
IF (NPLT .EQ. 2) NOPFLG(5) = 5
ENDIF
NOPF = NOPFLG(3)
NRECL = 0
IF (NOPF.EQ.1 .OR. NOPF.EQ.5. OR. NOPF.EQ.7 .OR. NOPF.EQ.8) THEN
NRECL = 4
IF (NOPF .EQ. 1 .OR. NOPF .EQ. 7) NRECL = 128

```

```

IF (NOPF .EQ. 8) NRECL = 32
OPEN(UNIT=IOCH1,FILE=SPC1ID(1:IEND),STATUS='OLD',
+ ACCESS='DIRECT',RECL=NRECL,ERR=19)
ELSE
IF (NOPF .EQ. 3) THEN
OPEN (UNIT=IOCH1,FILE=SPC1ID(1:IEND),STATUS='OLD',
+ ACCESS='SEQUENTIAL',ERR=19)
ELSE
OPEN(UNIT=IOCH1,FILE=SPC1ID(1:IEND),STATUS='OLD',
+ FORM=FLFORM,ERR=19)
ENDIF
END IF
GOTO 20
C
19 CLOSE(IOCH1)
WRITE (NDEV2,1042)SPC1ID
1042 FORMAT(' CANNOT OPEN SPECTRUM FILE: ',A,' TRY AGAIN')
IF (NOPFLG(3) .EQ. 6) STOP
GO TO 2
C
20 NTYP = NOPFLG(3)
IF (NTYP .EQ. 6) NTYP = 4
NS = 1
NE = NCHANS-1
C
nh = 0
MNTH = 0
CALL DATARD (IOCH1,NTYP,NS,NE,TIME1,RLTIME,YDATA,ISTYR,
+ MNTH,NDAY,nh,nm,SAMPID)
C
IF (NOPFLG(3) .EQ. 6) THEN
This IAEA "PURAW.BIN" file. Therefore, delete.
CLOSE(IOCH1,STATUS='DELETE')
ELSE
CLOSE(IOCH1)
ENDIF
C
If two detectors, then get second spectrum ID.
21 IF (NUMDET .EQ. 2) THEN
WRITE (NDEV2,1100)
1100 FORMAT (' Enter ID of 2nd Spectrum: ', $)
READ (NDEV1,1020)SPC2ID
C
If no ID is entered, assume one spectrum mode.
IF (SPC2ID(1:1) .EQ. ' ') NUMDET = 1
ENDIF
C
IF (NUMDET .EQ. 2) THEN
IF (NRECL .GT. 0) THEN
OPEN(UNIT=IOCH2,FILE=SPC2ID(1:IEND2),STATUS='OLD',
+ ACCESS='DIRECT',RECL=NRECL,ERR=23)
ELSE
IF (NOPF .EQ. 3) THEN
OPEN (UNIT=IOCH2,FILE=SPC2ID(1:IEND2),STATUS='OLD',
+ ACCESS='SEQUENTIAL',ERR=23)
ELSE
OPEN(UNIT=IOCH2,FILE=SPC2ID(1:IEND2),STATUS='OLD',
+ FORM=FLFORM,ERR=23)
ENDIF
ENDIF
GOTO 22
CLOSE(IOCH2)
WRITE (NDEV2,1042) SPC2ID
GO TO 21
ENDIF
C
22 NS = 1
IF (NUMDET .EQ. 2) CALL DATARD (IOCH2,NTYP,NS,NE,

```

```

+      TIME2,RLTIM2,YDATA(NE+1),N1,N2,N3,nh,nm,SMPID2)
CLOSE(IOCH2)
IF (NTYP .EQ. 0) STOP ' CANNOT READ THIS FILE.'
C
IF (NTYP .EQ. 1) THEN
  L = 1
  K = INDEX(SPC1ID(L:),'.')
  L = J+1
  LST = INDEX(SPC1ID(L:),'.')
C      !Use 1st 2 characters of spectrum extension
  LST = LST + 1
  SFIL(7:8) = SPC1ID(LST:LST+1)
ENDIF
C
IF (NOOUT .GT. 0) THEN
  IF (NOPFLG(3) .NE. 6 .AND. NTYPE .GT. 1) THEN
42      WRITE (NDEV2,9002)
9002    FORMAT (' Detector Label (if any)(2 Characters): ', $)
    READ (NDEV1,1020) DETSYS(1:2)
    IF (DETSYS(1:2) .EQ. ' ' .AND. NTYPE .GT. 0) THEN
      WRITE (NDEV2,*)
      ' Solution Analyses require a detector label'
      GOTO 42
    ENDIF
    Assume that U is not present
    ICMP(10) = -1
    SFIL(7:8) = DETSYS(1:2)
  ENDIF
C
  WRITE (NDEV2,9007)
9007    FORMAT (' Sample Information: ', $)
    READ (NDEV1,1020) SAMPID(1:26)
C
  IF (NOPFLG(3) .NE. 6 .AND. MNTH .EQ. 0) THEN
    WRITE (NDEV2,9003)
9003    FORMAT (' Enter measurement date (Day,Month,Year): ', $)
    READ (NDEV1,9004) NDAY,MNTH,ISTYR
9004    FORMAT (3I5)
  ENDIF
C
  !Obtain zero date
  IF (MNTH .NE. 0) THEN
    WRITE (NDEV2,9080)
9080    FORMAT(' Enter sample declaration date (Day,Month,Year) :', $)
    READ (NDEV1,9004) NDAYTO,MNTHTO,ITOYR
  ENDIF
C
  ENDIF
  IF (MNTH .GT. 0) THEN
    MONTH = MNTH
  ELSE
    ERRFLG(2) = 1
    IF (NOOUT .GT. 0) WRITE (NDEV2,9009)
9009    FORMAT (' *** No measurement date given. ***/'
+ ' The computer clocktime will be used for the measurement date.')
    CALL GETDAT (ISTYR,MONTH,NDAY)
    CDATE = 'Computer Analysis'
  ENDIF
C
  IF (ISTYR .GT. 1900)ISTYR = ISTRYR - 1900
  CALL C2JDAT(MONTH,NDAY,ISTYR+1900,DSTIM)
  DSTIM = DSTIM + nh/24.
C
  COUNTS = 0.
  Integrate spectrum to establish count rate
  DO 43 I = 10,NCHANS
43    COUNTS = COUNTS + YDATA(I)

```

```

IF (COUNTS .LT. 2.E5) THEN
  ERRFLG(8) = 1
  IF (NOOUT .GT. 0) THEN
9035    WRITE (NDEV2,9035)
    FORMAT (' Very low counts in spectrum. Continue? (Y/N):', $)
    READ (NDEV1,1020) M
    IF (M .EQ. 'y' .OR. M .EQ. 'Y') THEN
      ERRFLG(8) = 0
    ELSE
      RETURN
    ENDIF
  ENDIF
  IF (TIME1 .LE. 0.) THEN
    TIME1 = 1.0
    RLTIME = 1.0
    ERRFLG(3) = 1
  ELSE
    RATE = COUNTS/TIME1
    ERRFLG(4) = RATE/2.E4
    !Calculate approx. RLTIME if it is less than TIME1.
    IF (RLTIME.LT.TIME1)RLTIME=TIME1+.00001*SHPTM1*RATE*TIME1
  ENDIF
C
  IF (NPLT .EQ. 1) THEN
    !Write spectrum data to "YDATA.DAT" file.
    OPEN (UNIT=NDEV,FILE='YDATA.DAT',ACCESS='SEQUENTIAL',
+      STATUS='UNKNOWN')
    DO 51 I = 1,4095
      IY = YDATA(I)
      WRITE (NDEV,9010) IY
9010    FORMAT (I8)
      CONTINUE
      CLOSE(NDEV)
    ENDIF
    IF (NPLT .EQ. 3) THEN
      !Type/print regions of interest.
9011    WRITE (NDEV2,9011)
      FORMAT (' Enter start and end channels (n,m): ', $)
      READ (NDEV2,9013) LST,LEND
9013    FORMAT (2I5)
      IF (LST .GT. 0) THEN
        WRITE (NDEV3,9014) (I,YDATA(I),I=LST,LEND)
9014    FORMAT (I5,F9.0)
        GOTO 52
      ENDIF
    ENDIF
C
    IDT1 = 100. * (RLTIME - TIME1) / RLTIME
    IF (RLTIME .EQ. TIME1) IDT1 = -1
    !Livetime in minutes
    TIME = TIME1 / 60.
C
    IF (NUMDET .EQ. 1) THEN
      !Cannot determine 238U or 241Am heterogeneity
      ICMP(11) = 0
      ICMP(15) = 0
    ELSE
      !Set ICMP(15) = 1 to correct or Am inhomogeneity.
      ICMP(15) = 1
      TIM = TIME2 / 60.
      IDT2 = -1
      IF (RLTIM2 .GT. 0.0) IDT2 = 100. * (RLTIM2 - TIME2) / RLTIM2
    ENDIF
C
    IF (NOOUT .NE. 0) THEN
      WRITE (NDEV2,9040)

```

```

9040  FORMAT (' 242Pu Abundance (%) (-1=Calc.(Old),0=Calc.(New) =', $)
      READ (NDEV1,9050)PU242
9050  FORMAT (F6.4)
      ENDIF
      COLTM = 0.0
      IF (PU242 .LT. 0.0) THEN
9060  WRITE (NDEV2,9060)
      FORMAT (' Reactor Discharge-to-Chem. Sep. Time (yrs): ', $)
      READ (NDEV1,9070)COLTM
9070  FORMAT (F4.1)
      END IF
C
      IF (NOPFLG(1) .EQ. 0) THEN
C
        IF (MNTHTO .EQ. 0) THEN
          !Set zero time equal to count date
          MNTHTO = MONTH
          NDAYTO = NDAY
          ITOYR = ISTRYR
        END IF
C
          !If only all 4 digits given
          IF (ITOYR .GT. 1900) ITOYR=ITOYR-1900
C
          !Convert to Julian date
          CALL C2JDAT(MNTHTO,NDAYTO,ITOYR+1900,DTO)
C
          !Decay days (measured - declared)
          DLTAT = (DSTIM-DTO) + 365.25*(ISTYR-ITOYR)
          IF (MNTHTO .EQ. 0) DLTAT=0.
        END IF
C
          Days decay since 1990
          IF (NTYPE .GT. 0) CDECAY = DSTIM +365.25*(ISTYR-90)
C
          IF (NOPFLG(8) .NE. 0) RETURN
C
          IF (NTYPE .EQ. 2) GOTO 80
C Read in peak shape file
C
          IF (NOPFLG(3) .NE. 6) THEN
C
            !Non-IAEA format
            OPEN (UNIT=NDEV,FILE=SFIL,STATUS='OLD',
+              ACCESS='SEQUENTIAL',ERR=57)
C
            ELSE
              !IAEA format
              OPEN (UNIT=NDEV,FILE='\\PUI5\MGA\SHAPC.TXT',STATUS='OLD',
+                ACCESS='SEQUENTIAL',ERR=57)
            ENDIF
          READ (NDEV,9142) (SHP(I), I=1,15)
9142  FORMAT (E15.7)
          DO 59 I=1,15
59    IF (SHP(I) .NE. 0.) SHAPC(I) = SHP(I)
          CLOSE (NDEV)
C
          GOTO 80
C
57    ERRFLG(5) = 1
60    CLOSE (NDEV)
          IF (SFIL(7:8) .EQ. ' ') THEN
C
            IF (NOOUT .GT. 0) THEN
C
              WRITE (NDEV2,9148)
C9148  FORMAT (' DEFAULT peak shape parameters will be used.')
C
            ENDIF
            GOTO 90
          ENDIF
          IF (NOPF .GT. 1) THEN
            WRITE (NDEV2,9150)SFIL
9150  +  FORMAT (' Shape constant file ',A,' does not exist.',
+            /' The following default values will be used.')
            WRITE (NDEV2,9160)(I,SHAPC(I), I = 1,15)

```

```

9160  FORMAT (' S(',I2,') =',F8.5)
62    WRITE (NDEV2,9170)
9170  FORMAT (' Do you want to change any values? (Y/N): ', $)
      READ (NDEV1,9176) M
9176  FORMAT(A)
      IF ((M .EQ. 'Y') .OR. (M .EQ. 'y')) THEN
        WRITE (NDEV2,9180)
9180  FORMAT (' Type index number, comma, and new value: ', $)
        READ (NDEV1,9190) I,CNST
9190  FORMAT (I3,E15.5)
        IF (I .LT. 1 .OR. I .GT. 15) GOTO 62
        SHAPC(I) = CNST
        GOTO 62
      ENDIF
    ENDIF
C
    !Check if det. slope and curvature is fixed.
80    IF (SHAPC(12) .NE. 0.) THEN
      ESLP = SHAPC(12)
      NOPFLG(4) = 1
    ENDIF
    IF (SHAPC(13) .NE. 0.) THEN
      ECURV = SHAPC(13)
      NOPFLG(4) = NOPFLG(4) + 2
    ENDIF
C
    !Check if Cd absorber thickness is specified
    IF (SHAPC(15) .NE. 0.) THEN
      CDABS = SHAPC(15)
      NOPFLG(9) = 1
    ENDIF
C
    !Check if Pu absorber thickness is specified
    IF (SHAPC(6) .GT. 0.) THEN
      PUGPSC = SHAPC(6)
      NOPFLG(10) = 1
    ENDIF
C
90    RETURN
    END

```

```

C
C      SUBROUTINE MGAGAN
C
C*****
C
C
C      MGAGAN
C      =====
C*****
C
C      Developed for : Safeguards Technology Program
C                    Nuclear Chemistry Division
C                    Lawrence Livermore National Laboratory
C                    Livermore, CA. 94550
C*****
C
C      Title       : MGAGAN
C      Author      : Ray Gunnink
C      Date        : 5/24/89
C      Language    : Microsoft Fortran  Ver. 4.01
C*****
C
C      Abstract :
C
C      This subroutine tries to establish the energy scaling used to obtain
C      the spectrum. The gain and zero channel intercept will be determined
C      primarily on the positions of the 59.5 and 208. keV peaks, which it
C      assumes are always present. These two peaks should normally be
C      located at channels 794 and 2773 respectively, corresponding to
C      a system gain of .075 keV/channel. The zero energy position should
C      be at channel zero. Deviations of +/- 200 channels in the positions
C      of these peaks will be tolerated but such poor set-up practices
C      are not recommended. (The IAEA version will be limited to +/- 50
C      channels.
C      If the 59.5 keV peak cannot be found, the routine will instead
C      use the 129.29 keV peak for the lower energy marker.
C*****
C
C      Arguments :
C*****
C
C      Routines
C      Called   : MAXVAL
C               TRNSFR
C
C      Libraries
C      Used    :
C*****
C      Revision History ( Include date, revision I.D., and details ) *
C*****
C      Version 8.0 for PC's.
C      Version date: 13-March-1991
C      Last change to this routine: 5-March-1991
C

```

```

C*****
C      COMMON and variable declarations
C*****
C
C      CHARACTER*1 M
C
C-----
C
C$INCLUDE: 'MGACOM.CMN'
C-----
C
C
C      EN = 59.5
C      ICH = EN/GAIN+ZERO
C      ICH59 = ICH
C      ICH1 = ICH-220
C      ICH2 = 208./GAIN-220.+ZERO
C      EN1 = EN
C      DELEN = 148.5
C      DDN = 144.
C      NJ = 4
C      BGMIN = 100.
C      RMIN = 1.3
C
C      First read in the channels that should contain the 59 keV peak.
C
C      CALL TRNSFR (440,YDATA(ICH1),Y)
C      Starting range is
C      NS = 170
C      NE = 270
C
C      Use successive widths in trying to find the 59 keV peak
C      DO 10 I = 1,NJ
C      IF (I.EQ. 1) THEN
C      CALL MAXVAL (NS,NE,J59,YMAX,Y)
C      ELSE
C      CALL MAXVAL (NS,NS+50,J59,YMAX,Y)
C      CALL MAXVAL (NE-50,NE,JX59,YXMAX,Y)
C      IF (YXMAX .GT. YMAX) J59 = JX59
C      ENDIF
C      BKG = Y(J59-10) + 10.
C      RATIO = (YMAX-SQRT(YMAX)) / (BKG+SQRT(BKG))
C      IF (RATIO .GT. 1.5) GOTO 20
C      Peak height to bkgrd was less than 1.5 Test further.
C      IF ((YMAX .GT. BGMIN).AND. (RATIO .GT. RMIN)) THEN
C      Warn user to use less absorber if 59 keV peak is weak.
C      IF (EN .EQ. 59.5) THEN
C      ERRFLG(9) = 1
C      GOTO 12
C      ENDIF
C      IF (NOOUT .LE. 0) GOTO 20
C      WRITE (*,9000)
C      FORMAT (' Trying to determine the energy scaling that was used.')

```

```

C
  ELSE
    IF (NOOUT .LE. 0) GOTO 12
    C      Increase the channel width of the search.
      NS = NS - 50
      NE = NE + 50
  C
  ENDIF
10  CONTINUE
  C      Cannot find a legitimate peak. Abort analysis.
12  IF (ICH .EQ. ICH59) THEN
  C      ! Try to use the 129 keV peak instead of the 59.
    IF (NOOUT .LE. 0) THEN
      ERRFLG(9) = 1
    ELSE
      WRITE (*,9036)
9036  FORMAT (/ ' 59 keV peak not found. 129 keV peak will be used')
      WRITE (*,9030)
    ENDIF
    ICH = 129.29/GAIN+ZERO
    DELEN = 78.71
    EN = 129.294
    ICH1 = ICH-210
    EN1 = EN
    DDN = 74.24
    RMIN = 1.2
    NJ = 2
    GOTO 2
  ELSE
    IF (NOOUT .GT. 0) GOTO 22
    C      Peak cannot be found. Abort analysis.
    WRITE (*,9090)
    ERRFLG(1) = 3
    GOTO 100
  ENDIF
  C      Peak found.
20  IF ((NE-NS) .GT. 100) THEN
  C      IF (NE-NS .GT. 400) THEN
22    IF (RMIN .EQ. 1.1) THEN
      WRITE (*,9080)
9080  FORMAT (' Peak is small. Try again? (Y/N): ', $)
      READ (*,9020) M
      BGMIN = 0.5*BGMIN
      IF (M .EQ. 'Y' .OR. M .EQ. 'y') GOTO 2
      WRITE (*,9090)
9090  FORMAT (' **ABORT** Peak not found. Check gain.')
      ERRFLG(1) = 3
      GOTO 100
    ENDIF
    WRITE (*,9000)
    WRITE (*,9100) EN, 50*NJ, ICH
9100  FORMAT (' No strong ', F5.1, ' keV peak could be found ',
+          ' within +/- ', I4, ' channels of chan. ', I4)
    WRITE (*,9038) EN
9038  FORMAT (' Enter approximate position of ', F5.1, ' keV peak?',
+          ' (0=Abort): ', $)
    READ (*,9039) ICHAN
9039  FORMAT (I4)
    IF (ICHAN .LE. 0) THEN
      C      No input channel. Abort analysis.
      ERRFLG(1) = 3
      GOTO 100
    ELSE
      ICH1 = ICHAN-220
      NJ = 1
      RMIN = 1.1

```

```

      GOTO 2
    ENDIF
  ELSE
    C      Peak not within +/- 50 channels of channel ICH.
    IF (NOOUT .LE. 0) THEN
      ERRFLG(6) = 1
    ELSE
      WRITE (*,9000)
      WRITE (*,9040) EN, ICH, J59+ICH1-1, EN
9040  FORMAT (' Cannot find the ', F5.1, ' keV near channel ', I4,
+          ' However, a peak was found at channel ', I5,
+          ' Is it the ', F5.1, ' keV peak? (Y/N): ', $)
      READ (*,9020) M
      IF ((M .EQ. 'Y') .OR. (M .EQ. 'y')) THEN
        WRITE (*,9035)
        WRITE (*,9050)
9050  FORMAT (' Please try to establish a better energy scaling.')
      ELSE
        C      Abort
        GOTO 22
      ENDIF
    ENDIF
  ENDIF
  C      Check to make sure the peak is not straddling a boundary.
  NS = J59 - 5
  NE = J59 + 5
  CALL MAXVAL (NS, NE, JPK, YMAX, Y)
  IF (JPK .NE. J59) J59 = JPK
  C
  C      Read in the channels which should contain the 208keV peak
  C
40  CALL TRNSFR (440, YDATA(ICH2), Y)
  CALL MAXVAL (1, 440, J208, Y208, Y)
  C
  NS = 170
  NE = 270
  EN = 208.
  ICH = 208./GAIN+ZERO
  C
  DO 60 I = 1, 4
    CALL MAXVAL (NS, NE, J208, YMAX, Y)
    BKG = Y(J208-12) + 10.
    RATIO = (YMAX-SQRT(YMAX)) / (BKG+SQRT(BKG))
    IF (RATIO .GT. 1.3) THEN
      IF (J208+80 .LT. 440) THEN
        CALL MAXVAL (J208+40, J208+80, JPK, YMAX2, Y)
        IF (YMAX2 .GT. YMAX) GOTO 58
      GOTO 70
    ELSE
      GOTO 70
    ENDIF
  ENDIF
  IF (YMAX .LT. 1000.) THEN
    IF (NOOUT .LE. 0) THEN
      ERRFLG(10) = 1
      NS = NS - 50
      NE = NE + 50
    ELSE
      WRITE (*,9000)
      WRITE (*,9010) J208+ICH2-1, EN
      READ (*,9020) M
      IF (M .EQ. 'Y') THEN
        WRITE (*,9035)

```



```
      GOTO 70
    ELSE
58      NS = NS - 50
      NE = NE + 50
C
      ENDIF
    ENDIF
  ENDIF
C
60  CONTINUE
    IF (NOOUT .GT. 0) WRITE (*,9090)
    ERRFLG(1) = 3
    GOTO 100
C
70  IF ((NE-NS) .GT. 100 .AND. NOOUT .GT. 0) THEN
72    IF (NE-NS .GT. 400) THEN
      WRITE (*,9000)
      WRITE (*,9100) EN,ICH
      WRITE(*,9038) EN
      READ(*,9039) ICHAN
      IF (ICHAN .LE. 0) THEN
        ERRFLG(1) = 3
        GOTO 100
      ELSE
        ICH2 = ICHAN-220
        GOTO 40
      ENDIF
    ELSE
      WRITE (*,9000)
      WRITE (*,9040) EN,ICH,J208+ICH2-1,EN
      READ (*,9020) M
      IF ((M .EQ. 'Y') .OR. (M .EQ. 'y')) THEN
        WRITE (*,9035)
        WRITE (*,9050)
      ELSE
C
C
      GOTO 72
    ENDIF
  ENDIF
C
  Check to make sure the peak is not straddling a boundary
  NS = J208 - 5
  NE = J208 + 5
  CALL MAXVAL (NS,NE,JPk,Y208,Y)
  IF (JPk .NE. J208) J208 = JPK
C
80  PK59 = ICH1-1 + J59
    PK208= ICH2-1 +J208
    GAIN = DELEN /(PK208-PK59)
    ZERO = PK59 - EN1/GAIN
C
C
    Test if 203 keV peak was mistakenly thought to be 208 peak.
    NPK = J208 +4.463 * (PK208-PK59)/DDN
    NPK10 = NPK-10
    IF (Y(NPK)-Y(NPK10)-5.*SQRT(Y(NPK10)+10.) .GT. 0.0) THEN
      CALL MAXVAL (NPK-5,NPK+5,J208,YMAX,Y)
      YMAX = YMAX - Y(J208-12)
      IF (YMAX*40. .GT. Y208) GOTO 80
    ENDIF
C
100 RETURN
C
END
```

SUBROUTINE MGA235

MGA235

=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Title : MGA235
Author : Ray Gunnink
Date : 5/24/89
Language : Microsoft Fortran Ver. 4.01
Revision : 9.0 (December-1991)

Abstract :

This routine determines whether 235U is present in the sample. The routine first integrates the 185 keV peak of 235U. If it is quite intense, it proceeds to interpret the 202-208 keV peak region. It does so by first integrating the entire 202-208 keV region and then proportioning the total net area according to the measured peak heights. If the system also contains a large Ge detector, a preliminary search is made for the 1001 keV peak of 238U. If present, the U X-ray component flag is set. (Not implemented in PC version.)

Arguments :

Routines
Called : TRNSFR
GFIT
MAXVAL
STRIP1
NCNTS
AVE

Libraries
Used :

Revision History (Include date, revision I.D., and details) *

Last change to this routine: 13-March-1991

COMMON and variable declarations *

DIMENSION EEN(3),HT(4)

\$INCLUDE: 'MGACOM.CMN'

DATA EEN/202.12,203.55,205.314/

AREA(13) = 0.0
AREA(15) = 0.0
ERR(13) = 100.0

!Read in 185 keV region.

NST = 183.37/GAIN + ZERO
CALL TRNSFR (100,YDATA(NST),Y)

!Find max counts.

CALL MAXVAL (28,38,JPk,YMAX,Y)
CALL MAXVAL (58,68,JPk2,YMAX2,Y)

IF (YMAX2 .GT. 20.*YMAX) THEN

!Wrong gain was used due to high 235U

PK59 = ZERO + 59.536/GAIN
GAIN = 126.18/(JPk2+NST-PK59)
ZERO = PK59 - 59.536/GAIN
JPk = JPk2

ENDIF

!Start of peak

NS2 = JPk - 15
NS1 = NS2 - 10
IF (NS1 .LE. 0) NS1 = 1

!End of Peak

NE1 = JPk + 14
NE2 = NE1 + 10
IF (NE2 .GT. 100) NE2 = 100

!Net cnts

CALL NCNTS (NS1,NS2,NE1,NE2,NDPTS,BGD,A185,ERR(13),AVEBG,Y,Y)

!Peak not present

IF (A185 .LE. 0.0) RETURN

!Uncertainty

ERR(13) = ERR(13)+.005*A185

!Check if Peak is significant

TSTLMT = .3

IF (NOPFLG(2) .EQ. 1) TSTLMT = .5

IF (ERR(13)/A185 .GT. TSTLMT) GOTO 25

!Store area in (13).

AREA(13) = A185
NOPFLG(2) = 2.+A185/AVEBG
ICMP(9) = 1

!Assume U X ray to be present

IF (ICMP(10) .EQ. 0) ICMP(10) = 1

!202 & 205 keV peaks are small

IF (NOPFLG(2) .LT. 20) GOTO 25
ICMP(12) = 0

NS = 200./GAIN + ZERO

!Read in 202-208 region

CALL TRNSFR (140,YDATA(NS),Y)

!Get net counts

NS2 = 1.36/GAIN

NE1 = 9.38/GAIN

NE2 = 10.13/GAIN

CALL NCNTS (1,NS2,NE1,NE2,NDPTS,BGD,TAREA,ER,AVBG,Y,Y)

!Find 208 keV peak

IS = 6.4/GAIN

CALL MAXVAL (IS,IS+10,JPk,YMAX,Y)

```

C          !Get shape parameters
ENX = 208.
CALL SPRAM (ENX,GAIN,SHAPC,SG,ALFA,F12M,EXP1)
C          !Get 208 keV ht.
CALL GFIT (JPK,4,ALFA,EXP1,EXP2,PK208,HT(4),ERFLG,Y)
IF (ERFLG .LT. 0.) ERRFLG(13) = 1
C
NS = JPK - 26
NE = NS + 6
C          !Position of 205 keV peak
PK205 = PK208 - 2.685/GAIN
C          !Approx. height of 205 keV peak
JPK = PK205 + .5
HT205 = Y(JPK)
C          !Strip out 205
WL = 0.
CALL STRIP1 (NS,NE,PK205,HT205,ALFA,EXP1,EXP2,WL,Y)
BG2 = 0.0
DO 5 I = NS,NE
5  BG2 = BG2 + Y(I)
C          !Average remaining bkgd above 205 peak.
BG2 = BG2 / FLOAT(NE-NS+1)
C
SUMHT = HT(4)
DO 10 J = 1,3
C          !Position of peak
JPK = PK208 - (208. - EEN(J)) / GAIN + .5
ALPHA = ALFA
CALL GFIT (JPK,3,ALPHA,EXP1,EXP2,PKP,HT(J),ERFLG,Y)
IF (ERFLG .LT. 0.) ERRFLG(13) = 2
C          !Subtract remaining bkgd.
HT(J) = HT(J) - BG2*(EEN(J)-201.4)/4.87
C          !HT(J) cannot be < 0.
IF (HT(J) .LT. 0.0) HT(J) = 1.0
C          !Integrate peak heights.
SUMHT = SUMHT + HT(J)
10 CONTINUE
C          !Total area no better than .5%
ER = ER + .005 * TAREA
C
DO 20 J = 1,4
C          !Proportion total area according
RATIO = HT(J) / SUMHT
C          !to peak heights.
AREA(J) = TAREA * RATIO
C          !Calc. errors.
ERR(J) = ER * SQRT(RATIO)
20 CONTINUE
C          !Use 205keV for 235U
IF (2.*AREA(3) .GT. AREA(2)) AREA(15) = AREA(3)
ERR(15) = ERR(3)
25 CONTINUE
C          !Check for 2nd spectrum
IF (SPC2ID(1:1) .NE. ' ') THEN
C          !Read in 1000 keV region
NST = 8026
CALL TRNSFR (130,YDATA(NST),Y)
C          !Look for 1001 keV peak
CALL MAXVAL (60,90,JPK,PHT,Y)
C          !Set integration bounds
NS2 = JPK - 14
NS1 = NS2 - 20
NE1 = JPK + 12
NE2 = NE1 + 15
C          !Transfer bkgd region
DO 30 I = NS1,NS2

```

```

30  Y(I) = Y(I-20)
C          !Net counts
CALL NCNTS (NS1,NS2,NE1,NE2,NPTS,S1,A1001,ER1001,AV,Y,Y)
C          !Peak not present
IF (A1001 .LE. 0.) RETURN
C          !Peak insignificant
ER1001 = ER1001/A1001
IF (ER1001 .GT. .3) RETURN
C          !238U probably present
IF (ICMP(10) .EQ. 0 .AND. ER1001 .LT. .1) ICMP(10) = 1
ICMP(11) = 1
C          END IF
C          RETURN
C          END

```

SUBROUTINE MGAPKS

```

C*****
C
C          MGAPKS
C          =====
C*****

```

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

```

C*****
C
C  Title   : MGAPKS
C  Author  : Ray Gunnink
C  Date    : 5/24/89
C  Language : Microsoft Fortran  Ver. 4.01
C  Revision : 8.0 (13-March-1991)
C*****

```

Abstract :

This is a subroutine called by MGA. It's principal purpose is to obtain the areas and associated statistical errors of the 59, 129, 148, 152, 164, 203, and 208 keV peaks from the low-energy spectrum (#1 spectrum).

Arguments :

```

C*****
C
C  Routines
C  Called   : BKGRD
C            TRNSFR
C            GFIT
C            MAXVAL
C            SPRAM
C            STRIP1
C            NCNTS
C            AVE
C
C
C  Libraries
C  Used     :
C*****

```

Revision History (Include date, revision I.D., and details) *

Date of last change to this routine: 5-March-1991

COMMON and variable declarations *

```

C
C  DIMENSION EST(9),NPTS(9),EBG(9),EN(9)
C  DIMENSION POS(3),PHT(3),FWHMSQ(3)
C

```

```

C-----
C
C $INCLUDE: 'MGACOM.CMN'
C $INCLUDE: 'MGACL.CMN'
C-----
C
C      DATA EN /59.54,129.29,148.57,152.68,164.96,203.54,208.,267.54,
C      +        300.1/
C      DATA EST /56.475,122.,142.125,150.,161.85,201.,205.5,264.75,
C      +        294./
C      DATA NPTS /100,150,150,80,130,70,100,100,120/
C      DATA EBG /.38,.76,.76,.38,.38,.38,.38,.76/
C-----
C
C      GSQ = GAIN * GAIN          !Gain squared
C      EXP2 = SHAPC(5) * GAIN     !Tailing slope in channel units
C
C      Calculate the area subtended by the peak "tail" of a 100 keV peak.
C      This data may be used in the analysis of the 165 keV peak in this
C      routine and will be preserved in SHAPC(6) for later use in MGA94.
C
C      !Get shape parameters
C      ENX = 100.
C      CALL SPRAM (ENX,GAIN,SHAPC,SG,ALFA,F12M,EXP1)
C      AR = 0.0
C
C      CA = .4*ALFA              !Tailing truncation constant
C
C      XX = 0.25                  !First 1/4 channel
C      CT = 0.5
C
C      DO 10 I = 1,100
C      XSQ = CA*XX**2
C      DELA = EXP(-EXP2*XX +XSQ *CT)
C      AR = AR + DELA *CT
C      XX = I
C      CT = 1.
C
C      !Test for convergence
C      IF (DELA *200. .LT. AR) GOTO 12
C      CONTINUE
C
C      !Tailing area
C      SHAPC(6) = EXP1 *(1./EXP2 - AR)
C
C      Determine whether there is a significant amount of 237Np present by
C      looking for the 86 keV peak. The presence of lead around the detector
C      area should be minimized to reduce the intensity of Pb X rays that
C      contribute to this region.
C
C      NP = 8
C      NST = 84./GAIN+ZERO
C
C      CALL TRNSFR (90,YDATA(NST),Y)          !Read in channel data.
C      !Test if 88 keV 109Cd is present
C
C      SUMFCT = 0.0
C      IF (SHAPC(10) .LT. 0 .OR. NTYPE .GT. 0) THEN
C      IS1 = 4./GAIN
C      CALL MAXVAL (IS1-6,IS1+6,JPK,YMAX,Y)
C      !Approx. height of 88keV peak
C      HT88 = YMAX - Y(JPK+13)
C      IF (NTYPE .GT. 0) THEN
C      NS = JPK-13
C      NE = JPK+12

```

```

      CALL NCNTS (NS-2,NS,NE,NE+10,ND,BGD,SUM,ERRR,AVEBG,Y,Y)
      ERRR = ERRR/SUM
      CPM88 = 0.0
      IF (SUM .GT. 1000. .AND. ERRR .LT. .05) THEN
        CPM88 = SUM/TIME
        ERR88 = 100.*ERRR
      ENDIF
    C
      !Find 176 keV sumpeak
      IS1 = 175.7/GAIN+ZERO
      CALL MAXVAL (IS1-6,IS1+6,JKP,YMAX,YDATA)
      CALL GFIT (JKP,4,ALFA,EXP1,EXP2,PKP,SUMHT,ERRR,YDATA)
      SUMHT = SUMHT - (YDATA(JKP+13)+YDATA(JKP+14)+YDATA(JKP+15))/3.
      SUMFCT = SUMHT/(HT88*HT88)
    C
      !Approx. height of 85keV Pb x ray
      IS1 = 1.08/GAIN
      IS2 = 1.83/GAIN
      HT=(Y(IS1)+Y(IS1+1)+Y(IS1-1)-Y(IS2)-Y(IS2+1)-Y(IS2-1))/3.
      IB = 4
      IF ((HT-SQRT(Y(IS2)))/Y(IS2) .GT. .2) THEN
        IF (NOOUT .GT. 0 .AND. NOPFLG(8) .EQ. 0) WRITE (NDEV2,8000)
        8000 FORMAT (' ***WARNING: Pb x rays being produced by surrounding',
        + ' materials.***')
        ERRFLG(11) = 1
        IB = 1
      ENDIF
    C
      !Approx. height of 87 keV x ray
      HT = .367 * HT
      IF (HT .LT. 0.0) GOTO 14
    C
      !Strip out 87 keV
      IS1 = 2.88/GAIN
      IS2 = 3.93/GAIN
      WL = 1.4
      CALL STRIP1 (IS1,IS2,3.4875/GAIN,HT,ALFA,EXP1,EXP2,WL,Y)
    C
      !Get net counts
      14 IS1 = 2.06/GAIN
      IS2 = 3.26/GAIN
      CALL NCNTS (IS1-IB,IS1,IS2,IS2+IB,ND,BGD,AREA86,ER86,AVEBG,Y,Y)
      A86 = 0.0
    C
      !Peak too small
      IF (AREA86 .GT. 5. * ER86) THEN
        A86 = AREA86
    C
      !Peak present; set flag.
      ICMP(8) = 1
      NP = 9
    C
      !Initialize
      ENDIF
      ENDIF
    C
      Obtain areas of 59,129,148,152,164,203,208,267, & 300 keV peaks.
    C
      30 TPHT = 0.0
    C
      IRANGE = 1.5/GAIN
      NUM = 3
      PILUP = 0.0
      LMAX59 = 0
      LST = 4
      DEL59 = 0.0
      CONST = 1.0
      HT61 = 0.0
    C
      DO 50 J = 1,NP
        NST = EST(J)/GAIN + ZERO
        IF (NST .GT. 3970) GOTO 50
    C
        !Read in data region

```

```

      CALL TRNSFR (NPTS(J),YDATA(NST),Y)
    C
      !Get shape parameters
      CALL SPRAM (EN(J),GAIN,SHAPC,SG,ALFA,F12M,EXP1)
    C
      !Peak position in array
      NPK = (EN(J)-EST(J))/GAIN+1.5
    C
      Get position of maximum count. Calculate 152 keV peak position
      C because it may be too small when analyzing low burnup samples.
      IF (J .EQ. 4) THEN
    C
      !152 keV peak position
      JPK = NPK+1
      IRANGE = .77/GAIN
      ELSE
        CALL MAXVAL (NPK-IRANGE,NPK+IRANGE,JKP,YMAX,Y)
      ENDIF
    C
      Find starting and ending positions, and bkgd regions for each peak
    C
      !Calc. bounds value for peak
      NBND5 = SQRT (-7.6 / ALFA)
    C
      !Start of peak
      NE = JPK - NBND5-3
    C
      !Start of front-side bkgd
      NBG = EBG(J)/GAIN
      NS = NE - NBG +1
    C
      !End of peak
      MS = JPK + NBND5
    C
      !End of aft-side bkgd
      ME = MS + NBG
      IS = 1.15/GAIN
      IF (J .EQ. 1) THEN
        MS = JPK + 1.53/GAIN
        ME = MS + IS
    C
      !Check for 61.5 keV peak of 239Np
      SHFT = 1.944/GAIN
      ISS = SHFT + JPK
      HT61 = Y(IS) - 0.5*(Y(ISS-5)+Y(ISS+5))
      IF (HT61 .LT. 0.0) HT61 = 0.0
      ALPHA = ALFA
      CALL GFIT (JKP,3,ALPHA,EXP1,EXP2,PKP,HT,ERRR,Y)
      IF (ERRR .LT. 0.) ERRFLG(13) = 3
      WL = 0.0
      CALL STRIP1 (IS-7,IS+7,PKP+SHFT,HT61,ALFA,EXP1,EXP2,WL,Y)
      END IF
    C
      !Do following only for 148 keV
      IF (J .EQ. 3) THEN
        IF (SUMFCT .GT. 0.) THEN
    C
      !Strip out 59-88 keV sum peak
      ALPHA = ALFA
      CALL GFIT (JKP,3,ALPHA,EXP1,EXP2,SUMPOS,YMAX,ERRR,Y)
      SUMPOS = SUMPOS - 1.03/GAIN
      SUMHT = SUMFCT *HT59S *HT88
      LLST = SUMPOS-10
      CALL STRIP1 (LLST,LLST+20,SUMPOS,SUMHT,ALFA,EXP1,EXP2,WL,Y)
      ENDIF
    C
      !Find lowest bkgd above 148 keV peak
      SUM = 0.0
      DO 28 I = MS,ME
        SUM = SUM + Y(I)
        SUMIN = SUM
      DO 29 I = MS,ME+7
        IF (SUM .LE. SUMIN) THEN
          SUMIN = SUM
          JMIN = I
        ENDIF
      SUM = SUM -Y(I) +Y(I+NBG)
    C
      29 CONTINUE

```



```

C      !Transfer bkgd next to peak
CALL TRNSFR (ME-MS+1,Y(JMIN),Y(MS))
C      !Adjust bkgd in front of peak
ALPHA = ALFA
CALL GFIT (JPK,3,ALPHA,EXP1,EXP2,PKP,HT,ERRR,Y)
IB = PKP - 3.3/GAIN
BK = (Y(IB)+Y(IB+1)+Y(IB+2))/3.
PK239 = 2.49/GAIN
C      !Position of 146 kev 239Pu peak
JP = PKP - PK239
HT239 = Y(JP) - BK
PK241 = 2.01/GAIN
C      !Position of 146.5 kev 241Am peak
JP = PKP - PK241 + 1.
HT241 = Y(JP) - BK
C      !Determine which is most intense.
IF (HT239 .GT. HT241) THEN
  HT1 = HT239
  PK1 = PKP - PK239
  PK2 = PKP - PK241
  NNB = 4
ELSE
  HT1 = HT241
  PK1 = PKP - PK241
  PK2 = PKP - PK239
  NNB = 1
ENDIF
IS2 = JPK - 3.53/GAIN
WL = 0.0
CALL STRIP1 (IS2,NE,PK1,HT1,ALFA,EXP1,EXP2,WL,Y)
JP = PK2
HT2 = Y(JP) - BK
IF (HT2 .GT. 0.) CALL STRIP1 (IS2,NE,PK2,HT2,ALFA,EXP1,
+      EXP2,WL,Y)
C      !Transfer bkgd to front side of peak
CALL TRNSFR (NBG-NNB+1,Y(IS2),Y(NS))
ENDIF
C      !Following only for 152 and 203 kev peaks
33 IF (J .EQ. 4 .OR. J .EQ. 6) THEN
  NB = NBG + 1
  JS1 = 1
  IF (J .EQ. 6) THEN
    JS1 = NS
C      Adjust if 101-101 kev summing is suspected
    IF (IDT1 .GT. 20.) JS1 = JS1 - 7
  ENDIF
  IF (JS1 .LT. 1) JS1 = 1
  JE1 = JS1 + NBG
  JS2 = JS1
  JE2 = NS+4
  DO 37 K = 1,2
    SUM = 0.0
    DO 34 I = JS1,JE1
      SUM = SUM + Y(I)
34    SUMIN = SUM
C      !Find minimum
    DO 35 I = JS2,JE2
      IF (SUM .LE. SUMIN) THEN
        SUMIN = SUM
        JMIN = I
      ENDIF
35    SUM = SUM - Y(I) + Y(I+NBG)
    CONTINUE
    IF (K .EQ. 1) THEN
      !Treat front side of peak
      IF (JMIN .GT. NS) THEN

```

```

      NS = JMIN
      NE = NS + NBG
    ENDIF
    JS = NS
    JE = NE
  ELSE
    !Treat aft side of peak
    IF (JMIN .LT. MS) THEN
      MS = JMIN
      ME = MS + NBG
    ENDIF
    JS = MS
    JE = ME
  ENDIF
C      Transfer data
CALL TRNSFR (JE-JS+1,Y(JMIN),Y(JS))
JS1 = MS
JE1 = ME
JS2 = MS-4
JE2 = NPTS(J)-NB
37 CONTINUE
ENDIF
C      !NS<1 is out of bounds
IF (NS .LT. 1) NS = 1
C      !NE cannot be < NS
IF (NE .LT. NS) NE = NS
C      !Out of bounds
IF (ME .GT. NPTS(J)) ME = NPTS(J)
C      !MS cannot be > ME
IF (MS .GT. ME) MS = ME
C      !Do following only for 164 kev
IF (J .NE. 5) GOTO 40
J169 = JPK + 4.5/GAIN
C      !Get height of 169 kev 241 peak
HT = Y(J169) - .5*(Y(J169-10)+Y(J169+10))
HT = 0.134*HT
PKP = JPK + 1.35/GAIN
C      !Strip out 166 kev 241Am peak
IF (HT .GT. 0.) CALL STRIP1 (MS,ME,PKP,HT,ALFA,EXP1,EXP2,
+      WL,Y)
L = JPK-2.4/GAIN
C      !235U present in significant amts.
IF (NOPFLG(2) .GT. 30) L = MS
C      !Transfer bkgd from another region
CALL TRNSFR (NE-NS+1,Y(L),Y(NS))
C      !Do following only for 208 kev
40 IF (J .NE. 7) GOTO 44
C      Check for the presence of 239Np by looking for the 210 kev peak.
C      MS = MS + 5
C      ME = ME + 5
C      IS = 1.73/GAIN
C      J209 = JPK + IS
C      !Look for 209kev peak
YNP = Y(J209) - 0.5*(Y(J209-6)+Y(J209+6))
C      !Test for significance
IF (YNP .LT. 5.*SQRT(Y(J209))) GOTO 44
C      !-1 is flag for MGA94 that 239Np is present
A106 = -1.
C      !Shift background data from above 210 kev peak
ND = 10
CALL TRNSFR (ND,Y(MS-5+IS),Y(MS-5))
C      !Position where area is to be stored
44 JJ = J + 4
IF (J .NE. 9) GOTO 46

```

```

      CALL TRNSFR (NE-NS+1,Y,Y(NS))
      JJ = 14
C      !Get area
46      CALL NCNTS (NS,NE,MS,ME,NDPTS,BGD,AREA(JJ),ERR(JJ),AVEBG,Y,Y)
C      !10 counts may mean trouble
      IF (AREA(JJ) .EQ. 0.0) AREA(JJ) = 1.
      IF (ERR(JJ) .EQ. 0.0) ERR(JJ) = 10.
C      !Error always > .1%
      ERR(JJ) = ERR(JJ) + .001*AREA(JJ)
C      !Check 235U
      IF ((J .NE. 5) .OR. (AREA(13) .LE. .0)) GOTO 47
C      !Peak is quite small
      IF (NOPFLG(2) .LT. 50) GOTO 47
C      !Get 165 keV height
      CALL GFIT (JPK-NE,4,ALPHA,EXP1,EXP2,PKP,HT,ERRR,Y)
C      IF (ERRR .LT. 0.) ERRFLG(13) = 4
C      !Get 165 keV area
      AREA(JJ) = HT *(SQRT(1.133*SG) + SHAPC(6))
C      !Do following only for 59,129,148 & 208 keV peaks
47      IF ((J .LE. 3) .OR. (J .EQ. 7)) THEN
C      !Approx position of peak
      JP = JPK
C      !Preserve width value
      ALPHA = ALFA
C      !Get exact position
      CALL GFIT (JP-NE,NUM,ALPHA,EXP1,EXP2,PKP,HT,ERRR,Y)
      IF (HT .LT. 100.) PKP = NPK - NE
      IF (J.EQ.1 .AND. HT.GT.500. .AND. NOPFLG(8) .EQ. 0)
+      SHAPC(1) = -2.7726*GSQ/ALPHA-.107
      IF (ERRR .LT. 0.) ERRFLG(13) = 5
      PKPOS = PKP + FLOAT(NST+NE-1)
      NUM = NUM+1
C      Obtain the area of the 125 keV peak of 241Am following 129keV meas.
C      IF (J .EQ. 2) THEN
      A125 = 0.0
C      !Read in data again
      CALL TRNSFR (NPTS(J),YDATA(NST),Y)
C      DIF = 4.03/GAIN
      PKP = PKP + FLOAT(NE)
      IF (ERR(JJ)/AREA(JJ) .GT. .1) THEN
C      !129 keV peak is small. Use 125 peak instead.
      NPK = NPK - DIF
      CALL MAXVAL (NPK-10,NPK+10,IS125,YMAX,Y)
      PKP = IS125 + DIF
      ENDIF
C      !Position of 123.6 keV peak of 239Pu
      PK123 = PKP - 5.677/GAIN
      IS = PK123
C      !Strip out the 123.6 keV peak
      CALL STRIP1(IS-9,IS+9,PK123,.004*HT,ALFA,EXP1,EXP2,WL,Y)
C      !Position of 125 keV peak of 241Am
      IS125 = PKP - DIF
C      !Position of 123 keV peak of 241Am
      PK123 = PKP - 6.3/GAIN
C      !Approximate height of 123 keV peak of 241Am
      IS123 = PK123
      CALL MAXVAL (IS123-2,IS123+2,IS123,HT123,Y)
      HT123 = HT123 - Y(IS123+13)
      IS = IS + 4
C      !Strip out 123 keV peak of 241Am
      CALL STRIP1(5,IS,PK123,HT123,ALFA,EXP1,EXP2,WL,Y)
      MS = IS125 + NBND5

```

```

C      !Integrate 124.5, 125.2 & 125.3 keV peaks of 239Pu and Am
      CALL NCNTS (IS-4,IS,MS,MS+10,ND,BGD,A125,ER125,AVEBG,Y,Y)
      ER125 = SQRT(ER125**2 + (.005*A125)**2)
      ENDIF
C      IF ((J .EQ. 1) .OR. (J .EQ. 7)) THEN
C      !Strip out peak
      JPK = PKP
      WL = 0.0
      CALL STRIP1 (1,NDPTS,PKP,HT,ALPHA,EXP1,EXP2,WL,Y)
C      !Don't compute pileup if dead time <5% or peak is small
      IF (IDT1 .GT. 5 .AND. ERR(JJ)/AREA(JJ) .LT. .05) THEN
C      !Transfer positive residuals on high side
      DO 48 I = JPK+4,NDPTS-1
      LST = LST + 1
C      !Add up net pileup counts
      PILUP = PILUP + CONST*Y(I)
      IF (J .EQ. 1) YPILUP(LST) = 0.
      YPILUP(LST)=YPILUP(LST)+CONST*(Y(I)+0.5*(Y(I-1)+Y(I+1)))
48      CONTINUE
      ENDIF
C      IF (LMAX59 .LT. LST) LMAX59 = LST
      LST = 2
      DEL59 = DEL59 + PKP - JPK + 1
      IF (J .EQ. 1) THEN
      HT59S = HT
      HT59 = CONST*HT
      AR59 = CONST*AREA(5)
      CONST = .3
C      !If sample is fresh, don't include 208 pk. 239Np may be present
      IF (ICMP(7) .GT. 0) CONST = 0.0
      GOTO 51
      ENDIF
      DEL59 = .5*DEL59
      HT = 2.*(HT59 + HT*CONST)
      DO 49 I = 1,LMAX59
      IF (YPILUP(I) .LT. 0.) YPILUP(I) = 0.
49      YPILUP(I) = YPILUP(I)/HT
C      PILUP = PILUP*100. / (AR59+CONST*AREA(11))
      IF (PILUP .GT. 1.) THEN
      IF (NOOUT .GT. 0) WRITE (NDEV3,9005) PILUP
9005      FORMAT (' **WARNING: High-Side tail =',F4.1,'% of Peak Area!')
      ERRFLG(12) = PILUP
      END IF
      ENDIF
      IF (J .EQ. 7) THEN
      IF (PHT(1) .GT. 1000.) GOTO 50
      GAIN = R129 *78.706/(PKPOS-POS(2)) + (1.-R129) *
+      59.5 / (PKPOS-POS(3))
      ZERO = POS(2) - 129.294/GAIN
      GOTO 50
      ENDIF
C      !Sum of peak heights
51      TPHT = TPHT + HT
      PHT(J) = HT
C      !Position in 4096 spect
      POS(J) = PKPOS
      FWHMSQ(J) = (-2.7726/ALPHA-.462) * GSQ
      ENDIF
C      Determine the zero energy intercept using the 59, 129 & 148 keV peaks.
C      IF (J .EQ. 3) THEN
C      !129/(129+148) peak height ratio
      R129 = PHT(2)/(PHT(2)+PHT(3))

```

```

      IF (PHT(1) .GT. 1000.) THEN
C          !GAIN weighted by the 129 and 148 intensities
          GAIN = R129*69.758/(POS(2)-POS(1)) + (1.-R129)*89.031/
+          (POS(3)-POS(1))
          ZERO = POS(1) -59.536/GAIN
      ENDIF
      FWCAL = SQRT (SHAPC(1) + 122. * SHAPC(2))
C          !Redetermine resol. slope
      IF (PHT(1) .GT. 1000.) SHAPC(2) = R129*((FWHMSQ(2) -
+          FWHMSQ(1))/69.8) + (1.-R129)* ((FWHMSQ(3) -FWHMSQ(1))/89.0)
C          !Don't redetermine SHAPC(1)
      IF (TPHT .LT. 3000.) GOTO 55
      SHAPC(1) = 0.0
      DO 52 I = 1,3
52      SHAPC(1) = SHAPC(1) + (FWHMSQ(I)-EN(I)*SHAPC(2))*PHT(I)/TPHT
C          !Meas. 122 keV fwhm
55      FWMEAS = SQRT (SHAPC(1) + 122.*SHAPC(2))
      IF (NOOUT .GT. 0 .AND. NOPFLG(5) .GT. 0) WRITE (NDEV3,9000)
+          FWCAL,FWMEAS,SHAPC(2)
9000      FORMAT (' Calc. 122keV FWHM =',F6.3,' Measured FWHM =',F6.3,
+          ' Slope =',F8.5)
C
      IF ((ABS(ZERO) .GT. 50.) .OR. (ABS(GAIN-.075) .GT. 0.0013))
+          THEN
          ERRFLG(6) = 1
          IF (NOPFLG(8) .EQ. 0 .AND. NOOUT .GT. 0) THEN
              WRITE (NDEV2,9010) ZERO,GAIN,POS(1),208./GAIN+ZERO
              IF (NOPFLG(5) .GT. 0 .AND. (IPRINT .EQ. 'Y'))
+                  WRITE (NDEV3,9010) ZERO,GAIN,POS(1),208./GAIN+ZERO
9010      FORMAT(' ***WARNING: ZERO Energy Channel =',F6.1,' GAIN = ',
+          F7.5,/' Ideal values are ZERO = 0.0, GAIN = .075'
+          /' 59.5 keV peak found at channel',F6.0/
+          ' 208. keV peak found at channel',F6.0/
+          ' Please center 59.5 KeV peak at channel 794',/
+          ' and the 208 KeV peak at channel 2773')
          ENDIF
      ENDIF
50      CONTINUE
C          Look for 133 keV peak of 144Ce
          NST = 132.9/GAIN + ZERO
          CALL NCNTS (NST-10,NST,NST+16,NST+26,NPT,$1,CTSFIS(6),
+          ERFIS(6),AVBG,YDATA,Y)
C
      IF (NOPFLG(2) .LT. 20) GOTO 60
C          !Transfer 203 and 208 peak areas generated
      AREA(10) = AREA(2)
C          !in MGA235 routine.
      ERR(10) = ERR(2)
      AREA(11) = AREA(4)
      ERR(11) = ERR(4)
60      RETURN
C
      END

```

SUBROUTINE MGA94

MGA94

=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Title : MGA94
Author : Ray Gunnink
Date : 5/24/89
Language : Microsoft Fortran Ver. 4.01
Revision : 9.0 of MGApc (Jan-1992)

Abstract :

This is a subroutine that was created to reduce the size of the former MGAPKS subroutine and to add a section of coding to obtain an initial estimate the Cd absorber amount and the Pu concentration.

This is a subroutine called by MGA. This routine reads in and prepares the 100 keV region for analysis.

Arguments :

Routines

Called : BKGRD
TRANSFR
MATINV
GFIT
SPRAM
MAXVAL
STRIP1
NCNTS
AVE

Revision History (Include date, revision I.D., and details)

Date of last change to this routine: 20-December-1990

COMMON and variable declarations

DIMENSION DIST1(9),DIST2(4),DIST3(7),YD(35)

DIMENSION XFCTR(4),PKHT(7)

\$INCLUDE: 'MGACOM.CMN'

DATA DIST1/0.,2.411,3.78,4.875,6.405,9.075,5.325,8.325,4.275/
DATA DIST2/2.00175,3.00175,4.098,6.0975/
DATA DIST3/-.81,1.592,-2.3625,-.525,-.435,2.325,3.42/

DATA XFCTR/.96,1.1,.0,1.6/

EXP3 = SHAPC(8) * GAIN !Long-term tailing slope

ENX = 100. !Get shape parameters
CALL SPRAM (ENX,GAIN,SHAPC,SG,ALFA,F12M,EXP1)

TAREA = SHAPC(6) !Peak area to height ratio
SHAPC(6) = SQRT (1.133*SG) + TAREA
WL = 1.3867* SQRT (-ALFA)

XRATO = 1. -.5642*WL +.25*WL**2 !X- to gamma ray peak height ratio.

Determine the slope in the background in front of the 100 keV region.
This slope is assume to apply also to the high side of the 100 keV region.

NS = 82.8 / GAIN + ZERO
CALL TRANSFR (10,YDATA(NS),Y)
CALL AVE (1,10,SUM,AVE1,Y)

IS = 8.18/GAIN
CALL TRANSFR (10,YDATA(NS+IS),Y)
CALL AVE (1,10,SUM,AVE2,Y)

BGSLP = .9*(AVE2 - AVE1) / (IS*AVE2) !Calc. the background slope

IF (BGSLP .LT. .0) BGSLP = .0 !Check lower limit

IF (BGSLP .GT. .0035) BGSLP = .0035 !Check upper limit

BKGDSL = SHAPC(10)
IF (SHAPC(10) .LT. 0.0) BKGDSL = -SHAPC(10)
!% slope in bkgd.

IF (SHAPC(10) .EQ. 0.0) THEN
BKGDSL = 100. *BGSLP
SHAPC(10) = BKGDSL

ENDIF

BKGDSL = BKGDSL+.0001*NOPFLG(2) !Adjustment for high 235U

Read in 100 keV region and prepare it for analysis

NDPTS = 440
NS94 = 90.375/GAIN + ZERO

!Read in data
CALL TRANSFR (440,YDATA(NS94),Y)

Find the approximate position and height of the 94 keV peak

!Find channel with maximum counts

```

      CALL MAXVAL (35,75,J94,HT94,Y)
      ALPHA = ALFA
C      !Get exact position
      CALL GFIT (J94,4,ALPHA,EXP1,EXP2,PK94,HT94,ER,Y)
      IF (ER .LT. 0.) ERRFLG(13) = 6
C      !237Np not present
      IF (ICMP(8) .LE. 0) GOTO 6
C      Strip out 92.3 keV peak due to 237Np-233Pa activity.
C      !Position of 92 keV peak
      PK92 = PK94 - 2.37/GAIN
      IS = 2.3/GAIN
      IS2 = 1.73/GAIN
      IS3 = 2.93/GAIN
C      !Ht of 92 keV peak
      HT92 = Y(J94-IS) - 0.5*(Y(J94-IS2)+Y(J94-IS3))
      WL = .105/GAIN
      CALL STRIP1 (15,35,PK92,HT92,ALFA,EXP1,EXP2,WL,Y)
C      Check for the presence of 105/106 keV peak due to 235U or 239Np
C      The 106.5 keV peak may also be due to alpha-induced Am X rays
C      IS = 2.025/GAIN+.5
C      !Starting position of peak region
      NS2 = J94 - IS
C      !Starting position of front-side bkgrd
      NS1 = NS2 - 1.51/GAIN
C      !Ending position of peak region
      NE1 = J94 + 10.83/GAIN
C      !Check for significant presence of 235U
      IF (NOPFLG(2) .LT. 5) GOTO 12
C      IS2 = 1.3/GAIN
      IS3 = .8/GAIN
C      !Approx. ht of 93 keV
      HT93 = Y(J94-IS2) - .5*(Y(J94-IS)+Y(J94-IS3))
C      !Cannot be negative
      IF (HT93 .LT. 0.0) HT93 = 0.0
C      !Position of 93 keV peak
      POS = PK94 - 1.31/GAIN
C      !Strip out from bkgrd
      CALL STRIP1 (NS1,NS2,POS,HT93,ALFA,EXP1,EXP2,WL,Y)
C      !Position of 92.27 keV peak
      POS = PK94 - 2.39/GAIN
C      !92 keV pk
      PHT = .1*HT93
      WL = .105/GAIN
      CALL STRIP1 (NS1,NS2+10,POS,PHT,ALFA,EXP1,EXP2,WL,Y)
C      !Position of 96.26 keV peak
      POS = PK94 + 1.6/GAIN
      WL = 0.
      CALL STRIP1 (J94+10,J94+20,POS,PHT,ALFA,EXP1,EXP2,WL,Y)
C      !Check for the possible presence of the
      J1 = J94-2.71/GAIN
C      !92.38 & 92.8 keV peaks of 238U
      J2 = J94-1.51/GAIN
      J3 = 1.96/GAIN
      J4 = 2.26/GAIN
C      !.7 for peak overlap
      HT = .7*(Y(J94-J3)+Y(J94-J4)-Y(J1)-Y(J2))
C      !Check for significance
      IF (HT .LT. 2.*SQRT(Y(J2))) GOTO 12
C      !Strip 92.38
      PKPS = PK94 - 30.4
      CALL STRIP1 (J1,J2,PKPS,HT,ALFA,EXP1,EXP2,WL,Y)

```

```

C      !Strip 92.8
      PKPS = PK94 - 24.9
      CALL STRIP1 (J1,J2,PKPS,HT,ALFA,EXP1,EXP2,WL,Y)
C      SHIFT = .3/GAIN
      XCOEF = .0027
      IF ((NOPFLG(2) .GT. 100) .OR. (ICMP(16) .GT. 1)) THEN
C      !High 235U content or Th is present
      HT = Y(NE1+1) - Y(NE1+12)
      POS = PK94 + 11.75/GAIN
C      !Strip out weak 106.6 keV peak
      WL = 0.
      PHT = .04*HT
      CALL STRIP1 (NE1+5,NE1+18,POS,PHT,ALFA,EXP1,EXP2,WL,Y)
C      !Shift background position by 14 channels
      SHIFT = 1.425/GAIN
      NE1 = NE1 + 1./GAIN
C      !X ray fluorescence contribution to bkgrd
      XCOEF = .0013
      GOTO 17
      END IF
C      NS = J94 + 11.3/GAIN
      NE = J94 + 12.0/GAIN
      CALL MAXVAL (NS,NE,J106,YMAX,Y)
      YBG = .5*(Y(J106+13)+Y(J106-13))
      HT = YMAX - YBG
      IF (HT .GT. SQRT(YBG)) THEN
      EN106 = (J106-J94)*GAIN
      IF (EN106 .LT. 11.6) THEN
C      !The peak is due to 239Np
      POS = PK94 + 11.465/GAIN
      WL = 0.0
      NS = POS - 10.
      NE = NS + 22
      CALL NCNTS (NS,NS+1,NE,NE+10,ND,BGD,A106,ER106,AVEBG,
      +      Y,YD)
      ER106 = 100.*ER106/(A106+1.)
      IF (ER106 .GT. 20.) A106 = 0.0
      J106 = POS+.5
      HT = Y(J106) - .5*(Y(J106-12)+Y(J106+12))
      IF (HT .GT. 0.) CALL STRIP1 (J106-14,J106+14,POS,HT,ALFA,
      +      EXP1,EXP2,WL,Y)
      ENDIF
      ENDIF
C      Determine bkgrd levels in front and in back of the 94-104 keV region.
C      !Initial front-side bkgrd (AVBG1)
      CALL AVE (NS1,NS2,BGD,AVBG1,Y)
C      !Ending position of bkgrd region
      NE2 = NE1 + 8
C      !Initial aft-side bkgrd (AVBG2)
      CALL AVE (NE1,NE2,YSUM,AVBG2,Y)
C      BG1 = AVBG2
C      Strip out the Ge "escape peaks" resulting from the 103. & 103.7 keV X ray
C      and gamma rays. Energy = (103.-9.8) and (103.7-9.8)
C      !Approx. height of 103. keV peak
      IS = 8.34/GAIN
      HT = .0018*(Y(J94+IS) - AVBG2)
      PKPS = PK94-1.56/GAIN
      CALL STRIP1(J94-25,J94-15,PKPS,HT,ALFA,EXP1,EXP2,WL,Y)
C      !Approx. height of 103.7 keV peaks

```



```

      IS = 9.1/GAIN
      HTXRA = Y(J94+IS) - AVBG2
C      !Approx. height of escape peak.
      HT = HTXRA * .0018
      PKPS = PK94 - .8175/GAIN
      WL = .105/GAIN
      CALL STRIP1 (J94-10,J94,PKPS,HT,ALFA,EXP1,EXP2,
+       WL,Y)
C
C In the following section of coding, the front side and aft side background
C levels (AVBG1 and AVBG2) are adjusted for contributions to these regions
C from long-term tailing of the peaks (only to the front side) and from the
C Lorentzian tail component of X rays in the regions.
C
C First, adjust AVBG1 for various contributions.
C
C      !Bkgrd slope
      S2 = .01*BKGDSL*AVBG2
C      !Adjust AVBG1 for slope to edge
      AVBG1 = AVBG1*(1+.1*BKGDSL)
C
      CONST = .106/(2.*3.14*GAIN*XRATO)
      DO 20 I = 1,9
C      !Approx. height of peak
      IF (I .LT. 9) THEN
        IDIST = DIST1(I)/GAIN+.5
        PHT=Y(J94+IDIST)-AVBG1
      ENDIF
C      !Distance from bkgrd center
      DIST = (DIST1(I)+2.03)/GAIN
C      !Long term tail contribution
      TALHT = SHAPC(7) *EXP (-DIST *EXP3)
C      !Lorentzian tailing contrib.
      XTAIL = CONST *SHAPC(6) / DIST**2
C      !These are gamma rays
      IF (I .GT. 6) XTAIL = 0.0
C      !Adjust AVBG1
      AVBG1 = AVBG1 - (XTAIL +TALHT) * PHT
C      !Save for later
      IF (I .EQ. 5) HT101 = PHT
20    CONTINUE
C
C Adjust the contributions that intense X rays make to AVBG2 bkgrd level.
C The 103.7 keV peak is due to both X and gamma rays. First subtract out
C the gamma ray contribution due to 241Pu.
C
C      !Start of 100 keV group
      NS = NS2 +1
C      !End of group
      NE = NE1 -1
C      !Height of 103.7 keV X ray
      HTXRA = HTXRA - .2 * HT101
C      !Adjust AVBG2 for various tailing contributions
      AVBG2 = AVBG2-.0006*HT101-XCOEF*HTXRA-.0004*HT94-
+       S2 *SHIFT
C      !Obtain net counts
C
      CALL BKGRD (NS,NE,AVBG1,AVBG2,S2,S2,NDPTS,YSUM,Y,Y)
      IF (A106 .EQ. 0.) THEN
        DO 32 I = NE1,NE+8
          Y(I-NS2) = Y(I)-AVBG2
          NDPTS = NDPTS+8
        ENDIF
C      !Save for K beta group analysis
      BG1 = AVBG2
C      !AVEBG is later used for weighting the equations

```

```

      AVEBG = AVBG1
C      !Ajust NDPTS if 239Wp is present
C      !New position of 94 keV peak
      J94 = J94 - NS2
      ALPHA = ALFA
C      !Exact position
      CALL GFIT (J94,4,ALPHA,EXP1,EXP2,PK94,HT94,ER,Y)
      IF (ER .LT. 0.) ERRFLG(13) = 7
C
C Also locate the exact position of the 101 keV peak for scaling purposes
C
      NS = 10.21/GAIN - NS2
      NE = NS + 10
C      !Approx. position
      CALL MAXVAL (NS,NE,J101,YMAX,Y)
C      !Exact position
      CALL GFIT (J101,4,ALPHA,EXP1,EXP2,PK101,HT101,ER,Y)
      IF (ER .LT. 0.) ERRFLG(13) = 8
C
C Obtain an accurate value of GAIN for the 100 keV region using the average
C of the positions found for the 94 and 101 keV peaks weighted by the
C intensities of each peak relative to their combined intensity.
C
C      !Total height
      TOTHT = HT94 + HT101
C      !Starting channel of 100 keV region
      ST = NS94-1 + NS2-ZERO
C      !New weighted gain
      GAIN = 94.66*HT94/(TOTHT*(ST+PK94))+101.063*HT101/
+       (TOTHT*(ST+PK101))
C      !Gain squared
      GSQ = GAIN * GAIN
C
C A correction must be made to the net counts on the high side of the
C 94.66 and 98.44 keV x-ray distributions due to an asymetry in the
C Lorentzian distribution of the 239Pu component of this x-ray distribution.
C An estimate of the 239Pu/241Pu DPM ratio is obtained using the 203/208 keV
C peak intensities. Equilibrium is assumed.
C
C      !239/241 xray ratio in 94 keV peak.
      CONST = 13.32 *AREA(10)/AREA(11)
C      !13.32 = 4.22E-5/3.02E-6*5.34E-6/5.6E-6
      CONST = CONST /(1.+CONST)
C      !Height of 239Pu component.
      H94 = HT94 *CONST
      IS = 5.86/GAIN
      H98 = Y(IS) * CONST
      SUM1 = 0.0
      SUM2 = 0.0
C
      NS = 2.26/GAIN
      NE = 9.76/GAIN
      IS = 6./GAIN
      DEL = .8625/GAIN
C
      DO 40 I = NS,NE
        XI = I
C      !Distance from 94 keV peak
      DLX = PK94 - XI
      AL = .15 *(DLX +DEL)**2
C      !Integrate Gaussian
      IF (AL .LT. 8.) SUM1 =SUM1 +EXP(-AL)
C      !Increment net counts
      Y(I) =Y(I) + SUM1/4.575 *H94/(DLX)**2
C      !Do following only for 98 keV peak.
      IF (I .LT. IS) GOTO 40

```

```

C      IDistance from 98 keV peak
      DLX = DLX + 3.782/GAIN
      AL = .15 *(DLX + DEL)**2
      IF (AL .LT. 8.) SUM2 = SUM2 + EXP(-AL)
      Y(I) = Y(I) + SUM2/4.575* H98 /(DLX)**2
40    CONTINUE
      IS = 4.44/GAIN
      IS2 = 5.8/GAIN
C      INet height of 97 keV peak
      HT97 = Y(IS) - .012 * Y(IS2)
      IS = 6.91/GAIN
C      INet height of 101 keV peak
      HT101 = HT101 - .005 * Y(IS)
C
C      The following section of code assembles data for the MGAABS routine which
C      estimates the amount of Cd absorber that was used and also the approximate
C      concentration (g/cm2) of Pu in the sample. See abstract for details.
C
C      IPeak width of 94 keV
      ENX = 94.66
      CALL SPRAM (ENX,GAIN,SHAPC,SGG,A94,F12M,EX1)
      WL = .104/GAIN* SQRT (-A94)
C      IX- to gamma ray peak height ratio.
      XRATO = 1. -.5642*WL +.25*WL**2
C      IGet 238Pu abundance of 94keV peak.
      J99 = PK101 - 1.3*GAIN
C      IStore in ANS(1) for use in MGAABS
      ANS(1) = .03*XRATO*Y(J99)/HT101
C      IArea/Peak Height ratio.
      AR94 = SQRT (1.133*SGG)/XRATO + TAREA
C      IArea/Peak Height ratio. 1.03 due to FWHM diff
      AR101 = AR94 * 1.03
C      ISubtract out tails from higher peaks.
      DO 50 J = 2,6
      DIST = DIST1(J)/GAIN
C      IDistance from peak
      JD = DIST+.5
C      ITail height at DIST2(J)
      TAIL = SHAPC(7)*EXP(-EXP3*DIST)
C      ISub. tail contrib.
      HT94 = HT94 - Y(J94+JD)*(.225*AR94/DIST**2+TAIL)
C      ILong term tail not part of area
      IF (J .EQ. 2) HT94 = HT94 *(1.-TAIL)
50    CONTINUE
C
C      ITransfer 59 keV peak area
      CNTS(1) = AREA(5)
C      IAssume at least 2% error
      ERR(5) = ERR(5) + .02*AREA(5)
C      IWeighting factor
      WT(1) = 1. / ERR(5)
C      ITotal calculated 94 keV peak area
      CNTS(2) = AR94 * HT94
C      IApprox. weighting factor
      WT(2) = 1./(.002*CNTS(2)+AR94*SQRT(HT94))
C      ISubtract tailing of 103.748 peak
      HT101 = HT101 - .013*Y(J101+36)
C      ITotal 101 keV peak area
      CNTS(3) = AR101 * HT101
C      IApprox. weighting factor
      WT(3) = 1./(.002*CNTS(3)+AR101*SQRT(HT101))
C
C      The intensity of the 103 keV peak will next be measured.

```

```

C      IApprox. peak height of 103.68 peak, assuming eq
      ---->uil.
      HT1036 = .19* HT101
      IS = 2.63/GAIN
C      IAssume small 103.75 Pu x ray.
      IF ((ICMP(7) .NE. 0) .AND. (A106 .EQ. 0.))HT1036 = Y(J101+IS)
C      IApprox. height of 103.748 peak
      HTXRA = Y(J101+IS+1) - HT1036
      NST = J101 + 1.53/GAIN
C      ITransfer 103 region into YD array
      IS = .77/GAIN
      IS1 = 1.14/GAIN
      IS2 = IS1**2
      DO 54 I = 1,IS2
      YD(I) = Y(I+NST)
      IF ( I .LT. IS) YD(I) = YD(I) - SHAPC(7)*HTXRA*EXP((IS1-I)*EXP3)
54    CONTINUE
C
C      IPeak position of 101 keV peak in YD array.
      PKP = PK101 - NST
C      IPeak position of 103 keV peak in YD array.
      J103 = PKP + 1.973/GAIN
      ENX = 103.
      CALL SPRAM (ENX,GAIN,SHAPC,SGG,A103,F12M,EX1)
C      IStrip out 101
      WL = .108/GAIN
      CALL STRIP1 (1,IS,PKP,HT101,A103,EX1,EXP2,WL,YD)
C      IPosition of 103.68 peak in YD array.
      PKP = PKP + 2.618/GAIN
C      IStrip out 103.68
      WL = 0.
      CALL STRIP1 (1,IS2,PKP,HT1036,A103,EX1,EXP2,WL,YD)
C      IPosition of 103.748 keV X ray; then strip out.
      PKP = PKP + .9
      WL = .112/GAIN
      IF ((ICMP(7) .EQ. 0) .OR. (A106 .NE. 0.))
      +    CALL STRIP1 (1,IS2,PKP,HTXRA,A103,EX1,EXP2,WL,YD)
C      IArea-to-peak ht ratio
      AR103 = SQRT (1.133 *SGG) + TAREA
C      IDefault minimum value
      HT103M = .001 *CNTS(1)/AR103
      DO 55 I = J103-2,J103+2
      IF (I .LT. 1 .OR. I .GT. 35) GOTO 56
      IF (YD(I) .LE. 0.0) GOTO 56
55    CONTINUE
C
C      IFit 103 peak top
      CALL GFIT (J103,2,A103,EX1,EXP2,PK103,HT103,ER,YD)
      IF (ER .LT. 0.) ERRFLG(13) = 9
56    JPK = PKP + 0.5295/GAIN
C      ISave for 240Pu calc. in MGAABS
      ANS(3) = YD(JPK)*XRATO/HT101
C      ISave for X Ray calc. in MGAABS
      ANS(6) = HTXRA / HT101
C      ISubstitute min. value
      IF (HT103 .LT. HT103M) HT103 = HT103M
C      I103 keV peak area
      CNTS(4) = HT103 * AR103
C      IWeighting factor
      WT(4) = 1./ (.002*CNTS(4)+AR103 *SQRT(HT103 +AVEBG))
C      IQuess of 103 kev height of 239Pu.
      HTPU = HT94 - .5 * HT101
      IF (HTPU .LT. 0.) HTPU = 0.
C      IApprox. 241Am 103 keV peak ht.
      HTAM = HT103 - .05 * HTPU
C      IApprox. 241Am counts in 103 peak

```

```

      CNT103 = HTAM / HT103 * CNTS(4)
      !May not be negative.
      IF ( CNT103 .LT. 0.) CNT103 = .1 * CNTS(4)
      !Get ready to look at K beta group
      NE = J94 + NS2 + 24.6 / GAIN
      NE2 = 32.8 / GAIN
      IF (NE2 .GT. 438) NE2 = 438
      !Initialize a high value
      SMIN = 1.E12
      !Find channels with lowest counts
      DO 57 I = NE, NE2
      !Three channel sum
      SUM = Y(I) + Y(I+1) + Y(I+2)
      !Test for smaller number
      IF (SUM .GT. SMIN) GOTO 57
      !Remember index
      JMIN = I
      !Remember sum
      SMIN = SUM
57    CONTINUE
      !Net counts
      S2 = 0.0
      SMIN = SMIN / 3.
      CALL BKGRD (NE1, JMIN, BG1, SMIN, S2, S2, NP, YSUM, Y, YNET)
      The following section obtains the intensity of the 110-111 keV K-beta x rays
      ENX = 111.
      CALL SPRAM (ENX, GAIN, SHAPC, SGG, A111, F12M, EX1)
      NS = 20.55 / GAIN - NE1
      !Find 111 keV peak position
      CALL MAXVAL (NS, NS+12, J111, YMAX, YNET)
      ALPHA = A111
      !Exact position
      CALL GFIT (J111, 3, ALPHA, EX1, EXP2, PK111, HT, ER, YNET)
      IF (ER .LT. 0.) ERRFLG(13) = 10
      WL = .125 / GAIN * SQRT(-A111)
      !Area/Ht
      AR111 = SQRT (1.133 * SGG) / (1. - .5642 * WL + .25 * WL ** 2) + TAREA
      !Start of stripping
      NS = J111 - 16
      !End of stripping
      NE = J111 + 5
      BGSUM = 0.0
      DO 60 J = 1, 4
      CNST = 0.0
      PKP = DIST2(J) / GAIN + PK111
      LPK = PKP + .5
      IS = 6.2 / GAIN
      IF (J .EQ. 3) CNST = .057 * HT + .026 * YNET(J111 + IS)
      !Height of peak to be stripped out.
      HT = YNET(LPK) * XFCTR(J) - CNST
      WL = .12 / GAIN
      *****
      CALL STRIP1 (NS, NE, PKP, HT, A111, EX1, EXP2, WL, YNET)
      !Long term tail amplitude
      TAIL = HT * SHAPC(7) * EXP(-EXP3 * PKP)
      DO 58 I = NS, NE
      !Subtract out long-term tail
58    YNET(I) = YNET(I) - TAIL

```

```

60    CONTINUE
      DIST = .88125 / GAIN
      JP = DIST + .5
      HT = .95 * YNET(J111 - JP)
      PKP = PK111 - DIST
      CALL STRIP1 (J111 - 4, J111 + 4, PKP, HT, A111, EX1, EXP2, WL, YNET)
      !Peak ht.
      CALL GFIT (J111, 4, ALPHA, EX1, EXP2, PK111, HT111, ER111, YNET)
      IF (ER111 .LT. 0.) ERRFLG(13) = 11
      !Find 110 keV peak
      CALL MAXVAL (J111 - 14, J111 - 9, J110, YMAX, YNET)
      CALL STRIP1 (J110 - 3, J110 + 3, PK111, HT111, A111, EX1, EXP2, WL, YNET)
      !Peak ht.
      CALL GFIT (J110, 3, ALPHA, EX1, EXP2, PK110, HT110, ER110, YNET)
      IF (ER110 .LT. 0.) ERRFLG(13) = 12
      !Area of 110 & 111 keV peaks
      CNTS(5) = AR111 * (HT110 + HT111)
      !Approx. weight factor
      WT(5) = 1. / (.002 * CNTS(5) + AR111 * SQRT (HT111))
      Transfer remaining peak areas to CNTS array for use in MGAABS.
      DO 70 J = 6, 11
      L = J
      IF (J .GT. 7) L = J+1
      CNTS(J) = AREA(L)
      WT(J) = 1. / ERR(L)
70    CONTINUE
      IF (NOPFLG(10) .GT. 0) GOTO 72
      !Very crude estimate of Pu g/cm2
      PUGPSC = HTXRA / HT1036
      !Upper bound condition
      IF (PUGPSC .GT. 5.) PUGPSC = 5.
      !Lower bound condition
      IF (PUGPSC .LT. .01) PUGPSC = .01
      !CDABS has been input from term.
72    IF (CDABS .GT. 0.) GOTO 80
      CDABS = -.2 * ALOG(.0023 * CNTS(1) / CNT103 * (1. - EXP(-1.77 * PUGPSC))) /
      + (1. - EXP(-7.54 * PUGPSC))
      !Lower bound condition
      IF (CDABS .LT. .1) CDABS = .1
      !Upper bound condition
      IF (CDABS .GT. 3.) CDABS = 3.
      The intensity of K alpha 3 X ray peaks is quite small but not negligible.
      The peak heights and positions are determined so that they can be stripped
      out in the DO loop below.
80    R = .0025
      !Peak height
      PKHT(1) = R * HT94
      !Peak height
      PKHT(2) = R * HT97
      !Peak Height
      PKHT(3) = R * HTXRA
      There are small contributions to the 100 keV region coming from the
      escape of germanium X rays (9.8 kev). Eg. the 114.37 keV peak produces
      a small satellite peak at 104.57 keV which interferes with the analysis
      of the 104.24 keV peak of 240Pu. The peak heights are estimated and
      subtracted from the net count distribution.

```

```
R = .002
PKHT(4) = R * HT110
PKHT(5) = R * HT111
JR = PK111 + 2./GAIN
PKHT(6) = R * YNET(JR)
JR = PK111 + 2.99/GAIN
PKHT(7) = R * YNET(JR)

C
C   PKP = PK94
C           !Strip out the above 7 peaks
C   WL = .105/GAIN
C   DO 92 J = 1,7
C     IF (J .EQ. 3) PKP = PK101
C     PKPOS = DIST3(J)/GAIN + PKP
C     NS = PKPOS - 8.
C     NE = NS + 16
C     CALL STRIP1 (NS,NE,PKPOS,PKHT(J),ALFA,EXP1,EXP2,WL,Y)
92  CONTINUE
C
C   DO 94 I = 1,NDPTS
94  YNET(I) = Y(I)
C
C   RETURN
C   END
```

SUBROUTINE MGAABS

```
C*****
C
C                                     MGAABS                                     *
C                                     =====                                    *
C                                                                                   *
C*****
C
C    Developed for : Safeguards Technology Program
C                  Nuclear Chemistry Division
C                  Lawrence Livermore National Laboratory
C                  Livermore, CA. 94550
C
C*****
C
C    Title       : MGAABS
C    Author      : Ray Gunnink
C    Date        : 5/25/89
C    Language    : Microsoft Fortran Ver. 4.01
C    Revision    : 9.0 (Jan-1992)
C*****
C
C    Abstract   :
C
C    This routine estimates the amount of Cd (g/cm2) absorber that
C    was used, the thickness (g/cm2) of the Pu sample, and small changes
C    in the shape of the detector efficiency curve from that of a nominal
C    200 mm2 x 13 mm deep Ge detector.
C    The calculations are made using the measured intensities of the
C    following 10 peaks: 59, 94, 101, 103, 110, 129, 148, 165, 203, and
C    208 keV. The equations involve 8 unknowns: the dpm's of 239Pu, 241Pu,
C    241Am, and 237U (when out of equilibrium), the Cd absorber thickness,
C    the Pu sample thickness, the change in the slope of the efficiency
C    curve, and the change in the curvature of the efficiency curve.
C    The equations take advantage of the discontinuity in the Pu absorption
C    curve at 121 keV to differentiate between absorbance due to Cd and
C    that due to Pu. The equations are non-linear in form, requiring an
C    iterative approach to their solution. The following boundary
C    constraints are also applied: Cd (g/cm2) > 0, .01 < Pu (g/cm2) < 5.
C*****
C
C    Arguments :
C*****
C
C    Routines
C    Called    : MATINV
C
C
C    Libraries
C    Used      :
C*****
C
C    Revision History ( include date, revision I.D., and details ) *
C*****
C
C    Date of last change to this routine: 25-July-1990
C*****
C
C    COMMON and variable declarations *
C*****
```

```

C      DIMENSION ENRG(11),BR(4,11),CDMU(11),PUMU(11),
+      INDX(9),TEMP(11),TEMP2(11),DAT(4,11),PBMU(11)
      REAL*8 XAPHA(9,9),XSUM(9),ANSR(9)

C-----
C
C $INCLUDE: 'MGACOM.CMN'
C $INCLUDE: 'MGACL.CMN'
C-----
C
C      DATA CDMU/6.05,1.754,1.476,1.405,1.157,.7894,.5668,.448,.29,
+      .278,.184/
C
C      DATA PBMU/4.64,6.187,5.252,5.009,4.153,2.835,2.003,1.547,.911,
+      .862,.53/
C
C      DATA PUMU/6.786,2.06,1.738,1.653,1.377,3.531,2.52,1.944,1.155,
+      1.1,.603/
C
C      DATA ENRG/59.536,94.66,101.063,103.,111.,129.29,148.567,164.58,
+      203.54,208.,267.54/
C
C      DATA DAT/
C      239Pu      241Pu      241Am      237U
+      0., 8.460E-6, .359, 0.,
+      4.307E-5, 3.060E-6, 0., 0.,
+      0., 6.250E-6, 1.970E-5, 0.,
+      2.170E-6, 0., 1.910E-4, 0.,
+      2.553E-5, 1.816E-6, 0., 0.,
+      6.26E-5, 0., 0., 0.,
+      0., 1.890E-6, 0., 0.,
+      0., 4.530E-7, 6.670E-7, 0.,
+      5.600E-6, 0., 0., 0.,
+      0., 5.330E-6, 7.910E-6, 0.,
+      0., 1.820E-7, 2.630E-7, 0./
C
C      DATA INDX/6,7,4,10,1,3,8,10,11/
C-----
C
C      If solution analysis, read in sample area and calibration.
C      IF (NTYPE.EQ. 1) THEN
C      CALL MGASOL
C      DEPTH = VOLUME/SURFC
C      DD = (XDIST +1.+ .5*DEPTH)**2
C      ATFCT = SQRT(DD/(DD+.159*SURFC))
C      NOPFLG(10) = 1
C      PUGPSC = .01
C      ENDIF
C
C      !Initialize parameters
C      NPK = 10
C      LMT = 5.
C      IREPET = 0
C      DELSLP = 0.0
C      DELCRV = 0.0
C      DELCD = 0.0
C      DELPU = 0.0
C      ITER = 0
C      ICNT = 0
C      OLDEL = 0.0
C      NPB = 0
C      PBMAS = SHAPC(9)
C      IF (PBMAS.GT. 0.) NPB = -1
C      IF (PBMAS.LT. 0.) THEN
C      PBMAS = .1
C      NPB = 1

```

```

      ENDIF
C----->      !Cd flag is initially on. Cd will be determined
C      NCD = 1
C----->      !Pu flag is initially on. Pu will be determined
C      NPU = 1
C      IF (NOPFLG(4) .EQ. 0) THEN
C          !Efficiency slope will be turned on later.
C          NSLP = 0
C          !Efficiency curvature will be turned on later.
C          NCURV = 0
C          PUDEL = .2
C      ELSE
C          IF (NOPFLG(4) .EQ. 1) NSLP = -1
C          IF (NOPFLG(4) .EQ. 2) THEN
C              NCURV = -1
C              WT(8) = .1*WT(8)
C              WT(9) = .1*WT(9)
C              WT(10) = .01*WT(10)
C              WT(11) = .01*WT(11)
C          ENDIF
C          IF (NOPFLG(4) .EQ. 3) THEN
C              PUDEL = .001
C              NSLP = -1
C              NCURV = -1
C              L = 0
C              DO 2 I = 1,7
C                  L = L + 1
C                  IF (L .EQ. 5) L = 8
C                  WT(L) = .1*WT(L)
C              CONTINUE
C          ENDIF
C      ENDIF
C      !Initially assume equilibrium and no U present.
C      NU = 0
C      !Initially assume 3 isotopic components
C      NUM = 3
C      NM = 3
C      !Fraction of sample considered to be Pu.
C      FRACX = 1.-PUFRAC
C      DO 4 J = 1,11
C      DO 4 K = 1,4
C          !Transfer branching coef. into arrays.
C          BR(K,J) = DAT(K,J)
C          !Correct for 238Pu x rays
C          IF (K .EQ. 2 .AND. (J .EQ. 2 .OR. J .EQ. 5))
C              BR(K,J) = BR(K,J) * (1.+ANS(1))
C          +
C          CONTINUE
C          !Default det. volume
C          IF (SHAPC(11) .EQ. 0.) SHAPC(11) = 3.
C          !Default value for efficiency slope
C          SHAPC(12) = 7.2+.126*SHAPC(11)
C          !Default efficiency curvature
C          SHAPC(13) = -.935-.0037*SHAPC(11)
C          IF (NOPFLG(9) .NE. 0.) THEN
C              !Use input value for CDABS
C              NCD = 0
C              !Reduce weight of 59 keV equation.
C              IF (WT(1) .GT. 2./CNTS(1)) WT(1) = 2./CNTS(1)
C          ENDIF

```

```

C      IF (NOPFLG(10) .NE. 0) THEN
C          !Use input value for PUGPSC
C          NPU = 0
C          IF (PUGPSC .LT. .01) THEN
C              !Ignore U fluorescence
C              NOPFLG(2) = 0
C              ICMP(10) = -1
C          END IF
C      END IF
C      IF ((ICMP(10) .GT. 0) .OR. (ICMP(8) .NE. 0)) THEN
C          !U X Ray fluorescence is present
C          NU = 1
C          NUM = 4
C          !Transfer 94 keV X Ray coeff.
C          BR(4,2) = BR(1,2)
C          !Transfer 111 keV X Ray coeff.
C          BR(4,5) = BR(1,5)
C          !Reduce weight of 148 keV pk.
C          IF (NOPFLG(1) .NE. 0) WT(7) = .001*WT(7)
C      ELSE
C          IF (ICMP(7) .EQ. 0) GOTO 12
C          !237U not in equilibrium with 241Pu
C          NU = -1
C          !One additional component now
C          NUM = 4
C          NM = 4
C          DO 10 I = 1,5
C              L = INDX(I+4)
C          !Transfer 237U coeff. to a separate column
C          BR(4,L) = BR(2,L)
C          !Remove coeff. from 241Pu column
C          BR(2,L) = 0.0
C          CONTINUE
C      END IF
C      !Get initial estimate of 239Pu, 241Pu, 241Am, and
C      ----> d 237U
C          ANSR(4) = 0.0
C          DO 16 J = 1,NM
C              K = INDX(J)
C              ELOG = ALOG(ENRG(K))
C              PUMAS = PUGPSC * PUMU(K)
C              CORR = EXP(-CDABS*CDMU(K))/PUMAS*(1.-EXP(-PUMAS))*EXP(SHAPC(12)
C          + * (ELOG-4.605)+SHAPC(13)*(ELOG**2-21.207))
C          !Total correction factor
C          !Initial answers in dpm.
C          ANSR(J) = CNTS(K) / BR(J,K) / CORR
C          CONTINUE
C      16
C      IF (NOOUT .GT. 0) THEN
C          WRITE (NDEV2,9102)
C          9102 FORMAT (' CD DELCD PU DELPU ESLP ',
C          + ' DELSLP ECURV DELCURV')
C          IF (NOPFLG(5) .GT. 0) THEN
C              WRITE (NDEV3,9102)
C              WRITE (NDEV3,9104) CDABS,DELCD,PUGPSC,DELPU,ESLP,DELSLP,
C          + ECURV,DELCRV
C          ENDIF
C          WRITE (NDEV2,9104) CDABS,DELCD,PUGPSC,DELPU,ESLP,DELSLP,
C          + ECURV,DELCRV
C      ENDIF
C      DO 30 J = 1,11
C          !Fe and Pb absorption

```



```

      ELOG = ALOG(ENRG(J))
      FEMU = EXP(28.27958-10.80034*ELOG+.959801*ELOG**2)
      FECOF = EXP(-FEMU*SHAPC(14))
      PBCOF = EXP(-PBMU(J) *PBMAS)
C      CDCOF = EXP(-CDMU(J) *CDABS)
      SHFTMU = .97
      IF (ENRG(J) .GT. 120.) SHFTMU = .956
      PUMAS = PUGPSC *PUMU(J)*(PUFRAC +SHFTMU*FRACX)
      PUEXP = EXP(-PUMAS)
C      PUCOF = (1.-PUEXP)/PUMAS
C      !det. eff
      EFF1 = EXP(SHAPC(12)*(ELOG-4.605)+SHAPC(13)*(ELOG**2-21.207))
      EN100 = ENRG(J) - 100.
C      !Change from nominal curve.
      EFF2 = (1. +ESLP*EN100 +ECURV*EN100**2)
C      !Do following only for 59 keV peak
      IF (J .EQ. 1) THEN
C      !Cannot be less than 1.0
        IF (EFF1 .LT. 1.0) EFF1 = 1.0
C      !No eff. change between 60 keV and 100 keV
        EFF2 = 1.0
      ENDIF
C      TEMP(J) = FECOF*CDCOF *PBCOF*PUCOF *EFF1 *EFF2
      TEMP2(J) = EFF1 *EFF2
C      !Composite correction
      CORR = TEMP(J) *WT(J)
C      ASUM = 0.0
      DO 20 K = 1,4
C      !Calculate coefficient
        ACOEF(J,K) = BR(K,J) *CORR
C      !Total dpms.
        ASUM = ASUM + BR(K,J) *ANSR(K)
20      CONTINUE
C      CORR = CORR * ASUM
      NCMP = NUM
C      !Cd not to be evaluated
      IF (NCD .GT. 0) THEN
        NCMP = NCMP + 1
C      !Cd amount coefficient
        ACOEF(J,NCMP) = -CDMU(J) * CORR
      ENDIF
C      IF (NPB .GT. 0) THEN
        NCMP = NCMP + 1
        ACOEF(J,NCMP) = -PBMU(J) * CORR
      ENDIF
C      !Pu not to be evaluated
      IF (NPU .GT. 0) THEN
        NCMP = NCMP + 1
        PUG = PUGPSC
C      !O.K. to use standard equation
      IF (PUGPSC .GT. 1.5) THEN
C      !Use modified equation due to
        PUG = PUGPSC**3
C      !approach to Pu saturation.
        PUMAS = PUMU(J)*1.5
        PUEXP = EXP(-PUMAS)
      ENDIF
C      !Pu coefficient
      ACOEF(J,NCMP)=CORR/PUCOF/PUG*(PUEXP-(1.-PUEXP)/PUMAS)
C      !Eff. slope not determined

```

```

      ENDIF
      IF (NSLP .GT. 0) THEN
        NCMP = NCMP + 1
C      !Eff. slope coefficient
        ACOEF(J,NCMP) = CORR/EFF2 *EN100
      ENDIF
C      !Eff. curvature not to be determined
      IF (NCURV .LE. 0) GOTO 30
      NCMP = NCMP + 1
C      !Eff. curvature coefficient
      ACOEF(J,NCMP) = ACOEF(J,NCMP-1) *EN100
C      !60 keV peak not used to calc. curv
      IF (J .EQ. 1) ACOEF(1,NCMP) = 0.0
30      CONTINUE
C      DO 40 J = 1,NCMP
      XSUM(J) = 0.0
      DO 40 K = 1,NCMP
40      XAPHA(J,K) = 0.0
C      DO 42 J = 1,NPK
      DO 41 K = 1,NCMP
C      XSUM(K) = XSUM(K) +ACOE(J,K) * CNTS(J) * WT(J)
C      DO 41 L = 1,NCMP
41      XAPHA(K,L) = XAPHA(K,L) +ACOE(J,K) *ACOE(J,L)
42      CONTINUE
C      !Invert matrix
      CALL MATINV (XAPHA,9,NCMP,DET)
C      IF (DET .EQ. 0.) THEN
        ERRFLG(1) = 7
        RETURN
      ENDIF
C      DO 44 K = 1,NCMP
      ANSR(K) = 0.0
      DO 43 L = 1,NCMP
C      !Calculate answers
43      ANSR(K) = ANSR(K) +XSUM(L) *XAPHA(K,L)
44      CONTINUE
C      DO 48 J = 1,11
      RM(J) = CNTS(J) * WT(J)
      DO 46 K = 1,NCMP
C      !Residual
46      RM(J) = RM(J) - ACOEF(J,K) * ANSR(K)
C      CONTINUE
C      ICNVRG = 1
      NCMP = NUM
      DELCD = 0.0
      ADELCD = 0.0
C      !Cd was not determined
      IF (NCD .EQ. 0) GOTO 50
      NCMP = NCMP + 1
      DELCD = ANSR(NCMP)
      ADELCD = ABS(DELCD)
C      !Not converged.
      IF (ADELCD .GT. .005) ICNVRG = 0
C      !Increment Cd answer
      CDABS = CDABS+DELCD*(1.-ADELCD/(CDABS+ADELCD+.1))
      IF (CDABS .GT. 0.0) GOTO 50

```



```

C          !CDABS cannot be negative.
C          CDABS = 0.0
C          !Increment counter
C          ICNT = ICNT + 1
C          !Turn off Cd determination
C          IF (ICNT .EQ. 4) NCD = 0
50          DELPB = 0.0
          ADELPB = 0.0
          IF (NPB .GT. 0) THEN
            NCMP = NCMP + 1
            DELPB = ANSR(NCMP)
            ADELPB = ABS(DELPB)
            IF (ADELPB .GT. .1) ICNVRG = 0
            PBMA = PBMA + DELPB*(1.-ADELPB/(PBMA+ADELPB+.1))
            IF (NOOUT .GT. 0) WRITE (NDEV2,*) ' PBMA= ',PBMA
            IF (PBMA .LT. 0.) THEN
              PBMA = 0.0
              NPB = -1
            ENDIF
          ENDIF
C          DELPU = 0.0
C          !Pu was not determined
C          IF (NPU .EQ. 0) GOTO 54
            NCMP = NCMP + 1
            DELPU = ANSR(NCMP)
            ADEL = ABS(DELPB)
            DELP = DELPU*(1.-ADEL/(PUGPSC+ADEL+.1))
            IF ((ABS((DELP+OLDEL)/DELP)) .LT. .1) DELP = 0.5*DELP
            !Increment Pu answer.
            PUGPSC = PUGPSC + DELP
            OLDEL = DELPU
C          !Check upper limit
C          IF (PUGPSC .LT. LMT) GOTO 52
            !Maximum value
C          PUGPSC = LMT
            IF (NOPFLG(4) .EQ. 3) ICNVRG = 0
            GOTO 54
C          !Check lower limit
52          IF (PUGPSC .GT. .002) GOTO 53
            PUGPSC = .002
            IF (NOPFLG(4) .EQ. 3) ICNVRG = 0
            GOTO 54
C          !Not converged
53          IF (ABS(DELPB) .GT. PUDEL) ICNVRG = 0
C          Determine if analysis is for solution concentration
54          IF (NTYPE .EQ. 1) THEN
            SUMANS = ANSR(1)/1.735E8+ANS(3)*.09*ANSR(2)/6.35E8+ANSR(2)/
              2.89E11
            PUGPSC = CCNST * SUMANS * CDFCT/(SURF*TIME*ATFCT)
            IF (ITER .LT. 2) ICNVRG = 0
          ENDIF
C          !Eff. slope not determined
C          IF (NSLP .LE. 0) GOTO 55
            NCMP = NCMP + 1
            DELSLP = ANSR(NCMP)
C          !Absolute value of ESLP change
C          ADEL = ABS(DELSLP)
C          !Check for convergence
C          IF (ADEL .GT. .0002) ICNVRG = 0
C          !Increment slope
C          ESLP=ESLP+DELSLP*(1.-ADEL/(ABS(ESLP)+.001+ADEL))
55          DELCRV = 0.0

```

```

C          !Eff. curvature not determined
C          IF (NCURV .LE. 0) GOTO 56
            NCMP = NCMP + 1
            DELCRV = ANSR(NCMP)
C          !Absolute value of ECURV change
C          ADEL = ABS(DELCRV)
C          !Increment
C          ECURV = ECURV + DELCRV*(1.-ADEL/(ABS(ECURV)+.00001+ADEL))
            IF (ADEL .GT. .000002) ICNVRG = 0
56          CONTINUE
C          IF (NOOUT .GT. 0) THEN
            WRITE (NDEV2,9104) CDABS,DELCD,PUGPSC,DELPB,ESLP,
              DELSLP,ECURV,DELCRV
C          + IF (NOPFLG(5) .GT. 0)
            WRITE (NDEV3,9104) CDABS,DELCD,PUGPSC,DELPB,ESLP,
              DELSLP,ECURV,DELCRV
C          +
9104          FORMAT (4F8.3,4F10.7)
            ENDIF
C          !Has converged
C          IF ((ABS(NSLP) .EQ. 1) .AND. (ICNVRG .EQ. 1)) GOTO 90
            !O.K. to calc. ESLP
C          IF ((ABS(DELCD) .LT. .1) .AND. (NSLP .EQ. 0)) NSLP = 1
C          ITER = ITER + 1
            IF (ITER .LT. 20) GOTO 18
            ERRFLG(14) = 1
            IF (ITER .GT. 22) NSLP = 0
            IF (NOOUT .GT. 0) WRITE (NDEV3,9110)
            FORMAT (' WARNING: DID NOT CONVERGE IN "MGAABS" ROUTINE')
9110          C
90          IF (NCURV .NE. 0) GOTO 92
            NCURV = 1
            IF (NOPFLG(4) .NE. 3) PUDEL = .1
            GOTO 18
92          RMU = RM(2) + RM(5) - RM(3)
            IF ((ICMP(10) .NE. 0) .OR. (RMU .LT. 5.)) GOTO 95
C          !Presence of U detected.
C          NOPFLG(2) = 1
            ICMP(10) = 1
            IF (SPC2ID(1:1) .NE. ' ') ICMP(11) = 1
C          !Restart analysis
C          IF (NOPFLG(8) .EQ. 0) NOPFLG(8) = 3
            IF (NOOUT .GT. 0) WRITE (NDEV2,9120)
            FORMAT (' Presence of U detected in "MGAABS" routine.')
9120          RETURN
C          IF (NPU .EQ. 0) GOTO 96
C          !Test Pu boundary conditions
C          NPU = 0
            GOTO 18
C          IF (NCURV .EQ. -1) GOTO 97
            NCURV = -1
            GOTO 18
C          IF (ADELCD .GT. .05) GOTO 18
C          IF (IREPET .EQ. 0) THEN
            !Check if 267 keV effic. is more than 10% out.
            PCT267 = 100.*RM(11)/(WT(11)*CNTS(11))
            IF (PCT267 .GT. 10.) THEN
              !Include 267 keV peak in efficiency calc.
              IF (NOPFLG(4) .EQ. 0) NPK = 11
              LMT = 8.
              IF (NOPFLG(9) .EQ. 0) NCD = 1
            ENDIF
          ENDIF

```

```

      IF (NOPFLG(10) .EQ. 0) NPU = 1
      IF (NOPFLG(4) .EQ. 0) THEN
        NSLP = 1
        NCURV = 1
      ENDIF
C      !Reduce the weight of the 59 keV peak.
      WT(1) = .1*WT(1)
      IREPET = 1
      ITER = 10
      GOTO 18
    ENDIF
  ENDIF
C
  IF (NPB .EQ. 0) THEN
    IF (ABS(ECURV) .GT. .00004) THEN .
      ERRFLG(15) = 1
      IF (NOOUT .GT. 0) THEN
        WRITE (NDEV2,9103)
        IF (NOPFLG(5) .GT. 0) WRITE (NDEV2,9103)
9103      FORMAT (' *** Det. Eff. Curvature is excessive. ***',/
        + '      Is Pb absorber being used???' )
      ENDIF
    ENDIF
  ENDIF
  IF (NOOUT .NE. 0) THEN
    WRITE (NDEV2,9130)
9130    FORMAT (' Energy Counts % Err. Residual Tot. Eff.',
    + '      Det. Eff.')
    WRITE (NDEV2,9090) (ENRG(J),CNTS(J),100./(WT(J)*CNTS(J)),
    + RM(J),TEMP(J),TEMP2(J), J=1,11)
9090    FORMAT (OPF9.3,F9.0,F7.2,F10.1,1P2E12.3)
    IF (NOPFLG(5) .GT. 0) THEN
      WRITE (NDEV3,9130)
      WRITE (NDEV3,9090) (ENRG(J),CNTS(J),100./(WT(J)*CNTS(J)),
      + RM(J),TEMP(J),TEMP2(J), J=1,11)
    ENDIF
  ENDIF
C
C Check if 237U is out of equilibrium with respect to its 241Pu parent.
C
  IF (ICMP(7) .EQ. 0) THEN
    RMU = RM(2) + RM(7) + RM(5) - RM(3) - RM(8)
    IF (ICMP(10) .NE. 0) RMU = 2. * (RM(7) - RM(3) - RM(8))
    IF (ABS(RMU) .GT. 20.) THEN
C
      IF (NOOUT .GT. 0) THEN
        WRITE (NDEV2,9140)
        IF (IPRINT .EQ. 'Y') WRITE (NDEV3,9140)
9140      FORMAT (' 237U-241Pu equilibrium imbalance detected in MGAABS.'/
      + '  "/F" option will be set and the analysis restarted.' )
      ENDIF
C
      NOPFLG(1) = -1
      ICMP(7) = 1
      !Set flag to "Fresh Sample" analysis
C
      !Set repeat flag
      IF (NOPFLG(8) .EQ. 0) NOPFLG(8) = 4
      RETURN
    ENDIF
  ENDIF
C
98  IF (((RM(1)-RM(4)) .LT. 2.) .OR. (NOPFLG(2) .GT. 0)) GOTO 100
C
C WRITE (NDEV2,9142)
C9142  FORMAT (' Sample may contain a high Z element affecting the ',
C + 'results.' ) ' Continue analysis? (Or abort and repeat using'
C + '  "/X" option) (Y/N): ', $)

```

```

C ACCEPT 9144, NOP
C9144  FORMAT (A1)
C      IF (NOP .NE. 'Y') STOP 'Try repeating analysis using "/X" option'
C
100  IF (TEMP2(9)/TEMP2(10) .LT. 1.02) THEN
      ERRFLG(15) = ERRFLG(15) + 1
      IF (NOOUT .NE. 0) WRITE (NDEV3,9146)
9146  FORMAT (' WARNING: Det. slope in 208keV region too shallow.',
      + '      Something is wrong with fit of efficiency curve.')
    ENDIF
C
      RAT = CNTS(1)/CNTS(2)
      IF (RAT .LT. .2) THEN
C
        ERRFLG(9) = .2/RAT
        ERRFLG(9) = 1
        IF (NOPFLG(8) .EQ. 0 .AND. NOOUT .EQ. 1) THEN
          WRITE (NDEV2,9200)
          IF (IPRINT .EQ. 'Y') WRITE (NDEV3,9200)
9200  FORMAT (' ***WARNING: 59 keV peak is weak compared to 100 keV',
      + ' peaks.' ) ' Please reduce absorbers if possible.' )
        END IF
      ENDIF
C
      IF ((RAT .GT. 5) .AND. (IDT1 .GT. 20)) THEN
C
        ERRFLG(9) = .2 * RAT
        ERRFLG(9) = 2
        IF (NOPFLG(8) .EQ. 0 .AND. NOOUT .EQ. 1) WRITE (NDEV2,9210)
9210  FORMAT (' ***WARNING: 59 keV peak is quite intense compared to ',
      + '100 keV peaks.' ) ' Please use more Cd absorber.' )
      ENDIF
C
C Calculate the residual slope in the 100 keV region based on the
C intensity residuals of the 94 and 111 keV peaks.
C
C      !Residual (fractional) of 94 keV peak
C      R94 = RM(2) / (CNTS(2) * WT(2))
C      !Residual (fractional) of 111 keV peak
C      R111 = RM(5) / (CNTS(5) * WT(5))
C      !Residual efficiency slope (frac./channel)
C      RSLP = (R111 - R94) * GAIN / 16.34
C
C Estimate initial DPM's for the various isotopes for use in MGA100
C      !Estimate for 238Pu dpm.
C      ANS(1) = .2 * AREA(8)/AREA(7)*ANSR(2)
C      !Estimate for 239Pu
C      ANS(2) = ANSR(1)
C      !Estimate for 241Pu
C      ANS(4) = ANSR(2)
C      !Estimate for 237U
C      IF (ICMP(7) .NE. 0) ANS(7) = ANSR(4)
C      !Estimate for 240Pu
C      ANS(3) = ANS(3) * .09 * ANS(4)
C      !Estimate for 241Am
C      ANS(5) = ANSR(3)
C      !Estimate for X Ray dpm.
C      ANS(6) = ANS(6) * .061 * ANS(4)
C
C SHAPC(9) = PBMAS
C RETURN
C
END

```

SUBROUTINE MGA45

MGA45
=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Title : MGA45
Author : Ray Gunnink
Date : 2/4/88
Revision : 8.0 (13-March-91)
Language : Microsoft Fortran Ver. 4.01

Abstract :

This routine was abstracted from MGAPKS to decrease its size. The routine is called only when analyzing freshly processed samples, that is, samples in which the ^{237}U is not in secular equilibrium with its ^{241}Pu parent and the ^{241}Am abundance is sufficiently low so that the peaks in the 38-51 keV region can be analyzed with good precision. This routine first obtains the areas of the 38 and 51 keV peaks due to ^{239}Pu and then reads in the 43-45 keV region to obtain the areas of the 43 and 45 keV peaks due to ^{238}Pu and ^{240}Pu respectively. Several small interferences found in this region must also be taken into account. We strongly suggest that anyone wishing to use this routine to analyze freshly separated Pu samples, contact the authors of MGA since this routine has been tested on only a limited number of samples.

Arguments :

Routines
Called : TRANSFR
GFIT
SPRAM
MAXVAL
STRIP1
NCNTS

Libraries
Used :

Revision History (Include date, revision I.D., and details) *

Date: 2-Nov-84

Update: 4-Feb-88
Modified for PC Microsoft FORTRAN
3-Mar-88
Changed to allow greater variation in gain and
zero-channel energy

COMMON and variable declarations

\$INCLUDE: 'MGACOM.CMN'

```

C      DO 10 J = 1,4
C          AREA(J) = 1.          !Default values
C          ERR(J) = 1.
C      ENX = 45.                  !Get peak width parameter
C      CALL SPRAM (ENX,GAIN,SHAPC,SS,ALPHA,F12M,EXP1)
C      NST = 36.75/GAIN + ZERO    !Read in 38 keV peak region
C      CALL TRANSFR (50,YDATA(NST),Y)
C      IS = 1.52/GAIN             !Find channel with maximum count
C      CALL MAXVAL (IS,IS+5,JPY,YMAX,Y)
C      NBND5 = SQRT(-7.6/ALPHA)
C      NE = JPY - NBND5 -1        !Start of peak
C      IS = .77/GAIN              !Start of front-side bkgnd
C      NS = NE - IS
C      MS = JPY + NBND5           !End of peak
C      ME = MS + IS              !End of aft-side bkgnd
C      CALL NCNTS (NS,NE,MS,ME,NPTS,BGD,A38,ER38,AVEBG,Y,Y)
C      IF (ER38/A38 .LT. .3) GOTO 20
C      IF (NOOUT .GT. 0) WRITE (NDEV2,9000)
C      FORMAT (' 38 keV peak is too weak. 38-51 keV region not used.')
C      NOPFLG(1) = -1            !Set flag
C      RETURN
C
C      AREA(1) = A38
C      ERR(1) = ER38
C      NST = 49.5/GAIN + ZERO     !Read in 51 keV peak region
C      CALL TRANSFR (45,YDATA(NST),Y)
C      IS = 1.14/GAIN            !Find channel with maximum count
C      CALL MAXVAL (IS,IS+40,JPY,YMAX,Y)
C      NE = JPY - NBND5 -4       !Start of peak
C      NS = NE - .62/GAIN        !Start of front-side bkgnd

```

```

C      MS = JPK + NBND5 +1      !End of peak
C      ME = MS + .4/GAIN        !End of aft-side bkgrd
C      CALL NCNTS (NS,NE,MS,ME,NPTS,BGD,AREA(4),ERR(4),AVEBG,Y,Y)
C      IF (AREA(4) .EQ. 0.) AREA(4) = 1.
C
C      Calculate the efficiency slope in the 35-60 keV region using the 38 and
C      51 keV peak of 239Pu.
C      ESLP1 = 3.458 *ALOG(0.37 *AREA(4) /AREA(1))
C      NST = 42./GAIN + ZERO      !Read in 41-47 keV region
C      CALL TRANSFR (100,YDATA(NST),Y)
C      IS = 3./GAIN                !Find 45 keV peak channel position
C      CALL MAXVAL (IS,IS+10,JPK,YMAX,Y)
C      CALL GFIT (JPK,4,ALPHA,EXP1,EXP2,REFPK,HT,ER,Y)
C
C      Subtract out the 44.19 and 46.23 keV peaks due to 241Pu and 239Pu.
C      PHT = 6.5E-4*(Y(21)-Y(12))*AREA(7)/AREA(8)
C      PPOS = REFPK - 1.0425/GAIN  !Position of 44.19 keV peak
C      IS = 2./GAIN                !Strip out peak
C      WL = 0.
C      CALL STRIP1 (IS,IS+12,PPOS,PHT,ALPHA,EXP1,EXP2,WL,Y)
C      C45 = SQRT (1.133 * SS)      !Area of 45 keV peak
C      PHT = .029*AREA(4)/C45*EXP(-0.1109*ESLP1)
C      PPOS = REFPK + .9675/GAIN    !Position of 46.23 peak
C      IS = 3.8/GAIN                !Strip out peak
C      CALL STRIP1 (IS,IS+15,PPOS,PHT,ALPHA,EXP1,EXP2,WL,Y)
C      IS = 2.42/GAIN              !239Np not detected.
C      IF (HT61 .LE. 0.) GO TO 30
C
C      Calculate 44.7 keV peak height of 239Np and strip out.
C      HT44 = 0.1 * HT61 * EXP (-0.32 * ESLP1)
C      IF (HT44 .LE. 0.0) GO TO 30
C      PPOS = REFPK - .525/GAIN      !Position of 44.7 keV peak
C      CALL STRIP1 (IS,IS+12,PPOS,HT44,ALPHA,EXP1,EXP2,WL,Y)
C
C      Determine integration limits for the 43 and 45 keV peaks.
C      NE1 = JPK - IS                !Start of 43 keV peak
C      IS = 1.9/GAIN                !Find the minimum sum of 4 successive
C      SMIN = Y(IS)+Y(IS+1)+Y(IS+2)+Y(IS+3)
C      SUM = SMIN                    !channels counts in the region
C      MS1 = IS+2                    !between the 43 and 45 keV peaks
C      DO 35 J = IS,IS+15

```

```

C      SUM = SUM + Y(J+4) - Y(J)    !Subtract low-side and add new
C      IF (SMIN .LE. SUM) GO TO 35  !Compare with previous sum
C      SMIN = SUM                    !Preserve lessor value
C      MS1 = J                        !Preserve position also
35  CONTINUE
C      Integrate the 43 keV peak
C      NS1 = NE1 - .38/GAIN          !Start of front-side bkgrd
C      IF (NS1 .LT. 1) NS1 = 1      !Outside of bounds
C      MS1 = MS1 + 1                 !End of 38 keV peak
C      ME1 = MS1 + 2                 !End of aft-side bkgrd
C      CALL NCNTS (NS1,NE1,MS1,ME1,ND,BGD,AREA(2),ERR(2),AVEBG,Y,Y(101))
C      IF (AREA(2) .EQ. 0.0) AREA(2) = 1.
C
C      Integrate the 45 keV peak
C      MS = JPK + NBND5 +1          !End of 45 keV peak
C      ME = MS + 2                   !End of aft-side bkgrd
C      CALL NCNTS (MS1,ME1,MS,ME,ND,BGD,AREA(3),ERR(3),AVEBG,Y,Y(101))
C      IF (AREA(3) .EQ. 0.0) AREA(3) = 1.
C
C      RETURN
C
C      END

```

SUBROUTINE MGA100

```
C *****
C
C                                     MGA100
C                                     =====
C
C
C Developed for : Safeguards Technology Program
C                 Nuclear Chemistry Division
C                 Lawrence Livermore National Laboratory
C                 Livermore, CA. 94550
C
C
C *****
C
C Title       : MGA100
C Author      : Ray Gunnink
C Date        : 5/24/89
C Language    : Microsoft Fortran   Ver. 4.01
C Revision    : 9.0 (Jan-1992)
C
C *****
C
C Abstract :
C
C This subroutine of the MGA program prepares the set of
C equations based on the 100 Kev peak grouping.
C
C *****
C
C Arguments :
C
C *****
C
C Routines
C Called      : BWF
C
C Libraries
C Used        :
C
C *****
C
C Revision History ( Include date,revision I.D., and details )
C *****
C
C Date of last revision: 4-Jan-1991
C
C *****
C
C COMMON and variable declarations
C *****
C
C DIMENSION NFLG(23),EN(23),PKPOS(23),EFACT(8,17),T1HLF(5),
C + CORR(23),BRZ35(6),INDEX(6),BRANCH(8,17),B235(6),BRPKS(7)
C -----
C $INCLUDE: 'MGACOM.COM'
C
C DATA NFLG/0,102,102,0,108,104,0,0,107,0,108,0,0,0,108,0,0,
C + 104,104,0,0,108,108/
C DATA T1HLF/87.7,24100.,8580.,14.3,433./
C DATA INDEX/1.0,0.0,1.1/
```

```

DATA EN/86.5,94.66,95.858,96.16,97.071,98.443,98.79,98.956,
+ 99.533,99.864,101.063,102.963,103.035,103.679,103.742,
+ 103.86,104.242,93.36,95.88,99.27,102.275,104.83,105.61/
C
C
238Pu 239Pu 240Pu 241Pu 241Am x ray 237 237Np
DATA BRANCH/.0, .0, .0, .0, .0, .0, .0, .0, .140,
+ 1.15 E-6,4.200E-5,6.800E-7,2.979E-6, .0, .0, .0, .0, .0931,
+ .0, .0, .0, .0, .0, .0, .0, .0, .029,
+ .0,3.680E-7, .0, .0, .0, .0, .0, .0, .0,
+ .0, .0, .0,3.802E-6,1.222E-5, .0, .0, .0, .0,
+ 1.850E-6,6.820E-5,1.000E-6,4.843E-6, .0, .0, .0, .0, .1400,
+ .0,1.220E-5, .0, .0, .0, .0, .0, .0, .0,
+ .0, .0, .0, .0, .0,2.011E-4, .0, .0, .0,
+ .0, .0, .0, .0, .0, .0,4.13E-5, .0, .0,
+ 7.100E-5, .0, .0, .0, .0, .0, .0, .0, .0,
+ .0, .0, .0, .0,6.118E-6,1.970E-5, .0, .0, .0, .0,
+ .0, .0, .0, .0, .0,1.945E-4, .0, .0, .0, .0,
+ .0,2.120E-6, .0, .0, .0, .0, .0, .0, .0,
+ .0, .0, .0, .0,0.985E-6, .0, .0, .0, .0,
+ .0, .0, .0, .0, .0, .0,6.60E-5, .0, .0,
+ .0, .0, .0, .0, .0, .0, .0, .0, .0083,
+ .0, .0, .0,6.960E-5, .0, .0, .0, .0, .0 /
DATA B235/.05, .008, .0025, .0042, .0052, .011/
DATA BRPKS/ 3.830E-04, 4.430E-04, 2.670E-04, 6.290E-05,
+ 1.870E-06, 9.260E-06, 5.330E-06/
-----
C
C
C ASLP = SHAPC(2) / GAIN !Change in peak width parameter per cha
C S100 = 100./GAIN -ST !Position of a 100 keV peak. Reference
C RATIO = 1. / SHAPC(6) !Peak Height to Area Ratio.
C TAMP2 = SHAPC(7) !Long-Term Tailing Amplitude
C EXP3 = SHAPC(8) * GAIN !Long-Term Tailing Slope
C
DO 901 J = 1,17
DO 901 K = 1,8
901 EFACT(K,J) = BRANCH(K,J)
DO 902 I = 1,6
902 BR235(I) = B235(I)
BRNCH(I) = BRPKS(I)
BRNCH(7) = BRPKS(7)
NDEV = 1
C !Open file of constants
OPEN (UNIT=NDEV,FILE='\\PUIS\MGA\GRP100.DAT',STATUS='OLD',
+ ACCESS='SEQUENTIAL',ERR=904)
904 GOTO 905
CLOSE (NDEV)
OPEN (UNIT=NDEV,FILE='GRP100.DAT',STATUS='OLD',
+ ACCESS='SEQUENTIAL',ERR=906)
905 DO 1 I = 1,17
C !Read in file
READ (NDEV,9010) EN(I),(EFACT(J,I), J = 1,8)
9010 FORMAT (F10.3,8E10.3)
1 CONTINUE
READ (NDEV,9011)(EN(I+17),BR235(I),I=1,6)
9011 FORMAT (2E12.3)
READ (NDEV,9012) (BRNCH(J), J = 1,7)
9012 FORMAT (E12.3)
906 CLOSE (NDEV)
C
SUMA = 0.0
C !Estimate the Am/Pu atom ratio.

```

```

C      DO 2 I = 1,5
C          !Used later to compute small Am
C          ATOMS = ANS(I) * T1HLF(I)
C          !alpha-induced X ray response
C          SUMA = SUMA + ATOMS
C          !at 102.03 keV.
C      CONTINUE
C      IF (PU242 .GT. 0.) SUMA = SUMA *(1.+PU242/(100.-PU242))
C      AMPU = .62*ATOMS / (SUMA - ATOMS)
C          !Approx. position of Am peak relative to Pu peak
C      ISHFT1 = 1.711/GAIN+0.5
C      ISHFT2 = 2.74 /GAIN+0.5
C
C      N242 = 0
C      IF (PU242 .GT. 50.) THEN
C          !Prepare to analyze for 242Pu content
C          EN(18) = 103.5
C          !Branching intensity of 103.5 keV peak
C          BR235(1) = 2.63E-5
C          NFLG(18) = 0
C          N242 = 1
C          ICMP(9) = 0
C      ENDIF
C
C      Determine exact position of each peak.
C
C      DO 3 J = 2,23
C          PKPOS(J) = EN(J) / GAIN - ST
C
C      Adjust the branching coefficients to account for the change in the
C      efficiency in the 94-104 keV region due to changing detector efficiency,
C      attenuation by absorbers (Cd should be used ) finally, due to attenuations
C      caused by the sample itself. The detector efficiency is defined to be 1.0
C      at 100 keV.
C
C      DO 4 J = 1,23
C          ENLOG = ALOG (EN(J))
C          ENLSQ = ENLOG**2
C          FEMU = EXP(28.27958-10.80034*ENLOG+.959801*ENLSQ)
C          FECOF = EXP(-FEMU*SHAPC(14))
C          !Attenuation coefficient for Cd
C          CDMU = EXP(12.452-2.6134*ENLOG)
C          PBMU = EXP(13.21214-2.503051*ENLOG)
C          !Attenuation coefficient for Pu
C          PUMU = EXP(12.2466-2.5326*ENLOG)
C          UMU = .97*PUMU
C          !Total coeff.
C          PUMUX = PUGPSC *(PUMU*PUFRAC + UMU *(1.-PUFRAC))
C          !Efficiency = 1.0 at 100keV
C          EN100 = EN(J)-100.
C          DETEFF = EXP(SHAPC(12)*(ENLOG-4.605) +SHAPC(13)*(ENLSQ-21.207))
C          !Det. Eff. Calculation
C          DETEFF = DETEFF *(1.+ESLP*EN100+ECURV*EN100**2)
C          !Correction due to Cd attenuation
C          CDCOF = EXP(-CDMU*CDABS)
C          PBCOF = EXP(-PBMU*SHAPC(9))
C          !Total Attenuation
C          CORR(J) = DETEFF*FECOF*CDCOF*PBCOF*(1.-EXP(-PUMUX))/PUMUX
C          !Residual slope (See MGAABS)
C          CORR(J)=CORR(J)*(1.+RSLP*(PKPOS(J)-100.))
C          IF (J .LT. 18) THEN
C      DO 5 K = 1,8
C          !Apply correction factor
C          EFAC(K,J) = EFAC(K,J) * CORR(J)

```

```

5      CONTINUE
C      ELSE
C          BR235(J-17) = BR235(J-17)*CORR(J)
C      END IF
C      CONTINUE
C          !Approx. correction for 106 keV area.
C          A106 = A106 / CORR(17)
C          !Number of isotopic components.
C          NUMC = 6
C          !ie. 238,239,240,241Pu, 241Am, and X ray fluores
C          !----->cence.
C          !Number of peaks in 100 keV group
C          NP = 17
C          !235U or Th xrays
C          IF ((NOPFLG(2) .GT. 5) .OR. (ICMP(16) .GT. 0)) NP = 23
C          Assume no fluorescence when 235U is weak and sample is dilute.
C          IF ((NOPFLG(2) .LT. 30) .AND. (PUGPSC .LT. .01)) ICMP(10) = -1
C          !Ignore Pu alpha induced fluorescence.
C          IF (PUGPSC .LT. .02) ICMP(6) = 0
C
C          NCMP = 0
C          DO 6 I = 1,16
C          IF (ICMP(I) .LE. 0) GOTO 6
C          !Increment component index counter
C          NCMP = NCMP + 1
C          !Determine column positions of components
C          ICMP(I) = NCMP
C      CONTINUE
C          NDIF = 0
C          IF (N242 .GT. 0) THEN
C              Add another component for 242Pu.
C              NCMP = NCMP + 1
C              ICMP(16) = NCMP
C              NP = 18
C          ELSE
C              IF (NP .EQ. 17 .AND. CNTS(2) .GT. 5000.) THEN
C                  !Add a component to account for differential nonlinearity.
C                  NCMP = NCMP + 1
C                  ICMP(16) = NCMP
C                  TOTCTS = CNTS(2) +CNTS(3)
C                  REFPK = (.94.66*CNTS(2)/TOTCTS+101.063*CNTS(3)/TOTCTS)/
C                  + GAIN - ST
C                  NDIF = 1
C              ENDIF
C          ENDIF
C          !Aged sample. Assume equilibrium.
C          IF (ICMP(7) .EQ. 0) GOTO 8
C          Fresh sample. 237U may not be in equilibrium with 241Pu. Therefore
C          treat 237U as a separate component.
C
C          !Column index for 237U component.
C          NC = 7
C          DO 7 I = 1,14
C          EFAC(NC,I) = 0.0
C          !Transfer 97 keV coeff. to 237U column
C          EFAC(NC,5) = EFAC(4,5)
C          !Remove from 241Pu column
C          EFAC(4,5) = 0.0
C          !Transfer 101 keV coeff. to 237U column
C          EFAC(NC,11) = EFAC(4,11)
C          !Remove from 241 Pu column
C          EFAC(4,11) = 0.0
C

```



```

C      NUMC = 7
C      IS = 1
C      Zero out coefficients of all equations before loading begins.
C
C      DO 10 I = 1,250
C      DO 10 K = 1,16
C      ACOEF(I,K) = 0.0
C
C      IF (ICMP(8) .GT. 0) THEN
C
C      When fitting the 100 keV region, and 237Np is known to be
C      present, an additional component must be include as a degree of freedom.
C      This component is very similar to the U X ray fluorescence component,
C      except that there are a few additional peaks in the region. Two of these
C      peaks are the 237Np X ray at 95.8 keV and a 233Pa peak at 103.86 keV.
C      (Remember, the 92.3 keV peak was already stripped out in MGAPKS subroutine.)
C      Also include the 86 & 312keV peaks as the 1st & 2nd equations of the set.
C
C      !1st and 2nd equations will now be the 86 and 30
C      ---->0(312) keV peaks
C      IS = 3
C      !First "channel" now equals 86 keV peak area
C      YNET(1) = A86
C      !Weight factor
C      WTF = 1. / (ER86+.01*A86)
C      Y(1) = WTF*A86
C      !237Np coef.
C      ACOEF(1,ICMP(8)) = WTF*EFACT(8,1)
C      END IF
C
C      A set of linear equations is now formed. Each equation describes the
C      contributions the components make to the corresponding observed net count
C      in a channel in the 94-100 keV energy region. The matrix of coefficients
C      that is generated correspond to normalized response profiles, one for each
C      degree of freedom.
C
C      DO 30 I = IS,NDPTS
C
C      !Float index
C      XI = I
C
C      The following instructions generate a coefficient (ACOE) for each component
C      (NP) contributing to equation I.
C
C      DO 25 J = 2,NP
C
C      !237Np peaks
C      IF (((J.EQ.3).OR.(J.EQ.16)).AND.(ICMP(8).EQ.0)) GOTO 25
C
C      !Distance from peak position
C      DX = XI - PKPOS(J)
C      DX1 = DX
C
C      !Distance squared
C      DDX = DX * DX
C
C      !Calc. peak width parameter
C      ALPHA = -2.7726/(ASLP*(PKPOS(J)-S100)+SG)
C
C      !Exponent of Gaussian term
C      ADX = ALPHA * DDX
C
C      !Initialize parameters
C      FXX = 0.0
C      EXPN3 = 1.0
C      DELA = 0.0
C      DELB = 0.0
C      EXPN2 = 0.0

```

```

C      !Calc. Gaussian contribution
C      IF (ADX .GT. -12.) FXX = EXP(ADX)
C      !Preserve Gaussian value
C      FX1 = FXX
C      XPON = .4 * ADX
C
C      !Tailing truncation term
C      IF (XPON .GT. -12.) EXPN3=1.-EXP(XPON)
C      !No tailing on high side
C      IF (DX .GE. 0) GOTO 12
C      !Now of aft side of peak
C      DX = -DX
C      !Long term tailing slope
C      TSLP2 = EXP (-EXP3 * DX)
C      XPON = -EXP2*DX
C
C      !Short term tailing slope
C      IF (XPON .GT. -14.) EXPN2=EXP(XPON)
C      !Add in tailing contribution
C      FXX = FXX+(EXP1*EXPN2+TAMP2*TSLP2)*EXPN3
C      !Increment on tailing amplitude
C      DELA = TSLP2 * EXPN3
C      !Increment on tailing slope
C      DELB = -DX * TAMP2 * DELA
C      GOTO 121
C
C      !Assume no pileup with low dead time.
C      12 IF ((IDT1 .LT. 0) .OR. (IDT1 .GT. 5)) THEN
C      L = DX + DEL59
C      IF (L .LT. LMAX59) FXX = FXX + YPILUP(L)
C      END IF
C
C      !Peak height adjustment factor
C      121 RAT = RATIO * SQRT(ALPHA/ALFA)
C      !Not an X ray if NFLG = 0
C      IF (NFLG(J) .GT. 0) THEN
C
C      Compute X ray profile. See Nucl. Inst. & Meths. 143,p145,(1977)
C
C      !X ray width parameter
C      WL = FLOAT (NFLG(J)) * .001 / GAIN
C      CC = SQRT (-ALPHA)
C      GAMA = WL * CC
C      CC = CC * DX
C
C      !Determine value of series
C      CALL BWF (CC)
C      C1 = 1. - .5642 * GAMA
C      C2 = .15915 * GAMA
C      C3 = .25 * GAMA * GAMA
C      C4 = .6366 * GAMA
C
C      !X ray profile
C      AA = (C1-C2*ADX+C3*(1.+2.*ADX))*FX1+C4*CC
C      !Substitute X ray response for gamma ray
C      FXX = FXX - FX1 + AA
C      END IF
C
C      !Adjust height for changing peak width
C      FXX = FXX * RAT
C
C      IF (J .GT. 17) THEN
C      NC = ICMP(9)
C
C      !235U coefs.
C      IF (NC .GT. 0) ACOEF(1,NC) = ACOEF(1,NC)+BR235(J-17)*FXX
C      NC = ICMP(16)
C
C      !Th xrays only
C      IF ((NC .GT. 0) .AND. (INDEX(J-17) .GT. 0))
C      + ACOEF(1,NC)=ACOEF(1,NC)+BR235(J-17)*FXX
C      ELSE
C
C      !Generate component coefficients

```


C
END

```

      DO 15 K = 1,NUMC
      L = 1
C      IF (EFACT(K,J) .EQ. 0) !No contribution to this peak
C      IF (EFACT(K,J) .EQ. 0) GOTO 15
C      !Add response to coefficient
      ACOEF(L,K) = ACOEF(L,K) + EFACT(K,J)*FXX
C      !Position of Am K-alpha1 peak relative to Pu K-A
      ---->2 peak
      LL = L - ISHFT1
      IF (LL .LE. 2) GOTO 15
C      !Add Am X ray fluorescence to Pu column.
      IF ((J .EQ. 15) .AND. (K .EQ. 6)) THEN
      ACOF = AMPU * EFACT(6,15) * FXX
      ACOEF(LL,6) = ACOEF(LL,6) + ACOF
      LL = L + ISHFT2
      IF (LL .LE. NDPTS) ACOEF(LL,6) = ACOEF(LL,6) + ACOF*
      + EFACT(6,15)/EFACT(6,9)
      ENDIF
15    CONTINUE
C
      GPM = 0.0
      DO 18 K = 1,NUMC
C      !Estimated gammas per min for Jth peak
18    GPM = GPM + ANS(K) * EFACT(K,J)
C      !Corresponding peak height
      PHT = GPM * RAT
C      !Column index for short-term tailing amplitude
      NC = ICMP(12)
      IF (NC .GT. 0) ACOEF(I,NC) = ACOEF(I,NC) + EXPN2*EXPN3*PHT
C      !Column index for long-term tailing amplitude
      NC = ICMP(13)
C      !Tail amplitude component
      ACOEF(I,NC) = ACOEF(I,NC) + DELA*PHT
C      !Column index for peak width parameter
      NC = ICMP(14)
      IF (NC .GT. 0) ACOEF(I,NC) = ACOEF(I,NC) + PHT*DDX*FX1
C      NC = ICMP(16)
C      IF (NC .GT. 0 .AND. NDIF .EQ. 1) THEN
C      !Calc. coeff. to account for differential nonlinearity.
C      ACOEF(I,NC) = ACOEF(I,NC) + PHT*DX1*FX1*(FLOAT(I)-REFPK)
C      ENDIF
C      IF ((J .EQ. 2) .OR. (J .EQ. 6)) THEN
C      !U K-alpha X rays
C      !Column index for U X ray component
      NC = ICMP(10)
      IF (NC .GT. 0) ACOEF(I,NC) = ACOEF(I,NC) + EFACT(2,J)*FXX
      END IF
C      !Column index for 237Np-233Pa component
      NC = ICMP(8)
      IF (NC .GT. 0) ACOEF(I,NC) = ACOEF(I,NC) + EFACT(8,J)*FXX
      END IF
C
25    CONTINUE
C
30    CONTINUE
C
      DO 40 I = 1,NDPTS
      YT = YNET(I) + AVEBG
C      !Weighting factor for equation
      WTF = 1.0/((.001*YT + SQRT(YT)))
C      !Weight the equations
      Y(I) = YNET(I) * WTF
      DO 38 K = 1,NCMP
38    ACOEF(I,K) = ACOEF(I,K) * WTF
40    CONTINUE
      RETURN

```

```

C
C      SUBROUTINE MGAEQS
C
C*****
C
C      MGAEQS
C      =====
C*****
C
C      Developed for : Safeguards Technology Program
C                    Nuclear Chemistry Division
C                    Lawrence Livermore National Laboratory
C                    Livermore, CA. 94550
C*****
C
C      Title       : MGAEQS
C      Author      : Ray Gunnink
C      Date        : 5/24/89
C      Language    : Microsoft Fortran  Ver. 4.01
C      Revision    : 8.0 (13-March-1991)
C*****
C
C      Abstract :
C
C      The individual equations corresponding to the peak areas determined
C      in subroutines MGAPKS and MGA45 are added to the equations formed by
C      MGA100. (This accumulated set of equations is then compressed, if
C      data from a second detector is also to be included.)
C=====
C*****
C
C      Arguments :
C*****
C
C      Routines
C      Called   :
C
C      Libraries
C      Used    :
C*****
C
C      Revision History ( Include date, revision I.D., and details ) *
C*****
C
C      Date of last revision: 1-June-1990
C*****
C
C      COMMON and variable declarations *
C*****
C
C      DIMENSION INDX1(8),INDX2(8),EFF(8),EE(10),EN(10),CDMU(10),
C      +          PUMU(10)
C      DIMENSION CUMU(5),PUMLO(5),CUAT(5),PUAT(5),ENLO(5),ERATO(3)
C-----

```

```

C
C $INCLUDE: 'MGACOM.CMN'
C
C-----
C
C      DATA EN/129.296,148.567,152.68,164.58,185.72,203.54,208.,
C      + 267.5,300.1,125.29/
C      DATA CDMU/.7894,.5668,.533,.452,.348,.29,.278,.172,.145,.855/
C      DATA PUMU/3.53,2.52,2.313,1.944,1.435,1.156,1.099,.63,.486,3.85/
C      DATA INDX1/2,3,5,6,7,10,13,15/
C      DATA INDX2/4,4,4,7,8,11,11,11/
C      DATA CUMU/5.515,3.94,3.54,2.44,1.63/
C      DATA PUMLO/21.5,16.1,14.2,10.2,7./
C      DATA ENLO/38.6,43.5,45.2,51.6,59.5/
C-----
C
C      Determine the relative counting efficiencies of the peaks in the 129 to
C      208 keV region. This is done by first calculating the contributions due to
C      Cd absorption, Pu self-attenuation, and the efficiency of the detector.
C      (The relative efficiencies and the Cd and Pu thicknesses are determined in
C      MGAABS.
C
C      DO 10 J = 1,10
C      ENLOG = ALOG(EN(J))
C      ENLSQ = ENLOG**2
C      EN100 = EN(J) - 100.
C      FEMU = EXP(28.27958-10.80034*ENLOG+.959801*ENLSQ)
C      FECOF = EXP(-FEMU*SHAPC(14))
C      PBMU = EXP(13.21214-2.503051*ENLOG)
C      PBCOF = EXP(-PBMU*SHAPC(9))
C      DETEFF = EXP(SHAPC(12) *(ENLOG-4.605) +SHAPC(13) *(ENLSQ -21.207))
C      !Detector efficiency
C      DETEFF = DETEFF *(1.+ESLP*EN100+ECURV*EN100**2)
C      UMU = .956*PUMU(J)
C      !absorption coeff.
C      PUMUX = PUGPSC *(PUMU(J) *PUFRAC +UMU*(1.-PUFRAC))
C      EE(J)=FECOF*EXP(-CDMU(J)*CDABS)*PBCOF*(1.-EXP(-PUMUX))/
C      + PUMUX*DETEFF
C      IF (J .EQ. 7) EFF208 = DETEFF
C      IF (J .GT. 6) EFF(J-6) = EE(J)/DETEFF
C 10 CONTINUE
C      !Ratio of 148/125 keV peak efficiencies
C      E4825 = EE(2)/EE(10)
C      !Ratio of 148/129 keV peak efficiencies
C      E4829 = EE(2)/EE(1)
C
C      !Do following if 237Np was detected.
C      IF (ICMP(8) .GT. 0 .AND. SPC2ID(1:1) .EQ. ' ') THEN
C      BB=-3.9748*ALOG(AREA(11)*1.82*EFF(2)/(AREA(12)*53.4*EFF(1)))
C      AA = -5.3375 *BB
C      EFF300 = EXP (AA + 5.704 * BB)
C      !Eff. of 300 relative to 208.
C      EFF300 = EFF300 * EFF(3) * EFF208
C      SIG = 2.*AREA(14)*SQRT((ERR(14)/AREA(14))**2+(ERR(12)
C      + /AREA(12))**2+(ERR(11)/AREA(11))**2+.0001)
C      YNET(2) = AREA(14)
C      !Insert as second equation
C      Y(2) = AREA(14)/SIG
C      ACOEF(2,ICMP(8)) = .066 * EFF300 / SIG
C      ENDIF
C
C      Calc. approximate efficiencies for the 45,148, and 208 keV peak regions
C      (Remember detector efficiency at 100 keV is defined to be 1.0)
C      These values are used only to provide an approximately correct weighting

```

```

C of the various peak pairs that also included in the analysis
C
C      !Efficiency for 43-51 keV region
C      EFF(1)= AREA(4)/(2.7E-4 *ANS(2))
C      !Approx. same for 45 keV peak
C      EFF(2) = EFF(1)
C      !Approx. same for 59 keV peak
C      EFF(3) = EFF(1)
C      !Efficiency for 129-148 keV peaks
C      EFF(4) = EE(1)
C      !Efficiency for 148-152 keV peaks
C      EFF(5) = EE(2)
C      !Efficiency for 203-208 keV peaks
C      EFF(6) = EE(6)
C      !Efficiency for 185-208 keV peaks
C      EFF(7) = EE(6)
C      !Efficiency for 205-208 keV peaks
C      EFF(8) = EE(6)
C
C Calculate weighting factors for the following 8 peak pairs: 43/51, 45/51,
C 59/51, 129/148, 148/152, 203/208, 185/208, and 300/208 keV.
C
C      !Assume 38-59 keV region will not be analyzed
C      IS = 4
C      !38-59 keV region will be analyzed
C      IF (NOPFLG(1) .EQ. 1) THEN
C        IS = 1
C
C Determine efficiency slope using measured intensities of the 38 and
C 51 keV peaks of 239Pu.
C      !Calc. attenuations due to Pu and Cu abs.
C      DO 30 J = 1,5
C        CUAT(J) = EXP(CUMU(J)*SHAPC(14))
C        PUM = PUMLO(J) * PUGPSC
C        EXPN = 1.
C        IF (PUGPSC .LT. 1.) EXPN = 1.-EXP(-PUM)
C        PUAT(J) = PUM/EXPN
C      CONTINUE
C      !Account for small interference from 237U
C      A51 = AREA(4) * (1.-.00033 * AREA(11)/AREA(10))
C      !Slope of residual efficiency curve
C      DPM51 = A51/2.67E-4
C      ESLOP = .077 *(DPM51 -AREA(1)/.99E-4*PUAT(1)/PUAT(4)*
C      +      CUAT(1)/CUAT(4))
C      !Zero intercept value
C      AZ = DPM51 - ESLOP * 51.6
C      !Calculate efficiencies relative to 51 keV
C      DO 40 J = 1,3
C        L = J+1
C        IF (J .EQ. 3) L = L + 1
C        ERATO(J)=DPM51/(AZ+ESLOP*ENLO(L))*PUAT(L)/PUAT(4)*CUAT(L)/
C      +      CUAT(4)
C      CONTINUE
C      ER38 = (0.5 *ERR(1)/AREA(1))**2
C      ENDIF
C
C DO 41 I = IS,8
C      !Index of peak
C      J = INDX1(I)
C      !Index of reference peak
C      K = INDX2(I)
C      IF (I .EQ. 4) ER38 = 0.0
C      !To prevent divide by zero
C      AR = AREA(J)+1.
C      WT(1) = EFF(1)/(AR*SQRT(ER38 +(ERR(J)/(AR))**2
C      !Weighting factor

```

```

1      + (ERR(K)/ (1. +AREA(K)))**2))
41     CONTINUE
C
C      !Prepare for additional equations
C      ND = NDPTS
C      NPT100 = NDPTS
C      DO 42 I = ND+1,ND+8
C        Y(I) = 0.0
C      !Assume 237U-241Pu in equilibrium
C      NN = 4
C      !237U-241Pu not in equilibrium
C      IF (NOPFLG(1) .NE. 0) NN = ICMP(7)
C      !Sample is aged material
C      IF (NOPFLG(1) .NE. 1) GOTO 45
C
C Include an equation representing the intensity ratio of the 43/51 keV pair
C      ND = ND + 1
C      !238Pu coefficient
C      ACOEF(ND,1) = BRNCH(1) * WT(1)
C      !241Am coefficient
C      ACOEF(ND,5) = 6.48E-4 * WT(1)
C      !239Pu coefficient
C      ACOEF(ND,2) = -BRNCH(3)*WT(1)*ERATO(1)*AREA(2)/AREA(4)
C      !Report area of 43 keV peak only
C      YNET(ND) = AREA(2)
C
C Include an equation representing the ratio of the 45/51 keV pair
C      ND = ND + 1
C      !239Pu coefficient
C      ACOEF(ND,2) = -BRNCH(3)*WT(2)*ERATO(2)*AREA(3)/AREA(4)
C      !240Pu coefficient
C      ACOEF(ND,3) = BRNCH(2) * WT(2)
C      !Report area of 45 keV peak only
C      YNET(ND) = AREA(3)
C
C Include an equation representing the ratio of the 59/51 keV pair
C      ND = ND + 1
C      !239Pu coefficient
C      ACOEF(ND,2) = BRNCH(3) * WT(3)
C      AR = AREA(4)/AREA(5)/ERATO(3)
C      !237U coefficient
C      ACOEF(ND,NN) = -8.49E-6 * WT(3) * AR
C      !241Am coefficient
C      ACOEF(ND,5) = -.359 * WT(3) * AR
C      !Report only the 51 keV area
C      YNET(ND) = AREA(4)
C
C Include the 185 and 205 keV peaks of 235U, if present.
C
45     IF (ICMP(9) .GT. 0) THEN
C      !208/185 keV efficiency ratio
C      EFR = EE(7)/EE(5)
C      !Position index of 235U
C      NC = ICMP(9)
C      !185 keV 235U coefficient
C      ND = ND + 1
C      ACOEF(ND,NC) = .561 * WT(7)
C      TEMP = -BRNCH(7)/AREA(11)
C      !208 keV 237U coefficient
C      ACOEF(ND,NN) = EFR*TEMP *AREA(13)*WT(7)
C      Y(ND) = 0.0
C      YNET(ND) = AREA(13)
C

```

```
C      IF (AREA(15) .GT. 0.0) THEN
C          !208/205 keV efficiency ratio
C          EFR = EE(7)*2./(EE(6)+EE(7))
C          !205 keV 235U coefficient
C          ND = ND + 1
C          ACOEF(ND,NC) = .0487 * WT(8)
C          !208 keV 237U coefficient
C          ACOEF(ND,NN) = EFR*TEMP* AREA(15)*WT(8)
C          Y(ND) = 0.0
C          YNET(ND) = AREA(15)
C
C      END IF
C      END IF
C
C      Include an equation representing the intensity ratio of the 129/148 keV pair
C
C      ND = ND + 1
C          !Efficiency ratio
C      EFR = EE(2) /EE(1)
C          !239Pu coefficient
C      ACOEF (ND,2) = BRNCH(4)* WT(4)
C          !Adjust BRNCH(5) because bkgrd taken only on high side
C      BRNCH(5) = 1.01 * BRNCH(5)
C          !241Pu coefficient
C      ACOEF (ND,4) = -BRNCH(5)*EFR*WT(4)*AREA(6)/AREA(7)
C          !Report area of 129 keV peak only
C      YNET(ND) = AREA(6)
C
C      Include an equation representing the intensity ratio of the 148/152 keV pair
C
C      ND = ND + 1
C          !Efficiency ratio
C      EFR = EE(3) /EE(2)
C          !238Pu coefficient
C      ACOEF(ND,1)=BRNCH(6)*WT(5)*EFR*AREA(7)/(AREA(8)+1.)
C          !241Pu coefficient
C      ACOEF(ND,4) = -BRNCH(5) * WT(5)
C          !Report area of 152 keV peak only
C      YNET(ND) = AREA(8)
C
C      Include an equation representing the intensity ratio of the 203/208 keV pair
C
C      ND = ND + 1
C          !Efficiency ratio
C      EFR = EE(7) /EE(6)
C          !239Pu coefficient
C      ACOEF(ND,2) = .56E-5 * WT(6)
C          !241Pu coefficient
C      ACOEF(ND,NN)=-BRNCH(7)*WT(6)*EFR*AREA(10)/AREA(11)
C          !241Am coefficient
C      ACOEF(ND,5)=-7.91E-6 *WT(6)*EFR*AREA(10)/AREA(11)
C          !Report area of 203 keV peak only
C      YNET(ND) = AREA(10)
C
C      NDPTS = ND
C
C      RETURN
C      END
```

SUBROUTINE MGAHIE

MGAHIE

Version: 9.0 for the PC
 Author: Ray Gunnink
 Date: December-91

Abstract:

This subroutine is called only when a second spectrum has been taken, containing spectral peaks of higher energy. This, and subsequent, subroutines use data from the 332-345, 367-380 and 630-664 Kev regions, and incorporate the corresponding equations representing these regions into the set already obtained from the lower energy spectrum. The primary function of subroutine MGAHIE is to develop an approximate relative efficiency curve for the 300-1000 keV region. Several ancillary analyses are performed during this process, such as, determining peak width parameters and the system zero and gain, removing some small interference peaks, and integrating of certain peak areas.

The following common block files are specific to the solid isotopic instrument.

\$INCLUDE: 'MGACOM.CMN'
 \$INCLUDE: 'MGAEL.CMN'
 \$INCLUDE: 'MGAHI.CMN'

DIMENSION EEN(7),CONST(7),HT(7),Y1(160),Y2(100),Y3(40),
 + AMPK(4),AMHT(4),PKER(7),AVBKG(7),WTS(7),AC(7,3),
 + NFLG(8),NYFLG(6),NXFLG(6),ENRG1(6),ENRG2(4),HIGHT(6),GAMA(6),
 + DELY(100),CONTS(7),PKCNTS(6),ERCTS(6)
 double precision AP(3,3),XSUM(3)
 REAL*8 AC1,AC2,ACSQ

EQUIVALENCE (Y2,Y(200))

DATA EEN/345.014,375.042,414.450,646.769,4.267,54/
 DATA CONST/5.592,15.7,14.8,1.91,.149,.112,.182/
 DATA AMPK/-5.97,-2.45,1.21,2.76/,AMHT/.001,.0255,.0242,.0136/
 DATA NFLG/1,1,0,0,0,0,0,0/
 DATA ENRG1/717.64,721.99,723.79,724.23,727.34,729.52/
 DATA ENRG2/763.3,765.78,766.6,769.4/
 DATA IODEV /4/

!Gain squared

GSQ = GAIN2*GAIN2

!Short term tailing slope constant

EXP2 = SHAPC2(5) * GAIN2

NSTART = 4096

Read in 208 keV peak as an energy reference

NST = NSTART + 200.1/GAIN2 + ZERO2

CALL TRNSFR (60,YDATA(NST),Y1)

!Find channel with max count

NS = 7./GAIN2

NE = 12.5/GAIN2

CALL MAXVAL (NS,NE,J208,PHT,Y1)

!Start of front-side bkgrd

N2 = J208 - 2.1/GAIN2

!Starting position of peak

N1 = N2 - 2

!End position of peak

M1 = J208 + 2.1/GAIN2

!235U is present

IF (NOPFLG(2) .GT. 5.) THEN

!Pick up different bkgrd.

DO 9 I = N1,N2

Y1(I) = Y1(N2+16)

END IF

!Get net counts

CALL NCNTS (N1,N2,M1,60,ND,S1,A208,ER208,AVBG,Y1,Y1)

!Get shape parameters

EPK = 208.

CALL SPRAM (EPK,GAIN2,SHAPC2,SG,ALPHA,F12M,EXP1)

!Approx. peak position

JMAX = J208 - N2

!Preserve width

ALFA = ALPHA

!Fit peak top

CALL GFIT (JMAX,4,ALFA,EXP1,EXP2,PK208,HT208,ERRR,Y1)

ARAT = ALPHA/ALFA

FW208 = F12M * SQRT(ARAT)

!Exact peak position

PK208 = PK208 + FLOAT(N2+NST-1)

Read in 414 keV peak region

NST = NSTART + 405.1/GAIN2 + ZERO2

CALL TRNSFR (60,YDATA(NST),Y1)

!Find approx. peak position

NS = 6.25/GAIN2

CALL MAXVAL (NS,NE,JMAX,PHT,Y1)

N1 = JMAX - 5.6/GAIN2

N2 = N1 + 3

M1 = JMAX + 3.1/GAIN2

M2 = M1 + 3

!Net counts

CALL NCNTS (N1,N2,M1,M2,NPTS,S1,S2,ERRR,AVBKG(3),Y1,Y1)

!Get shape parameters

EPK = 413.7

CALL SPRAM (EPK,GAIN2,SHAPC2,SG414,ALPHA,F12M,EXP1)

!Preserve width

ALFA = ALPHA

JMAX = JMAX - N2

!Fit peak top

CALL GFIT (JMAX,3,ALFA,EXP1,EXP2,PK414,HT(3),ERRR,Y1)

!Position in 4096 channels

PK414 = PK414 + N2 + NST - 1

IF (ABS(PK414-1655.) .GT. 10.0) ERRFLG(21) = 1

!Gain in keV /channel

GAIN2 = 205.7/(PK414 - PK208)

GSQ = GAIN2*GAIN2

!Zero energy channel position

ZERO2 = PK208 - 208. / GAIN2 - NSTART

!Zero energy noise

S1 = (-2.7726/ALFA-.462)*GSQ- SHAPC2(2)*413.7

IF (FW208*GAIN2 .GT. 1.1) ERRFLG(22) = 1

```

C      IF (FW208*GAIN2 .GT. 1.6) THEN
C1302  WRITE (NDEV3,1302)
        FORMAT(' FWHM of 2nd detector unacceptable. Spectrum not used')
        ERRFLG(22) = 2
        ICMP(8) = 0
        SPC2ID(1:1) = ' '
        RETURN
      ENDIF
C      !Store new zero energy resolution
      SHAPC2(1) = S1
C      !Change in peak-width per channel
      ASLP = SHAPC2(2) / GAIN2
C      An estimate of the Pb absorber thickness is now made that will be used to
C      delineate an intrinsic efficiency curve. The relative intensities
C      of the 208 and 267 keV peaks are used for the Pb thickness determination.
C      !Start position of 267 keV peak region
      NS = NSTART + 265.13/GAIN2 + ZERO2
      CALL TRNSFR (24,YDATA(NS),Y3)
      NE = 4.4/GAIN2
C      !Get area.
      CALL NCNTS (1,4,NE,NE+4,ND,S1,A267,ERRR,AVBKG(7),Y3,Y3)
      CALL MAXVAL (4,ND,JMAX,PHT,Y3)
      ALFA = ALPHA
      CALL GFIT (JMAX,4,ALFA,EXP1,EXP2,PKP,HT(7),ERRR,Y3)
C      !Normalized intensity ratio
      RATIO = A267/A208 * 29.3
C      !Estimate correction for Pu, Cd and det. eff. (.66=.67/1.2*1.16)
      RATIO = RATIO*.66*(1.-EXP(-1.2*PUGPSC))/(1.-EXP(-.67*PUGPSC))
C      !Absorber thickness > 0
      IF (RATIO .LT. 1.0) RATIO = 1.0
C      !2.6 = 1./(Pbmu(208)-Pbmu(267))
      PBMA5 = 2.6*ALOG(RATIO)
C      Read in 375 keV peak region
C      !Starting channel
      NS = NSTART + 357.76/GAIN2 + ZERO2
      CALL TRNSFR (120,YDATA(NS),Y1)
      NS = 4.76/GAIN2
      NE = 28.6/GAIN2
C      !Find minimum front-side bkgrd
      SUMIN = 1.E10
      SUM = 0.0
      DO 10 I = 1,8
      SUM = SUM + Y1(I)
      DO 12 I = 1,NS
      IF (SUM .LT. SUMIN) THEN
        JMIN = I
        SUMIN = SUM
      ENDIF
      SUM = SUM + Y1(I+10) - Y1(I)
      CONTINUE
      DO 13 I=1,8
      L = I-1
      Y1(NS+L) = Y1(JMIN+L)
      CONTINUE
C      !Net counts
      CALL NCNTS (NS,NS+11,NE,NE+3,N370,S1,S2,ERRR,AVBG3,Y1,Y1)
      AVBKG(2) = AVBG3
C      !Find approx peak position
      MS = 8.25/GAIN2
      CALL MAXVAL (MS,MS+17,JMAX,PHT,Y1)
C      !Peak shape parameters
      EPK = 375.
      CALL SPRAM (EPK,GAIN2,SHAPC2,SG,AL375,F12M,EXP1)

```

```

C      !Preserve tailing amplitude
      EX370 = EXP1
C      !Preserve width parameter
      ALFA = AL375
C      !Fit peak top
      CALL GFIT (JMAX,4,ALFA,EXP1,EXP2,PK375,HT(2),ERRR,Y1)
C      !Preserve net counts for
      DO 14 I = 1,N370
C      !later analysis
      Y2(I) = Y1(I)
C14
C      Strip out 376 keV 241Am peak so that the 375 keV peak can be used as an
C      accurate energy reference. An estimate of the 241Am / 239Pu abundance
C      ratio is obtained from the preliminary analysis of the 100 keV region.
C      !Height of 376 keV peak
      HT376 = HT(2) * 1.38 *ANS(5) /(15.75 *ANS(2))
C      !Position of peak
      PK376 = PK375 + 1.5525/GAIN2
C      !Strip out peak
      WL = 0.0
      CALL STRIP1 (38,N370,PK376,HT376,AL375,EXP1,EXP2,WL,Y1)
C      !Fit peak top
      CALL GFIT (JMAX,3,AL375,EXP1,EXP2,PK375,HT(2),ERRR,Y1)
C      Read in 450 keV peak region.
C      NS = NSTART + 448./GAIN2 + ZERO2
      CALL TRNSFR (80,YDATA(NS),Y1)
      NE = 14.25/GAIN2
C      !Net counts
      CALL NCNTS (1,7,NE,NE+2,NPTS,S1,S2,ERRR,AVBKG(4),Y1,Y1)
C      !Approx. peak position
      CALL MAXVAL (5,15,JMAX,PHT,Y1)
C      !Preserve width
      ALFA = ALPHA
C      !Fit peak top
      CALL GFIT (JMAX,4,ALPHA,EXP1,EXP2,PKP,HT(4),ERRR,Y1)
C      !Small 241Am interference
      HT(4) = HT(4) - .05*Y1(JMAX+12)
C      Examine the 497 and 622 keV regions for presence of 103Ru & 106Ru.
C      NS = NSTART + 495./GAIN2 + ZERO2
      CALL NCNTS (NS-4,NS,NS+14,NS+20,NPTS,S1,CTSFIS(3),ERFIS(3),
      +      AVBG,YDATA,Y1)
      NS = NSTART + 613.5/GAIN2 + ZERO2
      CALL TRNSFR (100,YDATA(NS),Y1)
      CALL MAXVAL (10,25,JMAX,PHT,Y1)
      ALFA = ALPHA
      CALL GFIT (JMAX,3,ALFA,EXP1,EXP2,PKP,PHT,ERRR,Y1)
      PHT = PHT - Y1(1)
      WL = 0.0
      CALL STRIP1 (JMAX-12,JMAX+10,PKP,PHT,ALPHA,EXP1,EXP2,WL,Y1)
      NS = 8./GAIN2
C      Transfer lower BG up
      CALL TRNSFR (9,Y1,Y1(NS-9))
      NE = NS + 7./GAIN2
      CALL TRNSFR (12,Y1(NE),Y1(NS+12))
      CALL NCNTS (NS-9,NS,NS+12,NS+23,NPTS,S1,CTSFIS(4),ERFIS(4),
      +      AVBG,Y1,Y1)
C      Read in 646 keV peak to help determine approx. efficiency curve
C      !Starting channel
      NS = NSTART + 628.51/GAIN2 + ZERO2

```



```

      CALL TRNSFR (155,YDATA(NS),Y1)
      NE = 37.6/GAIN2
C      !Net counts
      CALL NCNTS (1,5,NE,NE+5,NPTS,S1,S2,ERRR,AVBKG(5),Y1,Y1)
C      !Find approx. peak position
      MS = 13.75/GAIN2
      CALL MAXVAL (MS,MS+15,JMAX,PHT,Y1)
C      !Get shape parameters
      EPK = 646.
      CALL SPRAM (EPK,GAIN2,SHAPC2,SG,ALPHA,F12M,EXP1)
C      !Preserve width
      ALFA = ALPHA
C      !Fit peak top
      CALL GFIT (JMAX,3,ALFA,EXP1,EXP2,PK646,HT(5),ERRR,Y1)
      SG = -2.7726/ALFA
      A646 = HT(5) * SQRT(1.133*SG)
      AL646 = ALFA
C      !Measured position of 646 keV peak.
      P646 = PK646 + NS + 4
C      Read in 722 keV peak of 241Am. This will be used to strip out 241Am
C      peaks in the 769 keV region.
C      !Starting channel
      NS = NSTART + 711.51/GAIN2 + ZERO2
C      !Read in data
      CALL TRNSFR (110,YDATA(NS),Y1)
      MS = 4.55/GAIN2
      NE = 19.05/GAIN2
      CALL TRNSFR (5,Y1(NE+MS),Y1(NE))
C      !Net counts
      CALL NCNTS (1,MS,NE,NE+2,NPTS,S1,S2,ERRR,AVBG,Y1,Y1)
C      !Channel of maximum count
      IS = 5.26/GAIN2
      CALL MAXVAL (IS,IS+9,JMAX,PHT,Y1)
      ALFA = ALPHA
      PKP = JMAX + 8
      CALL STRIP1(JMAX-2,JMAX+5,PKP,.02*PHT,ALFA,EXP1,EXP2,WL,Y1)
C      !Fit peak top
      CALL GFIT (JMAX,3,ALFA,EXP1,EXP2,PK722,HT722,ERRR,Y1)
C      !Measured position of 722 keV peak
      P722 = PK722
      PK722 = PK722 + NS + MS-1
C      !Weighting factor
      RAT = HT722 / (HT(5)+HT722)
C      !New gain
      GAN = (1.-RAT)*GAIN2 + RAT*308.34/(PK722-PK414)
C      !Calculated position of 646 keV peak
      CAL646 = 645.969/GAN
C      !Calculated position of 722 keV peak
      CAL722 = 722.04/GAN
C      !Weighted average position
      PK646=P646+RAT*(PK722-CAL722-P646+CAL646)
C      !Calc. weighted peak width at 646 keV
      AL646 = AL646*(1.-RAT)+RAT*1.04*ALFA*(SHAPC2(1)+722.*SHAPC2(2))/
+      (SHAPC2(1)+646.*SHAPC2(2))
C      !Adjusted for efficiency
      HT722 = (0.95+.007*(PBMAS+PUGPSC))*HT722
C      Determine 724 keV peak intensity of 952r
      DO 50 I = 1,6
      NXFLG (I) = -2
      NYFLG(I) = 1
      HIGHT(I) = 0
      GAMA(I) = 0
50  C      NXFLG(2) = 1

```

```

      NYFLG(3) = -2
      HIGHT(3) = -.014
      NYFLG(6) = -2
      HIGHT(6) = -.0055
      NFLG(6) = NOPFLG(5)-1
      IF (NOOUT .EQ. 0) NPLG(6) = -1
      NPLG(7) = NDEV3
      REFEN = 721.99 - P722*GAIN2
      IF (NOPFLG(8) .EQ. 0) THEN
+      CALL PKFIT (6,NPTS,NFLG,NYFLG,NXFLG,REFEN,GAIN2,AVBG,
+      ENRG1,HIGHT,GAMA,Y1,DELY,RM,SHAPC2,PKCNTS,ERCTS)
      CTSFIS(1) = PKCNTS(4)
      ERFIS(1) = ERCTS(4)
      ENDIF
C      Read in 769keV peak to help determine approx. efficiency curve
C      !Starting channel
      NS = NSTART + 761.35/GAIN2 + ZERO2
      CALL TRNSFR (60,YDATA(NS),Y1)
      NE = 13.25/GAIN2
C      !Net counts
      CALL NCNTS (1,5,NE,NE+5,NPTS,S1,S2,ERRR,AVBKG(6),Y1,Y1)
C      !Find approx. peak position
      MS = 6./GAIN2
      CALL MAXVAL (MS,MS+6,JMAX,PHT,Y1)
C      !Get shape parameters
      EPK = 769.4
      CALL SPRAM (EPK,GAIN2,SHAPC2,SG,ALPHA,F12M,EXP1)
C      !Preserve width
      ALFA = ALPHA
C      !Fit peak top
      CALL GFIT (JMAX,3,ALFA,EXP1,EXP2,PK769,HT769,ERRR,Y1)
C      !Better position
      IF (HT722 .GT. 5.*HT769) PK769=PK722+47.38/GAIN2-NS-4
C      Strip out 766, 770, & 772 keV peaks
      DO 68 J = 1,4
C      !Position of 241Am peak
      POS = PK769 + AMPK(J)/GAIN2
C      !Height of 241Am peak
      PHT = HT722 * AMHT(J)
      NST = POS-10
      IF (NST .LT. 1) NST = 1
C      !Strip out pk.
      CALL STRIP1 (NST,NST+20,POS,PHT,ALPHA,EXP1,EXP2,WL,Y1)
68  C      CONTINUE
C      !Refined the peak position
      CALL MAXVAL (MS,MS+8,J769,HT769,Y1)
C      IF (NOPFLG(8) .EQ. 0) THEN
C      Get 766 keV peak intensity of 95Nb.
      NPLG(2) = 0
      MS = 2.9/GAIN2
      DO 69 I = 1,4
      IF (Y1(J769-MS) .LT. HT769) THEN
C      Make 769 keV the reference peak
      NXFLG(I) = -4
      ELSE
C      Make 766 keV the reference peak
      NXFLG(I) = -3
      ENDIF
      NYFLG(I) = 1
      HIGHT(I) = 0.0
69  C      CONTINUE
      IF (NXFLG(1) .EQ. -4) THEN

```



```

      NXFLG(4) = 1
    ELSE
      NXFLG(3) = 1
    ENDIF
  C
  REFEN = 769.4 - PK769*GAIN2
  CALL PKFIT (4,NPTS,NFLG,NYFLG,REFEN,GAIN2,AVBG,
+   ENRG2,HIGHT,GAMA,Y1,RM,DELY,SHAPC2,PKCNTS,ERCTS)
  CTSFIS(2) = PKCNTS(2)
  ERFIS(2) = ERCTS(2)
  ENDIF
  C
  HT766 = 0.0
  C
  !Initialize
  C
  !Large 238U peak present
  IF (Y1(J769-MS) .GT. HT769 .AND. ANS(5) .LT. .5*ANS(2))THEN
  C
  !Find 766 keV peak.
  CALL MAXVAL (10,19,J766,HT766,Y1)
  CALL STRIP1 (J766-5,J766+5,PK769,HT769,ALPHA,EXP1,EXP2,WL,Y1)
  C
  !Position of 766 keV peak
  PK766 = PK769-2.96/GAIN2
  CALL STRIP1 (J769-5,J769+5,PK766,Y1(J766),ALPHA,EXP1,EXP2,WL,Y1)
  ALFA = ALPHA
  C
  !Fit peak top
  CALL GFIT (J766,3,ALFA,EXP1,EXP2,PK766,HT766,ERRR,Y1)
  ENDIF
  C
  ALFA = ALPHA
  C
  !769 pk height
  CALL GFIT (J769,3,ALFA,EXP1,EXP2,PK769,HT(6),ERRR,Y1)
  C
  !Peak height ratio
  R6669 = HT766/HT(6)
  C
  !For weighting
  IF (HT766 .GT. 0.) S6669=2.5*HT766*SQRT((1./SQRT(HT(6))))**2+
+   (1./SQRT(HT766))**2)
  C
  Read in 332-335 kev peak region
  C
  !Starting channel
  NS = NSTART + 326.1/GAIN2 + ZERO2
  CALL TRNSFR (100,YDATA(NS),Y1)
  C
  !Find approx. position of 345 kev peak
  MS = 17./GAIN2
  CALL MAXVAL (MS,MS+17,JMAX,PHT,Y1)
  C
  Obtain net counts of peaks in this region.
  C
  !Start of peak grouping
  N1 = JMAX - 15.4/GAIN2
  C
  !End of peak grouping
  M1 = JMAX + 2.6/GAIN2
  C
  !End of aft-side bkgrd
  M2 = M1 + 7
  C
  !Approx. position of 345 kev peak
  MAX = JMAX - N1
  C
  !Get net counts
  CALL NCNTS(1,N1,M1,M2,NPTS,S1,S2,ERRR,AVBKG(1),Y1,YNET(NDPTS+1))
  C
  !Get shape parameters
  EPK = 333.
  CALL SPRAM (EPK,GAIN2,SHAPC2,SG,AL333,F12M,EXP1)
  C
  !Preserve width
  ALFA = ALPHA
  C
  !Fit peak top
  CALL GFIT (MAX,4,ALFA,EXP1,EXP2,PK345,HT(1),ERRR,YNET(NDPTS+1))
  C
  !Starting position for integration
  A333 = 0.0
  DO 80 I = 1,MAX-24
  C
  !Accumulate area of 332-335 kev peaks
  A333 = A333 + YNET(NDPTS+1)

```

```

  C
  !Preserve array for next subroutine
  C80 YNET(I+NDPTS) = Y1(I)
  C
  C
  C Integrate the 345 kev peak due to 239Pu
  C
  A345 = 0.0
  DO 82 I = MAX-9,NPTS
  C
  !Accumulate sum into A345.
  A345 = A345 + YNET(NDPTS+1)
  C
  !Average bkgrd
  AVBG2 = AVBKG(1)
  C
  !Statistical error of A345.
  ER345 = SQRT(A345+43.*AVBG2)+.002*A345
  C
  C The intense 312 keV peak of 233Pa is used to measure the abundance of
  C the parent 237Np in the sample (if they are in equilibrium). This peak
  C can also be used to correct for a small 233Pa peak contribution to the
  C 375 keV peak. However, 239Pu also has a small peak at 312 keV. Its
  C intensity is estimated from the 345 keV peak intensity and the remainder
  C assign to 233Pa.
  C
  NST = NSTART + ZERO2 + 309.1/GAIN2
  CALL TRNSFR (24,YDATA(NST),Y3)
  NE = 5.25/GAIN2
  C
  !Get net area of pk.
  CALL NCNTS(1,4,NE,NE+3,ND,S1,A312,ER312,AVBG,Y3,Y3)
  C
  !345/414 efficiency ratio
  EFF345 = HT(1)/HT(3)*CONST(3)/CONST(1)
  EFF312 = EXP(1.5532 * ALOG(EFF345))
  EFF375 = EXP(.54264 * ALOG(EFF345))
  HT375=(.0172*EFF375/EFF312*(A312-.046*EFF312/EFF345*A345))
+   /SQRT(1.133*SG414)
  C
  !Subtract out 237Np contribution to 375 keV pk.
  HT(2) = HT(2) - HT375
  C
  !Adjust peak position for interference
  PK375 = PK375-1.5*HT375/HT(2)
  C
  !Adjust peak width for interference
  AL375 = AL375*(1+.4*HT375/HT(2))
  C
  C The relative counting efficiencies of peaks within a given peak cluster
  C must be known. To obtain the intra-group efficiency slope, an approximate
  C relative efficiency curve is delineated by fitting the intensities (peak
  C areas) of 6 peaks belonging to 239Pu versus their energies. (Fit function is
  C quadratic curve on a log-log scale.) The peaks used are: 345, 375, 414,
  C 450, 646, 769 and 267 keV.
  C
  NC = 4
  IF (ICMP(7) .GT. 0) NC = ICMP(7)
  FCT = ANS(2)/(ANS(NC)*(1+.445*ANS(5)/ANS(NC)))
  C
  PUMAS = PUGPSC
  NUM = 2
  C
  !Pu thickness O.K.
  IF (PUMAS .LT. 2.) NUM = 7
  85 I = 1
  C
  DO 86 L = 1,NUM
  ENLOG = LOG(EEN(1))
  ENSQ = ENLOG**2
  C
  !Cd abs. coef.
  CDMU = EXP(13.26072-4.240781*ENLOG+.2761391*ENSQ)
  C
  !Cd absorption
  CDMAS = EXP(1.36*CDMU)
  C
  !Pb abs. coef.
  PBMU = EXP(27.72774-8.025776*ENLOG+.5248603*ENSQ)
  C
  !Pb absorption

```

```

      PBABS = EXP(PBMU *PBMAS)
      C      !Pu abs. coef.
      PUMU = EXP(27.85913-7.840364*ENLOG+.4977653*ENSQ)
      PUMUX = PUMU*(PUMAS+.1)
      C      !Pu attenuation
      PUABS = PUMUX/(1.-EXP(-PUMUX))
      C      !Shape parameter
      SG = (SHAPC2(1) +EEN(1) *SHAPC2(2))/0.0625+.462
      C      !Normalized intensity
      CONTS(1) = HT(1) * SQRT(1.133 *SG)
      C      !Frac. error
      PKER(1) = SQRT((CONTS(1)+20.*AVBKG(1))/CONTS(1)**2+.0001)
      C      !Corrected counts
      IF (1 .EQ. 7) CONTS(1) = FCT * CONTS(1)
      CONTS(1) = CONTS(1)*CDMAS*PBABS*PUABS/CONST(1)
      C      !Weighting factor
      WTS(1) = 1./PKER(1)**2
      AC(1,1) = 1.
      AC(1,2) = ENLOG
      AC(1,3) = ENSQ
      I = L+1
      IF (NUM .EQ. 2) I = 6
86  CONTINUE
      C
      IF (NUM .EQ. 7) GOTO 87
      IF ((CONTS(1)/CONTS(6) .GT. 1.5) .OR. (PUMAS .GT. 40.)) THEN
        NUM = 7
      ELSE
        C      !Increment the Pu thickness
        PUMAS = PUMAS + 2.
      END IF
      C      !Repeat process
      GOTO 85
      C
87  LCMP = 3
      DO 90 J = 1,LCMP
        XSUM(J) = 0.0
        DO 90 K = 1,LCMP
          AP(J,K) = 0.0
        C
        DO 92 J = 1,7
          C      !Log of intensity
          CONTS(J) = LOG(CONTS(J))
          DO 91 K = 1,LCMP
            XSUM(K) = XSUM(K) + AC(J,K) * CONTS(J)*WTS(J)
            DO 91 L = 1,LCMP
              AC1 = AC(J,K)
              AC2 = AC(J,L)
              ACSQ = AC1 * AC2
              AP(K,L) = AP(K,L) + ACSQ*WTS(J)
            C
            CONTINUE
          C
          CALL MATINV (AP,LCMP,LCMP,DET)
          C
          DO 94 K = 1,LCMP
            ANSR(K) = 0.0
            DO 93 L = 1,LCMP
              ANSR(K) = ANSR(K) + XSUM(L) * AP(K,L)
            C
            CONTINUE
          C
          write (*,*) ansr(1),ansr(2),ansr(3)
          C
          do 990 j = 1,7
            enlog = log(een(j))
            C
            ensq = enlog**2
            C
            calc = exp(ansr(1) +ansr(2)*enlog +ansr(3)*ensq)
            C
            cc = exp (conts(j))
            C990 write (*,992) een(j),pker(j),WTS(J),calc,cc,100.*(calc-cc)/cc

```

```

C992  format (f6.1,f7.5,f8.2,2f8.0,2f8.3)
C      EF1001 = CALERR (6.909,AP,LCMP,CHISQR)
C
C500  RETURN
      END

```

```
C
C The following block-data subprogram initializes the MGACOM
C common block.
C
C      BLOCK DATA MGACMN
C
C $INCLUDE: 'MGACOM.CMN'
C
C      DATA SPC2ID(1:1)/' ',IOCH1/2/,IOCH2/3/
C      DATA SFIL /'SHAPC.
C      DATA SFIL2 /'SHAPC2.
C      DATA SHAPC/.06,.001751,-4.,.005,3.,0.,.004,.4,.0,.0,
+      3.,.0,.0,.0,.0/
C      DATA SHAPC2/.4,.00195,-4.7,.004,1.5,0.,.002,.15,0.,0./
C      DATA ICMP/1,2,3,4,5,6,0,0,0,0,0,1,1,1,0,0/
C      DATA SPC1ID/'
C      DATA ERRFLG/0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/
C      DATA CDATE/' Measurement'
C      DATA CMONTH/'Jan','Feb','Mar','Apr','May','Jun','Jul','Aug',
+      'Sep','Oct','Nov','Dec'/
C
C      !Spectrum format index LLNL=1, LANL(Canberra)=2, ASCII=3, IAEA=4
C      !6="MGA\PUIS\PURAW.BIN" file exists in IAEA format.
C      !7=New Canberra S100 format.
C      !Set NOPFLG(3) to specify data format type.
C      DATA NOPFLG/0,0,4,0,0,0,0,0,0/
C      DATA NOOUT /1/
C
C      Device assignments: 5=read, 6=screen, 7=printer, 9=OUTMGA.TXT
C      DATA NDEV1/5/,NDEV2/6/,NDEV3/7/,NDEV9/9/
C
C      END
```

```

=====
SUBROUTINE MGA300
=====
                                MGA300
=====

Version: 9.0 for PC
Author: Ray Gunnink
Date: December-91

*****

Abstract:

This routine is a sequel to subroutine MGAHIE and is used only when a
second spectrum available, containing spectral peaks of higher
energy. This subroutine develops equations representing data
found in the 332-345 and 367-380 keV regions and adds these to
the set obtained from the lower energy spectrum.

=====

The following common block files are specific to the solid isotopic
instrument.

$INCLUDE: 'MGACOM.CMN'
$INCLUDE: 'MGAHI.CMN'

C
+ DIMENSION NFLG(8),ENRG(20),BR(20,3),NYFLG(20),NXFLG(20),
+ HIGHT(20),GAMA(20),NXXFLG(20),NYYFLG(20),ENR(20),BRANCH(3,20)
+ DIMENSION EF(20),Y2(80),DELY(100),RM(100)
+ EQUIVALENCE (Y2,Y(200))

C
+ DATA NXFLG/1,-8,-1,-8,-1,-8,-8,1,-6,-6,-6,-6,-6,1,-6,-6,-6,-6,-6,
+ -6,0/

C
+ DATA NYFLG/1,-8,1,-8,1,1,-8,1,-6,-6,1,1,-6,1,1,-6,-6,-6,-7,0/

C
+ DATA ENRG/ 332.390, 332.853, 335.430, 336.116, 337.760,
+ 340.470, 341.526, 345.014, 367.048, 368.543,
+ 368.610, 370.933, 372.470, 375.039, 376.600,
+ 379.050, 380.160, 382.732, 383.744, 311.900/

C
+ DATA BRANCH/
+ 239Pu 241Pu 241Am
+ .0, 2.977E-7, 1.484E-6,
+ 5.044E-6, .0, .0,
+ .0, 2.494E-8, 4.940E-6,
+ 1.135E-6, .0, .0,
+ 0., 2.120E-9, .0,
+ 0., .0, .0,
+ 6.62 E-7, .0, .0,
+ 5.590E-6, .0, .0,
+ 8.67 E-7, .0, .0,
+ 9.00 E-7, .0, .0,
+ 0., 1.077E-8, 2.168E-6,
+ 0., 2.740E-8, 5.250E-7,
+ 2.1 E-8, .0, .0,
+ 1.57 E-5, .0, .0,
+ 0., .0, 1.405E-6,
+ 1.05 E-8, .0, .0,
+ 3.051E-6, .0, .0,
+ 2.587E-6, .0, .0,
+ .0, .0, 2.82 E-7,
+ .38, .0, .0,
+ DATA NFLG/ 1,1,0,1,0,0,0,0/

```

```

C
C
10      DO 10 J = 1,20
        DO 10 K = 1,3
        BR(J,K) = BRANCH(K,J)
        DO 13 J = 1,20
        HIGHT(J) = 0.0
        GAMA(J) = 0.0
13
C
C          !Read in file of energy and branching intensity constants
C          !For 332-345 keV group and 367-376 keV group
C          !Default values will be used when file is missing.
C
C      OPEN (UNIT=IODEV,NAME='GRP340.DAT',TYPE='OLD',
C      +      ACCESS='SEQUENTIAL',ERR=9)
C      +      READ (IODEV,9020) (ENRG(I),BR(I,1),BR(I,2),BR(I,3),
C      +      NXFLG(I),NYFLG(I), I = 1,20)
C9020      FORMAT (F10.3,3E10.3,2I5)
C      READ (IODEV,9022) (NFLG(I),I=1,7)
C9022      FORMAT (7I5)
C9      CLOSE (IODEV)
C
C      DO 105 I = 1,20
        ENLOG = ALOG(ENRG(I))
        ENSQ = ENLOG**2
C
C          !Cd abs. coef.
        CDMU = EXP(13.26072-4.240781*ENLOG+.2761391*ENSQ)
C          !Cd attenuation
        CDMAS = EXP(-1.36*CDMU)
C
C          !Pb abs. coef.
        PBMU = EXP(27.72774-8.025776*ENLOG+.5248603*ENSQ)
C          !Pb attenuation
        PBABS = EXP(-PBMU*PBMAS)
C
C          !Pu abs. coef.
        PUMU = EXP(27.85913-7.840364*ENLOG+.4977653*ENSQ)
C          !Pu attenuation
        PUABS = (1.-EXP(-PUMUX))/PUMUX
C          !Calc. relative eff.
        EF(I) = CDMAS*PBABS*PUABS*EXP(ANSR(1)+ANSR(2)*ENLOG+
105      +      ANSR(3)*ENSQ)
C      CONTINUE
C
C      Correct the branching intensities for change in efficiency using the
C      efficiency of the 1st peak (332 keV) as a reference.
C
C      DO 109 I = 1,20
        RATIO = EF(1)/EF(1)
        DO 109 K = 1,3
        BR(I,K) = BR(I,K) *RATIO
109
C      DO 110 I =1,12
        NXXFLG(I) = NXFLG(I+8)
        NYFLG(I) = NYFLG(I+8)
        ENR(I) = ENRG(I+8)
110      CONTINUE
        REFEN = 345.011 - GAIN2*PK345
        HIGHT(2) = -BR(2,1)/BR(8,1)
        HIGHT(4) = -BR(4,1)/BR(8,1)
        HIGHT(7) = -BR(7,1)/BR(8,1)
C
C      NFLG(6) = NOPFLG(5)-1
        IF (NOOUT .EQ. 0) NFLG(6) = -1
        NFLG(7) = NDEV3
        NFLG(8) = 0

```

```

C
C   IF (NOPFLG(8) .EQ. 0)
C   +   CALL PKFIT(8,NPTS,NFLG,NYFLG,NXFLG,REFEN,GAIN2,AVBG2,
C   +   ENRG,HIGHT,GAMA,YNET(NDPTS+1),DELY,RM,SHAPC2,CTS330,ER330)
C
C 312 keV peak intensity must be corrected for a small contribution from
C a peak due to 239Pu before it can be used to estimate the contribution
C to the 375 keV peak intensity.
C
C   A345 = CTS330(8)
C   ER312 = ER312 + .01*A312      !Uncertainty > 1%.
C   ERATIO = EF(20)/EF(8)        !Ratio of 312/345 keV efficiencies
C   A312 = A312 - 0.0461 *ERATIO *A345      !Subtract out 239Pu peak
C   IF (A312 .LE. 0.0) A312 = 1.0      !Do not let value be neg. or zero
C   IF ((ER312/A312 .LT. .2) .AND. (ICMP(8) .LE. 0)) THEN
C   !Add 237Np to component list
C   NCMP = NCMP + 1
C   ICMP(8) = NCMP      !Record index of component position
C
C   END IF
C
C An approximate ratio of counting efficiencies between the 100 keV region
C of the 1st spectrum and the 332 keV position of the 2nd spectrum must
C be determined. This ratio is used only in calculating weighting factors
C to the equations generated in the 300 keV regions so that they can be
C added to the 100 keV region equations for the final least-squares solution.
C The procedure used here is to compare the 241Pu answer calculated from
C the 332 keV peak with that obtained from the 100 keV region.
C
C   EFF2 = CTS330(1) / (ANS(4) *BR(1,2)+ ANS(5) *BR(1,3))
C   !Efficiency relative to 101 keV
C
C If 237Np is present, then insert the 312 keV peak of 233Pa (its daughter)
C as another equation.
C
C   IF (ICMP(8) .GT. 0) THEN
C   !237Np is present
C   WTF = EFF2/(A345*SQR((ER312/A312)**2+(ER345/A345)**2))
C   NDPTS = NDPTS + 1
C   Y(NDPTS) = 0.0
C   YNET(NDPTS) = A312      !Insert 312 peak as equation #2
C   ACOEF(NDPTS,2) = -WTF * BR(8,1)      !239Pu coef.
C   ACOEF(NDPTS,(ICMP(8))) = WTF * A345/A312 * BR(20,1)      !237Np coef.
C   END IF
C   !Assume 237U-241Pu equilibrium
C   NC = 4
C   !Reassign column index for 237U if not.
C   IF (ICMP(7) .GT. 0) NC = ICMP(7)
C
C The 332 and 335 keV peaks due to 241Pu and 241Am are ratioed to the
C 345 keV intensity (A345) due to 239Pu, thereby producing two new equations.
C
C   L = NDPTS      !241Am inhomogeneity flag
C   IC = ICMP(15)
C   !Set K1 index for 332.39 keV area
C   K1 = 1
C   !Total error assigned to the 345 keV peak.

```

```

C   ER345 = SQR(ER330(8)**2+.7*AVBG2+.002*CTS330(8))
C
C   DO 140 J = 1,2
C   L = L + 1
C   !Ratio 345 area to 332/336 keV areas
C   RATIO = CTS330(8) / CTS330(K1)
C   !Weight factor
C   WTF=EFF2/(CTS330(8)*SQR((ER330(K1)/CTS330(K1))**2+(ER345/
C   +   CTS330(8))**2))
C   !Store weighting factor
C   Y(L) = 0.0
C   YNET(L) = CTS330(K1)
C   !239Pu coefficient
C   ACOEF(L,2) = -WTF*BR(8,1)
C   !237U coefficient
C   ACOEF(L,NC) = WTF * RATIO * BR(K1,2)
C   !241Am coefficient
C   ACOEF(L,5) = WTF * RATIO * BR(K1,3)
C   !Am inhomogeneity
C   IF (IC .GT. 0) ACOEF(L,IC) = ACOEF(L,5)
C   !Account for 345 keV 235U peak
C   IF (NOPFLG(2) .GT. 10) ACOEF(L,ICMP(9)) = -.00034 * WTF
C   !Set indices for 336.104 keV peak
C   K1 = 3
C   140 CONTINUE
C
C The 368-376 keV region is now analyzed and made ready for inclusion.
C
C   HIGHT(1) = -BR(9,1)/BR(14,1)
C   HIGHT(2) = -BR(10,1)/BR(14,1)
C   HIGHT(4) = 0.0
C   HIGHT(5) = -BR(13,1)/BR(14,1)
C   HIGHT(7) = 0.0
C   HIGHT(8) = -BR(16,1)/BR(14,1)
C   HIGHT(9) = -BR(17,1)/BR(14,1)
C   HIGHT(10) = -BR(18,1)/BR(14,1)
C   HIGHT(11) = -BR(19,3)/BR(15,3)
C   REFEN = 375.039 - GAIN2*PK375
C   NFLG(8) = 2
C
C Add 375 keV peak due to Pa233. Fix in position and hight.
C
C   NPKS = 11
C   IF (A312 .GT. 0.0) THEN
C   NPKS = 12
C   HIGHT(12) = .0172*A312*EF(14)/EF(20)
C   ENR(12) = 375.4
C   NYFLG(12) = 0
C   NXXFLG(12) = 0
C   ENDIF
C
C   IF (NOPFLG(8) .EQ. 0)
C   +   CALL PKFIT (NPKS,N370,NFLG,NYFLG,NXXFLG,REFEN,GAIN2,AVBG3,
C   +   ENR,HIGHT,GAMA,Y2,DELY,RM,SHAPC2,CTS370,ER370)
C
C   !Set index for 368 keV peak
C   K1 = 3
C   !Total error assigned to the 375 keV peak.
C   ER375 = SQR(ER370(6)**2+.002*CTS370(6))
C
C   NJ = 2
C   IF (CTS370(7)/CTS370(6) .GT. .15) NJ = 3
C   DO 150 J = 1,NJ
C   L = L + 1
C   !Ratio 375 area to 368/370 keV areas
C   IF (CTS370(K1) .LE. 0.) CTS370(K1) = 1.0

```

```
C      RATIO = CTS370(6) / CTS370(K1)
C      !Weight factor
      WTF=EFF2/(CTS370(6)*SQRT((ER370(K1)/CTS370(K1))**2+(ER375/
+      CTS370(6))**2))
      Y(L) = 0.0
      YNET(L) = CTS370(K1)
C      !239Pu coefficient
      ACOEF(L,2) = -WTF*BR(14,1)
C      !237U coefficient
      ACOEF(L,NC) = WTF * RATIO * BR(K1+8,2)
C      !241Am coefficient
      ACOEF(L,5) = WTF * RATIO * BR(K1+8,3)
C      !Am inhomogeneity
      IF (IC .GT. 0) ACOEF(L,IC) = ACOEF(L,5)
C      !Set index for 370 or 376 keV peak
      K1 = 4
      IF (J .EQ. 2) K1 = 7
150    CONTINUE
C      !Final number of equations
      NDPTS = L
C
500    RETURN
      END
```

```
C=====C
C                                     MGA600
C
C-----C
C
C      Version: 9.0 for the PC
C      Author: Ray Gunnink
C      Date: December-91
C
C      * * * * *
C
C      This subroutine establishes equations representing data in
C      the 630-664 keV region and adds these equations to the set
C      obtained from the lower-energy spectrum. It also analyses the
C      1000 keV region when 238U is found to be present.
C-----C
C
C$INCLUDE: 'MGACOM.CMN'
C$INCLUDE: 'MGAEL.CMN'
C$INCLUDE: 'MGAHI.CMN'
C
C      DIMENSION NPLG(8),NYPLG(20),NXPLG(20),IFLG(8),IXPLG(3),
C      + IYPLG(3),ENRG(3),HIGHT(20),GAMA(20),EN(23),BR(17,3),EF(23),
C      + Y1(200),CTS(5),ERRR(5),DELY(180),BRANCH(3,17),BRFIS(5)
C
C      DATA NXPLG/-5,-5,-5,-5,1,-5,-5,-5,-5,-5,-5,1,1,-5,-13,0,0,0,0,0/
C      DATA IXPLG/-2,1,-2/
C
C      DATA NYPLG/-5,-5,-13,1,1,-5,-5,-5,-5,-13,-5,-5,1,-5,1,0,0,0,0,0/
C      DATA IYPLG/1,1,1/
C      DATA NPLG/ 1,1,0,1,0,0,0,2/
C      DATA IFLG/ 1,0,0,0,0,0,0,0/
C
C      DATA BRFIS/.4,1,.85,.1,.85/
C      DATA ENRG/1000.8,1003.24,1006./
C
C      DATA EN/ 637.862, 640.096, 641.436, 642.500, 645.969, 647.296,
C      +         649.377, 650.529, 652.052, 652.990, 654.890, 658.919,
C      +         662.456, 664.587, 661.615, 766.000,1001.200, 724.230,
C      +         765.786, 497.080, 622.100, 661.646, 375.039/
C
C      DATA BRANCH/
C      239Pu    240Pu    241Am
C      2.56 E-8, .0, .0,
C      +       8.20 E-8, .0, .0,
C      +       .0, .0, 7.10 E-8,
C      +       .0, 1.260E-7, .0,
C      +       1.489E-7, .0, .0,
C      +       1.90 E-9, .0, .0,
C      +       7.12 E-9, .0, .0,
C      +       2.60 E-9, .0, .0,
C      +       6.55 E-8, .0, .0,
C      +       .0, .0, 3.77 E-7,
C      +       2.25 E-8, .0, .0,
C      +       9.900E-8, .0, .0,
C      +       .0, .0, 3.67 E-6,
C      +       1.657E-8, .0, .0,
C      +       .0, .0, .0,
C      +       3.15 E-3, 2.19 E-7, .0,
C      +       9.10 E-3, 1.05 E-8, .0/
```

```

DO 10 J = 1,17
DO 10 K = 1,3
BR(J,K) = BRANCH(K,J)
10 C Calculate relative efficiencies
C DO 105 I = 1,23
ENLOG = ALOG(EN(I))
ENSQ = ENLOG**2
C !Cd abs. coef.
CDMU = EXP(13.26072-4.240781*ENLOG+.2761391*ENSQ)
C !Cd attenuation
CDMAS = EXP(-1.36*CDMU)
C !Pb abs. coef.
PBMU = EXP(27.72774-8.025776*ENLOG+.5248603*ENSQ)
C !Pb attenuation
PBABS = EXP(-PBMU*PBMAS)
C !Pu abs. coef.
PUMU = EXP(27.85913-7.840364*ENLOG+.4977653*ENSQ)
C !Pu attenuation
PUMUX = PUMU * (PUMAS +.1)
C !Efficiency correction
PUABS = (1.-EXP(-PUMUX))/PUMUX
105 EF(I)=CDMAS*PBABS*PUABS*EXP(ANSR(1)+ANSR(2)*ENLOG+ANSR(3)*ENSQ)
C CONTINUE
C Correct the branching intensities for changes in efficiency using
C the efficiency of the first peak as reference.
C DO 112 I = 1,17
RATIO = EF(1)/EF(I)
C !Except for 766 keV peak. It is ratioed to 769 peak
IF (I .EQ. 15 .OR. I .EQ. 16) GOTO 112
110 DO 110 K = 1,3
112 BR(I,K) = BR(I,K) * RATIO
C CONTINUE
C Read in 630 to 666 keV region and form equations for region
C NSTART = 4096
C !Starting channel
NST = NSTART +627.35/GAIN2 + ZERO2
C !Read in 600 keV region
CALL TRNSFR (200,YDATA(NST),Y1)
C !Transfer bkgrd from lower region
NS = 4.3/GAIN2
CALL TRNSFR (NS,Y1,Y1(NS+1))
C !Transfer bkgrd from higher region
NS = 6.9/GAIN2
NE = 38.9/GAIN2
SUM = 0.0
DO 302 I = NE,NE+8
302 SUM = SUM + Y1(I)
SUMIN = SUM
DO 304 I = NE,NE+20
304 IF (SUM .LE. SUMIN) THEN
SUMIN = SUM
JMIN = I
ENDIF
SUM = SUM - Y1(I) + Y1(I+9)
304 CONTINUE
CALL TRNSFR (8,Y1(JMIN),Y1(NE))
C !Get net counts
CALL NCNTS (NS,NS+7,NE,NE+8,NPTS,S1,S2,ERROR,AVBG3,Y1,Y1)
C !Find 646 keV peak
MS = 8.25/GAIN2

```



```

      CALL MAXVAL (MS,MS+13,JP,K,PHT,Y1)
      !Detr. shape parameters
      EPK = 646.
      CALL SPRAM (646.,GAIN2,SHAPC2,SG,ALPHA,F12M,EXP1)
      !Get 646 keV position
      CALL GFIT (JP,K,3,ALPHA,EXP1,EXP2,P646,HT646,ERROR,Y1)
      Determine a more accurate background value in the 648 keV region and
      make corresponding adjustments to the net counts this region.
      !Transfer data to new array
      CALL TRANSFR (30,Y1(JPK+1),RM)
      PKP = P646 - JPK
      !Subtract 646 pk.
      WL = 0.0
      MS = 1.77/GAIN2
      ME = MS + 8
      CALL STRIP1 (MS,ME,PKP,HT646,AL646,EXP1,EXP2,WL,RM)
      !Subtract 649 peak.
      CALL STRIP1 (MS,ME,PKP+3.3525/GAIN2,.0478*HT646,AL646,EXP1,
      EXP2,WL,RM)
      !Subtract 650 peak
      CALL STRIP1 (MS,ME,PKP+4.56/GAIN2,.018*HT646,AL646,EXP1,
      EXP2,WL,RM)
      !Subtract 652 peak.
      CALL STRIP1 (MS,ME,PKP+6.5/GAIN2,RM(26),AL646,EXP1,EXP2,WL,RM)
      SUM = 0.0
      DO 307 I = MS,ME
      !Integrate bkgrd region
      SUM = SUM + RM(I)
      !Ave. bkgrd per channel
      SUM = SUM / (ME-MS+1)
      ND = JPK + 10
      DEL1 = SUM / ND
      DEL2 = SUM / (NPTS-ND-1)
      DO 308 I = 1,NPTS
      YY = Y1(I)
      IF (I .LE. ND) THEN
      Y1(I) = Y1(I) - I*DEL1
      ELSE
      Y1(I) = Y1(I) - (NPTS-I)*DEL2
      END IF
      CONTINUE
      DO 310 J = 1,15
      GAMA(J) = 0.0
      HIGHT(J) = 0.0
      !Establish relative peak heights
      IF (BR(J,1) .GT. 0.) THEN
      !239Pu peaks referenced to 646 keV peak
      K = 1
      L = 5
      ELSE
      !241Am peaks referenced to 662 keV peak
      K = 3
      L = 13
      ENDIF
      IF (J .NE. L) HIGHT(J) = -BR(J,K)/BR(L,K)
      CONTINUE
      !Reference peak
      P646 = PK646
      PK646 = PK646 - (NST + NS+6)
      REFEN = EN(5) - PK646*GAIN2
      !Allow short- and long-term tailing amplitudes to be free.

```

```

      NFLG(6) = NOPFLG(5)-1
      IF (NOOUT .EQ. 0) NFLG(6) = -1
      NFLG(7) = NDEV3
      IF (NOPFLG(8) .EQ. 0) THEN
      CALL PKFIT (15,NPTS,NFLG,NYFLG,NXFLG,REFEN,GAIN2,AVBG3,EN,
      HIGHT,GAMA,Y1,DELY,RM,SHAPC2,CTS600,ER600)
      +
      Store 662 keV intensity of 137Cs
      CTSFIS(5) = CTS600(15)
      ERFIS(5) = ER600(15)
      Calculate the fission product abundances relative to 239Pu.
      DO 315 I = 1,5
      FCT = 1.57E-5*2.29E9*EF(23)/(EF(1+17)*CTS370(6)*BRFIS(1))
      2.29E9=.693*6.023E23/(24119*365*1440*60*239)
      CTSFIS(1) = CTSFIS(1)*FCT
      ERFIS(1) = ERFIS(1)*FCT
      315 CONTINUE
      ENDIF
      L = NDPTS
      IC = ICMP(15)
      !Uncertainty greater than .3%.
      A646 = CTS600(5)
      ER646 = SQRT(ER600(5)**2+.003*A646)
      !Approx. 646 keV eff relative to 100 keV
      EFF4 = A646/(ANS(2) *BR(5,1))
      K = 4
      !Enter two equations for 642/646 and 662/646 keV ratios.
      DO 320 J = 1,2
      L = L + 1
      RATIO = A646/CTS600(K)
      !Weighting factor.
      WTF = EFF4/(A646*SQRT((ER600(K)/CTS600(K))**2+(ER646/A646)**2))
      Y(L) = 0.0
      YNET(L) = CTS600(K)
      ACOEF(L,2) = -WTF*BR(5,1)
      ACOEF(L,3) = WTF*RATIO*BR(K,2)
      ACOEF(L,5) = WTF*RATIO*BR(K,3)
      IF (IC .GT. 0) ACOEF(L,IC) = ACOEF(L,5)
      K = 13
      320 CONTINUE
      NDPTS = L
      !Include 1001 keV peak as an equation
      IF (ICMP(11) .GT. 0) THEN
      GAIN2 = 645.969/(P646-ZERO2-NSTART)
      !Start of peak region
      NST = 982.5/GAIN2 + ZERO2
      NEND = 1023./GAIN2 + ZERO2
      IF (NEND .GT. 4095) NEND = 4095
      NPTS = NEND - NST
      !Read in peak region
      CALL TRANSFR (NPTS,YDATA(NST+NSTART),Y1)
      !Find peak position
      PK1001 = 1000.8/GAIN2 + ZERO2 - NST+1
      !End of bkgrd in front of peak
      NS2 = PK1001-15
      !Start of bkgrd in front of peak
      NS1 = NS2 - 20
      !Start of bkgrd aft of peak
      NE1 = NS2 + 45
      !End of bkgrd aft of peak
      NE2 = NE1 + 15

```

```

C      CALL AVE (1,20,SUM,AV,Y1)
C      SMIN = 1.E10
C
C      DO 56 I = 1,40
C          !Find the lowest 20 channel region
C          IF (SUM .LT. SMIN) THEN
C              SMIN = SUM
C              L = I
C          END IF
C          SUM = SUM - Y1(I) + Y1(I+20)
56      CONTINUE
C
C      DO 60 I = NS1,NS2
C          !Transfer bkgnd down from higher region
C          Y1(I+65) = Y1(I+101)
C          !Transfer bkgnd up from lower region
C          Y1(I) = Y1(L)
60      L = L+1
C
C      CALL NCNTS (NS1,NS2,NE1,NE2,NPTS,S1,A1001,ER1001,AV,Y1,Y1)
C
C      REFEN = (NST+NS2-1-ZERO2)*GAIN2
C          !Set analysis flags.
C      IF (Y1(14) .GT. Y1(23)) THEN
C          IXFLG(1) = 1
C          IXFLG(2) = -1
C          IXFLG(3) = -1
C      ENDIF
C      HIGHT(1) = 0.0
C      HIGHT(2) = 0.0
C      HIGHT(3) = 0.0
C
C      CALL PKFIT(3,NPTS,IFLG,IYFLG,IXFLG,REFEN,GAIN2,AV,
+          ENRG,HIGHT,GAMA,Y1,DELY,RM,SHAPC2,CTS,ERRR)
C
C      A1001 = CTS(1)+1.
C      ER1001 = ERRR(1)
C      RATIO = A646/A1001
C
C          Weight
C      WTF = EFF4/(A646*SQRT((ER1001/A1001)**2+(ER646/A646)**2
+          + EF1001**2))
C
C      IF (R6669 .GT. 0.0) THEN
C          NDPTS = NDPTS + 1
C          Y(NDPTS) = 0.0
C          WTF = EFF4/S6669
C
C          !238Pu coef.
C          ACOEF(NDPTS,1) = WTF * BR(16,2)
C          !239Pu coef.
C          ACOEF(NDPTS,2) = -WTF * (1.12E-7*R6669-2.75E-9)
C          !238U coef.
C          ACOEF(NDPTS,ICMP(11)) = WTF * BR(16,1)
C      END IF
C      NDPTS = NDPTS + 1
C      Y(NDPTS) = 0.0
C
C          !Save for long printout
C      YNET(NDPTS) = A1001
C
C          !238Pu coef.
C          ACOEF(NDPTS,1) = WTF*RATIO*BR(17,2)
C          !239Pu coef.
C          ACOEF(NDPTS,2) = -WTF * BR(5,1)
C          !238U coef.
C          ACOEF(NDPTS,ICMP(11)) = WTF * RATIO * BR(17,1)
C
C      END IF
C

```

```

500      RETURN
      END

```

SUBROUTINE MGACAL

MGACAL
=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Title : MGACAL
Author : Ray Gunnink
Date : 5/24/89
Language : Microsoft Fortran Ver. 4.01
Revision : 8.1 (13-June-1991)

Abstract :

This is a subroutine called by MGA. It first performs the least-squares analysis of the set of equations representing the isotopic data of the spectra and the corresponding uncertainty of the components that were freed in the fitting process. Following the fit, several flags and conditions are checked to determine data processing procedures. The second section of the routine evaluates the results of the fit and translates these values into isotopic ratios and percent abundances. The specific power values, the total specific power, and associated uncertainties of all reported values are also propagated from the statistically uncertainties related to the the spectral regions analysed.

Arguments :

Routines
Called : MATINV

Libraries
Used :

Revision History (Include date,revision I.D., and details)

Date of last revision: 14-Dec-1991

COMMON and variable declarations

INTEGER*4 ICOUNT
REAL*8 ACSQ,AC1,AC2,ANSWER
DOUBLE PRECISION XSUM(16),AP(2,2)

DIMENSION T1HLF(5),SPHT(5),C242(7,4),E1(5),E2(5),
+ RS(5),LC(5),ENB(3),INDX(16),PMTRX(16),XM(8,8),IE(8,8),
+ YAM(3),AC(3,3)

\$INCLUDE: 'MGACL.CMN'
\$INCLUDE: 'MGACOM.CMN'
\$INCLUDE: 'MGAHI.CMN'

Half lives of the isotopes

DATA T1HLF/87.74,24119.,6562.,14.348,433.6/

Specific power values for each of the isotopes

DATA SPHT/567.57,1.9288,7.0824,3.412,114.2/

Starting and ending energies for background regions.

DATA E1/.13,2.71,7.51,9.01,12.25/

DATA E2/.751,3.7,8.,9.47,100./

DATA ENB/90.,85.,75./

DATA C242/-9.687828E-5, 4.050273E-3, 2.160181E-2,0.,0.,0.,0.,
+ -1.000922E-0,-9.948685E-1,-9.726411E-1,0.,0.,0.,100.,
+ -1.008097E-0,-9.494748E-1,-9.421364E-1,0.,0.,0.,100.,
+ -2.680647,-.3865709,-.6504233,.0120555,-.01155217,-.01151055,
+ 150./

Convert units of the coefficients from relative DPM's to GRAMS of isotope.

DO 3 I = 1,NDPTS

ATNUM = 237.

DO 3 K = 1,16

IF (K .LE. 5) THEN

IF (K .LT. 5) ATNUM = ATNUM + 1.

ACOE(I,K) = 1.E15*ACOE(I,K)/(ATNUM *T1HLF(K))

!!E15 is an arbitrary constant to prevent overflow

ELSE

NC = ICMP(K)

IF (NC .GT. 0) THEN

IF (K .EQ. 7) ACOEF(I,NC)=1.E15*ACOE(I,NC)/(241.*T1HLF(4))

IF (K .EQ. 8) ACOEF(I,NC)=1.E15*ACOE(I,NC)/(237.*2.140E6)

IF (K .EQ. 9) ACOEF(I,NC)=1.E15*ACOE(I,NC)/(235.*7.040E8)

IF (K .EQ.11) ACOEF(I,NC)=1.E15*ACOE(I,NC)/(238.*4.468E9)

IF (K .EQ.15) ACOEF(I,NC)=1.E15*ACOE(I,NC)/(241.*T1HLF(5))

IF (K .EQ.16) ACOEF(I,NC)=1.E15*ACOE(I,NC)/(242.*3.760E5)

ENDIF

ENDIF

CONTINUE

The set of equations are now solved for the indicated unknowns by the method of least-squares.

MFLG = 0

Set 241Am homogeneity flag

NAMFLG = 0

```

1 DO 2 J = 1,NCMP
  XSUM(J) = 0.0
2 DD 2 K = 1,NCMP
  APHA(J,K) = 0.
  DO 4 I = 1,NDPTS
    DO 4 J = 1,NCMP
      XSUM(J) = XSUM(J) + ACOEF(I,J) * Y(I)
      DO 4 K = 1,NCMP
        AC1 = ACOEF(I,J)
        AC2 = ACOEF(I,K)
        ACSQ = AC1 * AC2
4      APHA(J,K) = APHA(J,K) + ACSQ
C
C      CALL MATINV(APHAS,16,NCMP,DET)
C      !Invert the matrix
      IF(DET.EQ.0.) THEN
        ERRFLG(1) = 9
        IF (NOOUT .GT. 0) WRITE (NDEV2,*)
          + ' MATRIX IS SINGULAR IN MGACAL.'
        RETURN
      ENDIF
C
      DO 5 J = 1,NCMP
        ANS(J) = 0.0
        answer = 0.0
        DM(J) = ABS(APHAS(J,J))
        DO 6 K = 1,NCMP
          !Compute the answers
6          ANS(J) = ANS(J) + XSUM(K) * APHA(J,K)
5          ans(j) = answer
          IF ((NOPFLG(8) .NE. 9) .AND. (MFLG .EQ. 0)) THEN
            NC = ICMP(13)
            IF (ABS(ANS(NC)) .GT. .003) THEN
              SHAPC(7) = SHAPC(7) + ANS(NC)
              NOPFLG(8) = 9
              ERRFLG(16) = 1
              IF (NOOUT .GT. 0) THEN
                BUFF=' Poor SHAPC(7) value. Repeat analysis.'
                IF (IPRINT .EQ. 'Y') WRITE (NDEV3,*)
                  + ' Poor SHAPC(7) value.'
                + ' Value will be adjusted and calculation repeated.'
              ENDIF
              GOTO 99
            ENDIF
          ENDIF
C
          !Calculate residuals
          DO 8 I = 1,NDPTS
            RM(I) = Y(I)
            DO 8 K = 1,NCMP
              RM(I) = RM(I) - ANS(K)*ACOE(I,K)
8          !Reset repeat flag
          NOPFLG(8) = 0
C
C      Examine residuals in 93.3 and 105 keV region for Th X rays.
C
      SUM = 0.0
      IF (ICMP(16) .EQ. 0) THEN
        NS = 5
        NE = 14
        DO 88 I = NS,NE
          SUM = SUM + RM(I)
88      !Accumulate residuals in 93 and 105 keV regions
      IF (SUM .GT. 30.) THEN
        ICMP(16) = SUM / 30.
        !Th X ray is present
        NOPFLG(8) = 5

```

```

C      !Set repeat flag
      ERRFLG(17) = 1
      IF (NOOUT .GT. 0) THEN
        WRITE (NDEV2,9079)
        IF (IPRINT .EQ. 'Y') WRITE (NDEV3,9079)
9079      FORMAT (' Th X rays were detected. Analysis will be repeated.')
      ENDIF
      GOTO 99
    END IF
  END IF
C
  SUMRM = 0.0
  SUMR = 0.0
  J = 0
  NE = 2
  NE2 = NPT100
  DO 7 I = 1,5
    RS(I) = 0.0
7
C
  DO 10 I = 3,NDPTS
    !Obtain total net count
    SUMR = SUMR + YNET(I)
    SUMRM = SUMRM + RM(I) * RM(I)
    IF (I .GT. NE) THEN
      !Average residual in region
      IF (J .GT. 0) RS(J) = RS(J)/(NE-NS+1)
      J = J + 1
      !Start channel of new region
      NS = E1(J)/GAIN
      !End channel of new region
      NE = E2(J)/GAIN
      !Center channel of region
      NNE = NE
      IF (NE .GT. NE2) NNE = NE2
      LC(J) = (NS+NNE)/2
    ENDIF
C
    IF ((I .GE. NS) .AND. (I .LE. NE) .AND. (I .LE. NE2)) THEN
      YT = YNET(I) + AVEBG
C
      !Sum of residuals for Jth region
      RS(J) = RS(J) + RM(I) * (.001*YT+SQRT(YT))
    ENDIF
10  CONTINUE
C
    !Check flag first
    IF (MFLG .EQ. 1) GOTO 128
    IF (NOPFLG(6) .EQ. 0) THEN
      !Redraw the background
      MFLG = 1
      RS(J) = RS(J)/(NNE-NS+1)
      J = 0
      NE = 0
      NS1 = 1
      IF (ICMP(8) .NE. 0) NS1 = 3
C
      DO 120 I = 1,NDPTS
        WTF = 1.0
        W1 = 1.0
        IF ((I .LT. NS1) .OR. (I .GT. NE2)) GOTO 108
        IF (I .GT. NE) THEN
          J = J + 1
          NS = LC(J)
          NE = LC(J+1)
          BGSLP = (RS(J+1)-RS(J))/(NE-NS+1)
          IF (J .EQ. 4) NE = NE + 20
        ENDIF
        YT = YNET(I) + AVEBG

```

```

      W1 = .001*YT+SQRT(YT)
      DBG = RS(J) + BGSLP*(I-NS)
C      !New YNET data values
      YNET(I) = YNET(I) - DBG
      YT = YNET(I) + AVEBG
      WTF = 1./(.001*YT + SQRT(YT))
C      !Rewrite equations
      Y(I) = YNET(I)*WTF
108      DO 110 K = 1,NCMP
110      ACOEF(I,K) = ACOEF(I,K) * WTF * W1 *ANS(2)
120      CONTINUE
      ANS2 = ANS(2)
      IF (ANS(5)/ANS(4) .GT. .05) THEN
        RAM241 = (A125/AREA(7))*E4825-.6577*T1HLF(4)/T1HLF(2)*
          + ANS(2)/ANS(4)*.045 *T1HLF(5)/T1HLF(4)
          + .6577*(6.9+5.4)*E-7/1.87E-6 & .045 = 1.87E-6/4.16E-5
        DIFF = 1.-RAM241*ANS(4)/ANS(5)
        CHNG = DIFF*(2./(2.+100.*ER125/A125))*(2./(2.+AREA(6)/A125))
        IF (ABS(CHNG) .GT. .01) THEN
          ERRFLG(18) = 100.*ABS(CHNG)
          IF (NOOUT .GT. 0) WRITE (NDEV3,9002) 100.*CHNG
9002      FORMAT(' WARNING: 129/148 keV efficiency changed in MGACAL by',
          + F5.1,'% ')
        ENDIF
      C      !Find index of 129/148 keV equation.
      JRM = NPT100 + 1
      IF (NOPFLG(1) .GT. 0) JRM=JRM+1
      IF (ICMP(9) .GT. 0) JRM=JRM+1
      IF (AREA(15) .GT. 0.) JRM=JRM+1
      ACOEF(JRM,4) = ACOEF(JRM,4) *(1.+CHNG)
      ENDIF
      GOTO 1
      ENDIF
C      !Chi-Square
128      IF (NOPFLG(5) .GE. 5) THEN
C      !Generate ASCII file of 100 keV data.
      NST = ST +ZERO
      XSTOP = -1.E10
      NDEV = 1
C      !Open file named "Y100.DAT"
      OPEN (UNIT=NDEV,FILE='Y100.DAT',ACCESS='SEQUENTIAL',
      + STATUS='UNKNOWN')
C      !Write 100 keV spectrum data to file
      WRITE (NDEV,9019) (YDATA(I),I=NST+2,NPT100+NST)
      WRITE (NDEV,9019) XSTOP
9019      FORMAT (1X,1PE15.7)
      DO 130 J = 1,9
        K = J
C      !Get index of 237Np, if any.
        IF (K .EQ. 7) K = ICMP(8)
C      !Get index of 235U, if any.
        IF (K .EQ. 8) K = ICMP(9)
C      !Get index of U x ray, if any.
        IF (K .EQ. 9) K = ICMP(10)
        IF (K .LE. 0) GOTO 130
        DO 129 I = 3,NPT100
          YT = YNET(I) + AVEBG
          WTF = 1./(.001*YT +SQRT(YT))
C      !Write component profiles to file.
          WRITE (NDEV,9019) ACOEF(I,K)/WTF*ANS(K)
129      CONTINUE
130      CONTINUE
      WRITE (NDEV,9019) XSTOP
      CLOSE(NDEV)
      ENDIF

```

```

C      DOF = NDPTS - NCMP
C      !Degrees of freedom
      QFIT = SUMRM / (DOF-1.)
      IF (QFIT .LT. 1.0) QFIT = 1.0
C      !Reduced Chi-Square
      RMSD = 1. + (QFIT-1.)/(10. +SUMR/(1000*DOF))
      ERRFLG(20) = 50.*(RMSD-1.)
C      !Intensity-normalized
      NC = ICMP(10)
      IF (NC .GT. 0) THEN
        IF ((ANS(NC) .LT. 0.) .OR. (SQRT(DM(NC))/ANS(NC) .GT. .1)) THEN
          IF (NOOUT .GT. 0) THEN
            WRITE (NDEV2,9026)
            IF (IPRINT .EQ. 'Y') WRITE (NDEV3,9026)
9026      +      FORMAT (' Analysis indicates fluorescence U xray is not ',
            + 'present.')
          ENDIF
          NOPFLG(8) = 1
          ICMP(8) = 0
C      !Make sure the 237Np flag is reset.
          ICMP(10) = -1
          RETURN
        ENDIF
      ENDIF
C      !Check if sample is fresh
C      IF (NOPFLG(1) .NE. 0) GOTO 11
C      Check for an imbalance in the fit of the 203-208 keV peaks.
C      Negative deviation indicates the presence of Uranium in sample.
C      IF ((NOPFLG(2) .NE. 0) .OR. (PUGPSC .LT. .01)) GOTO 11
      JRM = NPT100 + 1
      IF (ICMP(9) .GT. 0) JRM = JRM + 1
      IF (AREA(15) .GT. 0.) JRM = JRM + 1
      IF (((RM(JRM)+1.)/ERR(6)+(RM(JRM+2)+1.)/ERR(10)).GT..03)
      +      GOTO 11
      IF (NOOUT .GT. 0) THEN
        WRITE (NDEV2,9080)
        IF (IPRINT .EQ. 'Y') WRITE (NDEV3,9080)
9080      FORMAT (' U x rays detected in MGACAL. Will repeat analysis.')
      ENDIF
      NOPFLG(2) = 1
      NOPFLG(8) = 2
      ICMP(8) = 0
      ICMP(10) = 1
      RETURN
C      11      IF (NOOUT .GT. 0) THEN
        IF (NOPFLG(5) .GT. 1) THEN
          IF (NOPFLG(5) .GT. 2) THEN
            DO 126 J = 1,NCMP
              WRITE (NDEV3,9031) (APHA(J,K),K=1,J)
9031      +      FORMAT (/1X,1P16E9.2)
            CONTINUE
          ENDIF
          DO 121 K = 1,NCMP
            DO 121 J = 1,16
              IF (ICMP(J) .EQ. K) INDX(K) = J
            CONTINUE
          121      CONTINUE
          WRITE (NDEV3,9032) (INDX(K),K=7,NCMP)
9032      +      FORMAT (/5X,'Correlation Matrix',
          + '      238      239      240      241      Am      XRay',16,917)
          DO 122 K = 1,NCMP
            DO 124 J = 1,K

```

```

124 PMTRX(J) = APHA(J,K)/(SQRT(APHA(J,J)*APHA(K,K)))
WRITE (NDEV3,9034) INDX(K), (PMTRX(J),J=1,K)
9034 FORMAT (I3,16F7.3)
122 CONTINUE
    ENDF
    ENDF
C
    SUM1 = 0.0
    SUM2 = 0.0
C
C The results of the least-squares solution of the set of equations used are
C in units of relative disintegration rates of each of the isotopic
C components. These results are now decay corrected (if requested), and
C reinterpreted for reporting purposes.
C
    PCTDIF = 0.0
C
C      ICheck 241Am heterogeneity flag
    IF (ICMP(15) .NE. 0) THEN
C
C      IColumn index for 600keV 241Am cmpnt
        NC = ICMP(15)
C
C      ITotal uncertainty of Am results
        ERTOT = SQRT (DM(5) + DM(NC))
C
C      IDifference in units of std. devs.
        STDDEV = ANS(NC) / ERTOT
C
C      IFind start of 300 keV equations.
        DO 12 I = NPT100,NDPTS
        IF (ACOE(I,NC) .NE. 0.0) GOTO 13
12 CONTINUE
13 NS = I
C
C      IDifference in 335 and 662 keV peaks
        RMDIF = ABS(RM(NS+1)-RM(NS+5))
        AM100 = ANS(5)
        DM100 = DM(5)
        IF (ABS(STDDEV) .GT. 3. .OR. RMDIF .GT. 10) THEN
            IF (NAMFLG .EQ. 0) THEN
                NCMP = NCMP+1
C
C      ITransfer 300 keV coeffs. to last column.
                DO 14 I = NS,NS+3
                ACOEF(I,NCMP) = ACOEF(I,NC)
14 ACOEF(I,NC) = 0.0
                NAMFLG = 1
                MFLG = 0
C
C      IRepeat analysis
                GOTO 1
            ELSE
C
C      IPercent difference in 100 and 600 keV answers.
                PCTDIF = 100.*ANS(NC)/ANS(5)
C
C      I241Am abundance based on 300 keV region.
                AM300 = ANS(5) + ANS(NCMP)
                DM300 = DM(5) + DM(NCMP)
C
C      I241Am abundance based on 600 keV region.
                AM600 = ANS(5) + ANS(NC)
                DM600 = DM(5) + DM(NC)
C
C We now have three results for 241Am based on analyses corresponding
C the 100, 300, and 600 keV region. The true average 241Am abundance
C relative to the Plutonium content of the sample depends on the
C nature and description of the segregation. J. Fleissner as proposed
C that the efficiency displacement is proportional to EXP(Beta/Energy).
C This formulation as been applied principally to the analysis of
C high 241Am salts containing a distribution of small Pu particles.
C Although this equation cannot be derived from first principles,
C and certainly should not be used to describe all heterogeneous
C materials, it is implemented in the following section to somewhat
C improve the 241Am result.

```

```

    WT100 = AM100/SQRT(DM100)
    WT300 = AM300/SQRT(DM300)
    WT600 = AM600/SQRT(DM600)
    YAM(1) = LOG(AM100)*WT100
    YAM(2) = LOG(AM300)*WT300
    YAM(3) = LOG(AM600)*WT600
    AC(1,1) = WT100
    AC(2,1) = WT300
    AC(3,1) = WT600
    AC(1,2) = WT100/.17
    AC(2,2) = WT300/.35
    AC(3,2) = WT600/.66
C
    DO 132 J = 1,2
    XSUM(J) = 0.0
    DO 132 K = 1,2
132 AP(J,K) = 0.0
    DO 133 I = 1,3
    DO 133 J = 1,2
    XSUM(J) = XSUM(J) + AC(I,J)*YAM(I)
133 DO 133 K = 1,2
    AP(J,K) = AP(J,K) + AC(I,J) * AC(I,K)
C
    NK = 2
    CALL MATINV (AP,NK,NK,DET)
C
    DO 134 J = 1,2
    AA(J) = 0.0
    DO 134 K = 1,2
134 AA(J) = AA(J) + XSUM(K)*AP(J,K)
    ANS(5) = EXP(AA(1))
    DM(5) = ABS(AP(1,1))*(ANS(5))**2
    APHA(5,5) = DM(5)
C
    ENDF
    ELSE
C
C      IPerform a weighted average.
        ANS(5)=ANS(5)*(1.+ANS(NC)/ANS(5)*DM(5)/(DM(5)+DM(NC)))
    ENDF
C
    ENDF
C
    SUMSG = 0.0
    DO 30 J = 1,5
    ER(J) = 100.*SQRT(QFIT *DM(J)) /ANS(J)
C
C      IStatistical error
        GRMS(J) = ANS(J)
C
C      Igrams of J
        DCNST = .693147 / (365.25*T1HLF(J))
C
C      IDecay constant
        GRMZ(J) = GRMS(J) * EXP (DCNST * DLTAT)
C
C      IGrams at zero time
        IF (J .GT. 4) GOTO 30
C
C      IOnly add up Pu
        SUM1 = SUM1 + GRMS(J)
C
C      ISum of grams
        SUM2 = SUM2 + GRMZ(J)
C
C      ISum at zero time
        SIG = ER(J) * GRMZ(J)/100.
        SUMSG = SUMSG + SIG**2
30 CONTINUE
    SIG = (100./SUM2)*SQRT(SUMSG)
C
C      ITime since 241Am separation
    TZ = -14.35/.693 *ALOG(GRMS(4)/(GRMS(4)+GRMS(5)))
    IF (TZ .LT..12 .AND. NOPFLG(1).EQ.0 .AND. NOPFLG(2).GT.0) THEN

```



```

C      !Very fresh sample. 237U may be out of equil. Repeat.
      NOPFLG(1) = -1
      NOPFLG(2) = 0
      NOPFLG(8) = 10
      ICMP(7) = 1
      ICMP(10) = 0
      ERRFLG(16) = ERRFLG(16) + 2
      GOTO 99
    ENDIF
    NPU = 1
C      !Default option: use input value
      ER242 = .01
C      !Assume 1% error in input 242Pu value
      IF (PU242) 32,34,36
C      !Options for treating 242Pu content
32      NPU = 0
C      !Calc. 242Pu using algorithm
      P241 = (GRMS(4)+GRMS(5))*EXP(.048135*COLTM)
C      !241Pu decay during cooling
      PU242 = GRMS(3)/GRMS(2)*P241/GRMS(2)*53.
C      !Calculated 242Pu content
C      !Assume 10% error in 242Pu calc.
      ER242 = .1
      GO TO 36
34      NPU = -1
C      !Use following default value
C      Calculate 242Pu abundance using new algorithm
      FCTR = 100./(SUM1+GRMS(5))
      PU239 = GRMS(2) * FCTR
      LL = 1
      DO 33 I = 1,3
      IF (PU239 .LT. ENB(I)) LL = LL + 1
33      CONTINUE
      PU242 = C242(7,LL)
      DO 35 I = 1,3
      CC = GRMS(I+1) * FCTR
      IF (I .EQ. 3) CC = CC + GRMS(5)*FCTR
      PU242 = PU242 + C242(I,LL) * CC + C242(I+3,LL) * CC**2
35      CONTINUE
      IF (PU242 .LT. 0.) PU242 = 0.0
C      !Assume 10% error in 242Pu calc.
      ER242 = .1
36      IF (PU242 .GT. 50.) THEN
C      Determine 242Pu content from spectrum
      NC = ICMP(16)
      G242 = ANS(NC)
      PU242 = 100.*G242/(G242+SUM1)
      ER242 = SQRT(QFIT*DM(NC))/G242
      SUM1 = SUM1 + G242
      SUM2 = SUM2 + G242
    ELSE
C      !Correction factors
      P242 = 1. - .01 * PU242
      FCTR = 1.+ .01*PU242*(SUM2-SUM1)/SUM1
C      !Correct total for missing 242Pu
      SUM1 = SUM1 *FCTR /P242
      SUM2 = SUM2 / P242
    ENDIF
      PU242C = PU242 * (1.+(SUM2-SUM1)/SUM1)
C
      DL = DLTAT*DCNST
      EX = EXP(DL)
      X241 = (GRMS(4)-GRMZ(4))*EXP(-.5*DL)
C      !241Am at zero time

```

```

      GRMZ(5) = EX *(GRMS(5) + X241)
      IF (ABS(PCTDIF) .GT. 0.0) THEN
        DM100 = 100. *SQRT(QFIT *DM100) /AM100
        AM100 = 100.*EX *(AM100 + X241)/SUM2
        DM300 = 100. *SQRT(QFIT *DM300) /AM300
        AM300 = 100.*EX *(AM300 + X241)/SUM2
        DM600 = 100. *SQRT(QFIT *DM600) /AM600
        AM600 = 100.*EX *(AM600 + X241)/SUM2
      ENDIF
C      !Preserve 239Pu result as a reference
      G239 = GRMS(2)
      POW = 0.0
      POWCNT = 0.0
C
      DO 38 J = 1,5
      PCT1(J) = GRMS(J) / SUM1 * 100.
C      !Percent abundance of Jth isotope
      PCT2(J) = GRMZ(J) / SUM2 * 100.
C      !Percent abundance at zero time
      SPPOW(J) = .01 * SPHT(J) * PCT2(J)
C      !Specific power due to isotope J
      POW = POW + SPPOW(J)
C      !Accumulate total power
      POWCNT = POWCNT + SPHT(J)*PCT1(J)
C      !Sum of count-time power
      GRMS(J) = GRMS(J) / G239
C      !Abundance relative to 239Pu
      IF (J .NE. 2) ER2(J) = SQRT(ER(2)**2+ER(J)**2-2.E4*APHA(2,J)
+      *QFIT/(ANS(2)*ANS(J)))
      GRMZ(J) = GRMZ(J) / G239
38      CONTINUE
      ER2(2) = 0.0
C
      SP242 = .01 * PU242 * .1146
C      !Specific power of 242Pu
      POW = POW + SP242
C      !Add to total power
C
C      The following set of calculations propagate the statistical uncertainties
C      of the original results to the quantities being reported. The treatment
C      acknowledges the covariant dependency of the normalized isotopic
C      abundances following the procedures reported in J. of INMM, Vol. 12, No.2,
C      (1983)
C
      SGPOW = 0.0
      XPCT = (.01*PU242 *ER242 *SUM2)**2
      DO 42 J = 1,5
C      !Count/zero time fractions of J component
      FRAC1 = .01 * PCT1(J)
      FRAC2 = .01 * PCT2(J)
C      Start with 242Pu component. Treat as independent variable.
      ER1(J) = ((1.-.01*PU242)*FRAC1 *PU242 *ER242/PCT1(2))**2
      PCT(J) = ((1.-.01*PU242)*FRAC2 *PU242 *ER242/PCT2(2))**2
C      !Save XPCT for calculations in MGAOUT.
      IF (J .LT. 5) XPCT = XPCT + APHA(J,J)
      NK = 4
C      !If component is 241Am, set new limit.
      IF (J .EQ. 5) NK = 5
C      !Do K-row by L-column integration.
      DO 41 K = 1,NK
C      !Assume it is a diagonal element (ie. K=J)
      FK1 = 1.-FRAC1
      FK2 = 1.-FRAC2
      IF (K .NE. J .OR. J .EQ. 5) THEN
C      !K not= J or component is 241Am.
      FK1 = -FRAC1

```



```

      FK2 = -FRAC2
    ENDIF
    IF (K .EQ. 5) THEN
      !Component is 241Am.
      FK1 = 1.0
      FK2 = 1.0
    ENDIF
    DO 40 L = 1,NK
      !Assume it is a diagonal element (ie. L=J)
      FL1 = 1.-FRAC1
      FL2 = 1.-FRAC2
      IF (L .NE. J .OR. J .EQ. 5) THEN
        !L not= J or component is 241Am.
        FL1 = -FRAC1
        FL2 = -FRAC2
      ENDIF
      IF (L .EQ. 5) THEN
        !Component is 241 Am.
        FL1 = 1.0
        FL2 = 1.0
      ENDIF
      !Integrate sigma squared.
      ER1(J) = ER1(J) + FK1 *FL1 *APHA(K,L)
      PCT(J) = PCT(J) + FK2 *FL2 *APHA(K,L)
      !Also integrate specific power error.
      IF (J .EQ. 5) THEN
        IF (K .EQ. 5) THEN
          FL3 = SPHT(5)
        ELSE
          FL3 = SPHT(K)-POW
        ENDIF
        IF (L .EQ. 5) THEN
          FL4 = SPHT(5)
        ELSE
          FL4 = SPHT(L)-POW
        ENDIF
        SGPOW = SGPOW + FL3 *FL4 *APHA(K,L)
      ENDIF
    CONTINUE
  40 CONTINUE
      !Calculate uncertainties.
      ER1(J) = 100.*SQRT(QFIT*ER1(J))/(SUM1*FRAC1)
      PCT(J) = 100.*SQRT(QFIT*PCT(J))/(SUM2*FRAC2)
  42 CONTINUE
      !Adjust t-zero 241Am uncertainty.
      PCT(5) = SQRT((ER1(5)*GRMS(5))**2+(PCT(4)*(GRMZ(4)-GRMS(4)))**2)
      + /GRMZ(5)
      !Uncertainty in specific power
      SGPOW=100.*SQRT((QFIT*SGPOW)/(SUM2*POW)**2+(.01*ER242*PU242)**2)
      XPCT = 1.E4*XPCT/SUM2**2
      W122 = SQRT (SHAPC(1) + 122.*SHAPC(2))
      IF (W122 .GT. .6) ERRFLG(19) = 1
      IF (SHAPC(2) .GT. .0019) ERRFLG(19) = ERRFLG(19)+2
      !Store data into \PUI\OUTMGA.TXT file
      !Calculate 122 and 208 keV resolutions.
      ICOUNT = COUNTS
      W208 = SQRT (SHAPC(1) + 208.*SHAPC(2))
      R3539 = 0.
      NC = ICMP(9)
      !235U index
      PCTA = 0.0
      IF (NC .GT. 0 .AND. ANS(NC) .GT. 0.) THEN
        GMS235 = ANS(NC)

```

```

      R3539 = GMS235/G239
      !U235 to Pu239 ratio
      PCTA = 100. * GMS235/SUM1
      !% U235 relative to Pu at count time
      PCTB = 100. * GMS235/SUM2
      !% U235 relative to Pu at zero time
      ER5 = 100*SQRT(DM(NC))/ANS(NC)
      ER5S = ER5**2
      ER235 = SQRT(XPCT+ER5S)
      ERR5 = SQRT(ER(2)**2+ER5S)
      !error in U235 analysis
    ENDIF
    NC = ICMP(11)
    !Index for U238. NC = 0 for 1 Det. system
    IF (NC .GT. 0) THEN
      GMS238 = ANS(NC)
      R3839 = GMS238 / G239
      !U238 to Pu239 ratio
      PCTA8 = 100.*GMS238 / SUM1
      !% U238 relative to Pu at count time
      PCTB8 = 100.*GMS238 / SUM2
      !% U238 relative to Pu at zero time
      ER8 = 100.*SQRT(DM(NC))/ANS(NC)
      ER8S = ER8**2
      ER238 = SQRT(XPCT+ER8S)
      ERR8 = SQRT(ER(2)**2+ER8S)
      !% error in U238 analysis
    ENDIF
    NC = ICMP(8)
    IF (NC .GT. 0) THEN
      !237Np present.
      PCTNP = 100.*ANS(NC)/SUM1
      ERNP = SQRT(XPCT+2.E4*DM(NC)/ANS(NC)**2)
    ENDIF
    UPU = 0.
    ERUPU = 0.0
    NC = ICMP(10)
    IF (NC .GT. 0) THEN
      UXRAY = ANS(NC)
      UPU = ANS(NC)/ANS(6)
      UPU = UPU - .3*(UPU/(UPU+1.))
      ERUPU=100.*SQRT(QFIT*(DM(6)/ANS(6)**2+DM(NC)/ANS(NC)**2))
    ENDIF
    RAM241 = GRMS(5)/GRMS(4)
    ERRAM = SQRT(ER(4)**2+ER(5)**2)
    TGRMS = GRMS(4)+GRMS(5)
    ERRTZ= .207*SQRT(ER(4)**2+((ER(4)*GRMS(4))**2+
      + (ER(5)*GRMS(5))**2)/(TGRMS**2)+1.)
    E242 = 100. * ER242
    PEFF = 2.49*PCT1(1)+PCT1(3)+1.57*PU242
    EPEFF = SQRT(((2.49*PCT(1)*PCT1(1))**2+(PCT(3)*PCT1(3))**2
      + (1.57*E242*PU242)**2))/PEFF
    ISPEC = YDATA(1)
    !Prepare to write out the covariance matrix.
    DO 50 I = 1,8
      K = I
      IF (I .GT. 6) THEN
        K = 0
        IF (ICMP(I+2) .GT. 0) K = ICMP(I+2)
      ENDIF
      DO 50 J = 1,8
        REAL = 0.0
        IF (K .GT. 0) THEN

```

```

      IF (J .LE. 6) THEN
        REAL = APHA(K,J)
      ELSE
        IF (ICMP(J+2) .GT. 0) REAL = APHA(K,ICMP(J+2))
      ENDIF
      IF (REAL .NE. 0.) CALL SPLIT(REAL,XM(I,J),IE(I,J))
    ENDIF
50  CONTINUE
    C      Calculate sample concentration if flag is set.
    IF (NTYPE .GT. 0) THEN
      IF (NTYPE .EQ. 2) THEN
        CCNST = CONC *VOLUME *TIME/(SUM2 *ANS2)
        CALL MGASOL
      ELSE
        CONC = 1000.*CCNST*SUM2*ANS2/(VOLUME*TIME)
        SIG = SQRT(SIG**2+ERR88**2 +CDERR**2 +.01)
      ENDIF
    ENDIF
    C
    IF (NOPFLG(3) .EQ. 6) THEN
      OPEN (UNIT=NDEV9,FILE='\\PUIS\MGA\OUTMGA.TXT',
+      ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
      WRITE (NDEV9,9100) ICOUNT,W122,W208,QFIT,RMSD,GRMS(1),GRMS(3),
+      GRMS(4),GRMS(5),R3539,UPU,ERUPU,((XM(I,J),IE(I,J),J=1,8),I=1,8),
+      (INT(CNTS(I)),I=1,10),PU242,.1*PU242,PEFF,.01*EPEFF*PEFF,POW,
+      .01*SGPOW*POW,TZ,ERRTZ,TIME,IDT1,ISPEC,(ERRFLG(I),I=1,25)
9100  FORMAT (I15,2F6.4,2F7.3,7F10.6,36(F9.6,I3),10I8,6F10.6,2F7.3,
+      F9.1,I2,I8,25I1)
      CLOSE (NDEV9)
    C
    ENDIF
    RETURN
99  ICMP(8) = 0
    C
    RETURN
  END

```

SUBROUTINE MGAOUT

```

C*****
C
C          MGAOUT
C          =====
C*****
C
C      Developed for : Safeguards Technology Program
C                    Nuclear Chemistry Division
C                    Lawrence Livermore National Laboratory
C                    Livermore, CA. 94550
C*****
C
C      Title       : MGAOUT
C      Author      : Ray Gunnink
C      Date        : 5/24/89
C      Language    : Microsoft Fortran  Ver. 4.01
C      Revision    : 9.0 (December-1991)
C*****
C
C      Abstract :
C
C      This is the final subroutine called by MGA. The results are
C      reported and net counts of each equation and the corresponding
C      weighted residual is also reported
C*****
C
C      Arguments :
C*****
C
C      Routines
C
C      Libraries
C      Used :
C*****
C
C      Revision History ( Include date, revision I.D., and details ) *
C*****
C
C      Date of last revision: 14-Dec-1991
C*****
C
C      COMMON and variable declarations *
C*****
C
C      CHARACTER*9 ADATE
C      CHARACTER*3 AMONTH(2)
C      CHARACTER*5 LABLE(6)
C      DIMENSION ENRG(300),ENG(4),YNT(4),RMO(4),PKEN(18)
C
C-----
C$INCLUDE: 'MGAOL.CMN'
C$INCLUDE: 'MGAH1.CMN'
C$INCLUDE: 'MGACOM.CMN'
C-----

```

```

C
C      DATA PKEN/43.5,45.2,51.6,185.7,205.,129.3,152.7,203.5,311.9,
C      + 332.3,335.4,368.6,370.9,376.6,642.6,662.4,766.4,1001.2/
C
C      DATA LABLE/' 95Zr',' 95Nb','103Ru','106Ru','137Cs','144Ce'/
C-----
C
C      IW122 = 1000.*SQRT(SHAPC(1)+122.*SHAPC(2))
C      IW208 = 1000.*SQRT(SHAPC(1)+208.*SHAPC(2))
C      AMONTH(1) = CMONTH(MONTH)
C      IF (MONTH.EQ. 0) AMONTH(1) = ' '
C      AMONTH(2) = CMONTH(MNTHTO)
C      IF (MNTHTO.EQ. 0) AMONTH(2) = ' '
C      IF (ICMPTR.EQ. 0) THEN
C          WRITE (NDEV3,1002) NDAY,AMONTH(1),ISTYR
C1002  FORMAT (20X,'MGApc (PC Version 9.0 23-Jan-92) Date: ',
C      +      12,'-',A,'-',12)
C      ELSE
C          CALL DATE (ADATE)
C          WRITE (NDEV3,1003) ADATE
C1003  FORMAT (32X,'MGA (Version 9.0 6-Mar-92) Date: ',A)
C      ENDIF
C      WRITE (NDEV3,1013)
C1013  FORMAT ( )
C      IF (ERRFLG(3).EQ. 1) WRITE (NDEV3,9016)
C9016  FORMAT (' NO INPUT LIVE TIME. Will set live time to = 1 min.')
C      WRITE (NDEV3,9018) SPC1ID(1:40),TIME,IDT1
C9018  FORMAT (' Spec. ID: ',A,' LT:',F6.1,' Mins DT:',12,'%')
C
C      !Skip if no 2nd spectrum given
C      IF (SPC2ID(1:1).NE. ' ')WRITE(NDEV3,9020)SPC2ID(1:37),TIM,IDT2
C9020  FORMAT(' Spec. #2 ID: ',A,' LT:',F6.1,' Mins DT:',12,'%')
C
C      IF (NOPFLG(1).EQ. 0)WRITE (NDEV3,9022) CDATE,NDAY,
C      +      AMONTH(1),ISTYR,NDAYTO,AMONTH(2),ITOYR
C9022  FORMAT (16X,A,' Date: ',12,'-',A,'-',12,
C      +      ' Declared Date: ',12,'-',A,'-',12)
C      WRITE(NDEV3,1013)
C      WRITE (NDEV3,9024) SAMPID(1:26),SFIL(7:8),COUNTS
C9024  FORMAT (' Sample ID: ',A,'Detector: ',A,
C      +      ' Total Counts:',1PE10.3)
C      IF (COLTM.GT. 0) WRITE (NDEV3,9025)COLTM
C9025  FORMAT (' Reactor Discharge-to-Chem. Sep. Time (yrs): ',F4.1)
C
C      IF (NOPFLG(5).GT. 0)WRITE (NDEV3,9026) ANS(ICMP(13)),RSLP,
C      +      SHAPC(10),SHAPC(2)
C9026  FORMAT (' Tail Amp =',1PE9.2,' Res. slope=',E9.2,
C      +      ' Bkgrd Slope =',0PF6.3,'%Ch./46X','Resolution slope =',F7.5)
C
C      WRITE (NDEV3,9029) PUGPSC,CDABS,IW122
C9029  FORMAT (/ ' PU g/cm2 =',F7.4,' CD g/cm2 =',F6.3,14X,
C      +      'FWHM at 122 keV =',14,' eV')
C
C      WRITE (NDEV3,9032) QFIT,IW208
C9032  FORMAT (' QFIT =',F5.2,44X,'at 208 keV =',14,' eV')
C      IF (IREPRT.EQ. 0) THEN
C          WRITE (NDEV3,9033) RMSD
C9033  FORMAT (' NQFIT= ',F5.3,22X,'ISOTOPIC ANALYSIS AT'/
C      +      9X,'RELATIVE %' *1.5X,'MEAS. DATE DECLARED DATE',
C      +      3X,'SPECIFIC POWER',/9X,'ABUNDANCE ERR WT.PCT. %ERR',
C      +      ' WT.PCT. %ERR (MILLIWATTS/GM)')
C      WRITE (NDEV3,9034) (237+J,GRMS(J),ER(J),ER2(J),PCT1(J),ER1(J),
C      +      PCT2(J),PCT(J),SPPOW(J), J= 1,4)
C9034  FORMAT (' Pu',13,' =',F9.6,2F5.1,F11.5,F6.2,F11.5,F6.2,F10.5)
C      ELSE

```

```

9035  WRITE (NDEV3,9035) RMSD
      FORMAT (' NQFIT= ',F5.3,37X,'ISOTOPIC ANALYSIS AT'/
+ 9X,'RELATIVE %'*,RELATIVE %* MEAS. DATE ',
+ ' DECLARED DATE'/9X,'TO Pu239 ERR ERR TO Pu241 ERR ',
+ ' WT.PCT. %ERR WT.PCT. %ERR')
      DO 10 J = 1,5
      RAT241 = GRMS(J)/GRMS(4)
      ER241 = 0.0
      IF (J .NE. 4) ER241=SQRT(ER(4)**2+ER(J)**2-2.E4*APHA(4,J)*
+ QFIT/(GRMS(4)*GRMS(J)))
      IF (J .LT. 5) WRITE (NDEV3,9036) 237+J,GRMS(J),ER(J),ER2(J),
+ RAT241,ER241,PCT1(J),ER1(J),PCT2(J),PCT(J)
9036  FORMAT (' Pu',I3,' =',F9.6,2F5.1,F10.4,F5.1,F11.5,F6.2,F11.5,
+ F6.2)
10    CONTINUE
      ENDIF
C
      IF (NPU) 53,54,55
C
53    IF (IREPRT .EQ. 0) THEN
      WRITE (NDEV3,9037) PU242C,PU242,SP242
9037  FORMAT (' Pu242 = (NEW ALGORITHM)',F12.3,F17.3,' (10)',
+ F10.5)
      ELSE
      WRITE (NDEV3,9038) PU242C/PCT1(4),PU242C,PU242
9038  FORMAT (' Pu242 = (NEW ALGORITHM)',F13.4,' (10)',F9.5,
+ ' (10)',F11.5,' (10)')
      ENDIF
      GO TO 56
C
54    WRITE (NDEV3,9039) PU242C,PU242,SP242
9039  FORMAT (' Pu242 = (OLD ALGORITHM)',F12.3,F17.3,' (10)',
+ F10.5)
      GO TO 56
C
55    IF (PU242 .LT. 50) THEN
      IF (IREPRT .EQ. 0) THEN
      WRITE (NDEV3,9040) PU242C,PU242,SP242
9040  FORMAT (' Pu242 = (DECLARED)',2F17.3,' (1)',F10.5)
      ELSE
      WRITE (NDEV3,9240) PU242C/PCT1(4),PU242C,PU242
9240  FORMAT (' Pu242 = (DECLARED)',F18.4,' (1)',2(F10.5,
+ ' (1)'))
      ENDIF
      ELSE
      NC = ICMP(16)
      PCTA = 100.*ANS(NC)/SUM1
      PCTB = 100.*ANS(NC)/SUM2
      R4239 = ANS(NC)/G239
      ER42 = 100.*SQRT(QFIT*DM(NC))/ANS(NC)
      ER2S = ER42**2
      ER242 = (1.-.01*PCTA)*ER42
      SGPOW = SQRT(SGPOW**2+(.01*ER242*PCTA)**2)
      ERR42 = SQRT(ER(2)**2+ER2S)
      WRITE (NDEV3,9041) R4239,ER42,ERR42,PCTA,ER242,PCTB,ER242,
+ SP242
9041  FORMAT (' Pu242= ',F10.6,2F5.1,F11.5,F6.2,F11.5,F6.2,F10.5)
      ENDIF
56    IF (IREPRT .EQ. 0) THEN
      WRITE (NDEV3,9042)GRMS(5),ER(5),ER2(5),PCT1(5),ER1(5),
+ PCT2(5),PCT(5),SPPOW(5)
9042  FORMAT (' Am241 =',F9.6,2F5.1,F11.5,F6.2,F11.5,F6.2,F10.5)
      ELSE
      WRITE (NDEV3,9242)GRMS(5),ER(5),ER2(5),RAT241,ER241,PCT1(5),
+ ER1(5),PCT2(5),PCT(5)
9242  FORMAT (' Am241 =',F9.6,2F5.1,F10.4,F5.1,F11.5,F6.2,F11.5,

```

```

+ F6.2)
      ENDIF
      NC = ICMP(9)
C
      IF (NC .GT. 0) THEN
      IS = 235
      IF (IREPRT .EQ. 0) THEN
      WRITE (NDEV3,9043) IS,R3539,ER5,ERR5,PCTA,ER235,PCTB,ER235
9043  FORMAT (' U',I3,' =',F8.5,F6.1,F5.1,2(F10.4,F7.2))
      ELSE
      WRITE (NDEV3,9044) IS,R3539,ER5,ERR5,PCTA,ER235,PCTB,ER235
9044  FORMAT (' U',I3,' =',F8.5,F6.1,F5.1,15X,2(F10.4,F7.2))
      ENDIF
      ENDIF
      NC = ICMP(11)
C
      !Index for U238. NC = 0 for 1 Det. system
      IF (NC .GT. 0) THEN
      IF ((PCTA8 .GT. 0.) .AND. (ER238 .LT. 30.)) THEN
      IS = 238
      IF (IREPRT .EQ. 0) THEN
      WRITE (NDEV3,9043)IS,R3839,ER8,ERR8,PCTA8,ER238,PCTB8,ER238
      ELSE
      WRITE (NDEV3,9044)IS,R3839,ER8,ERR8,PCTA8,ER238,PCTB8,ER238
      ENDIF
      ENDIF
      ENDIF
C
      NC = ICMP(8)
      IF (NC .GT. 0) THEN
C
      IF (IREPRT .EQ. 0) THEN
      WRITE (NDEV3,9051) PCTNP,ERNP
9051  FORMAT (' Np237 =',20X,F9.4,F7.2)
      ELSE
      WRITE (NDEV3,9052) PCTNP,ERNP
9052  FORMAT (' Np237 =',35X,F9.4,F7.2)
      ENDIF
      ENDIF
C
      IF (IREPRT .EQ. 0) THEN
      WRITE (NDEV3,9050) POW,SGPOW
9050  FORMAT (25X,'* = Error in Ratio',13X,'TOTAL =',F6.3,' +/- ',
+ F4.2,'%')
      ELSE
      WRITE (NDEV3,9250)
9250  FORMAT (25X,'* = Error in Ratio')
      ENDIF
C
58    IF (A106 .LE. 0.0) GOTO 60
C
      !239Np not present
      IF (IPRINT .EQ. 'Y') WRITE (NDEV3,9053)
9053  FORMAT ('/ Am243-Np239 also found to be present.')
C
      AM243 = A106 * 8.965E-9 / SUM1
C
      !Calc. 243Am content
C
      WRITE (NDEV3,9053)AM243,ER106
C
      !Report
C9053  FORMAT (' AM243 (NP239)',14X,1PE11.2,0PF7.2)
C
60    IF (ICMP(10) .GT. 0) THEN
C
      !U is present
      IF (ERUPU .GT. 30. .OR. .01*PCTA/UPU .GT. 1.) GOTO 63
      IF (IPRINT .EQ. 'Y') WRITE (NDEV3,1013)
      WRITE (NDEV3,9056) UPU,ERUPU
9056  FORMAT (' Approx. U/Pu ratio by fluorescence x rays = ',
+ F5.2,' +/- ',F5.2,'%')
      ENDIF

```

```

C      xrayr = 100.*sqrt(dm(6))/ans(6)
C      write (ndev2,8880) ans(6),xrayr
C      if (iprint .eq. 'Y') write (ndev3,8880) ans(6),xrayr
C8880  format (' Pu x ray fluorescence intensity =',1PE11.3,
C      +      ' +/-',OPF5.1,'%')
63      IF (IPRINT .EQ. 'Y') WRITE (NDEV3,1013)
      WRITE (NDEV3,9055) RAM241,ERRAM
9055  FORMAT (' Am/Pu241 Wt. Ratio =',F7.5,' +/-',F5.2,'%')
      IF (ABS(PCTDIF) .GT. 0.) THEN
9057  WRITE (NDEV3,9057) -STDDEV,-PCTDIF
      FORMAT (' The 100-662 keV 241Am peak results differ by',
      +      F5.1, ' Std. Devs. or',F6.1,'%')
      KEV = 100
9058  WRITE (NDEV3,9058) KEV,AM100,DM100*PCT(5)/ER1(5)
      FORMAT (I4,' keV peaks yields a zero-time Wt. Pct. of',F7.4,
      +      ' +/-',F5.1,'%')
      KEV = 300
      WRITE (NDEV3,9058) KEV,AM300,DM300*PCT(5)/ER1(5)
      KEV = 600
      WRITE (NDEV3,9058) KEV,AM600,DM600*PCT(5)/ER1(5)
      WRITE (NDEV3,9059) AA(2)
9059  FORMAT (' Beta =',F6.3)
      ENDIF
C
      AVBG = AVEBG
      LL = NDPTS + 1
      JRM = NPT100+1
C
      IF (NOPFLG(1) .GT. 0) THEN
          JRM = JRM + 3
          RDPM = ANS(ICMP(7))/ANS(4)
          IF (RDPM .GT. .75) THEN
              WRITE (NDEV3,9060)
9060  FORMAT (' 237U Separated > 10 Days Ago.')
          ELSE
              TZ237 = -9.7383*ALOG(1.-RDPM)
              WRITE (NDEV3,9062) TZ237
9062  FORMAT (' 237U Separated About',F5.1,' Days Ago.')
          ENDIF
      ENDIF
      IF (IPRINT .EQ. 'Y') WRITE (NDEV3,1013)
      WRITE (NDEV3,9064) TZ,ERRTZ
9064  FORMAT (' 241Am Separated About',F7.3,' +/-',F5.3,' Years Ago.')
C
      IF (IPRINT .EQ. 'Y') WRITE (NDEV3,1013)
      WRITE (NDEV3,9065) PEFF,EPEFF
9065  FORMAT (' Pu-240 effective =',F7.3,' +/-',F6.2,'%')
C
      IF (NTYPE .GT. 0) THEN
          IF (IPRINT .EQ. 'Y') WRITE (NDEV3,1013)
          IF (NTYPE .EQ. 2) CONC = 1000.*CONC
          IF (CDFCT .GT. 0.) WRITE (NDEV3,9067) CONC*CDFCT,SIG
9067  FORMAT (' Pu Solution Concentration (Monitor Corrected) =',
      +      F7.3,' g/L +/-',F5.2,'%')
          WRITE (NDEV3,9068) CONC,SIG
9068  FORMAT (' Pu Solution Concentration (Time-Based Only) =',
      +      F7.3,' g/L +/-',F5.2,'%')
      ENDIF
C
      Print out fission product assays
C
      IF (IREPRT .EQ. 1 .AND. NUMDET .EQ. 2) THEN
          WRITE(NDEV3,9200)
9200  FORMAT (' Assay of Fission Product Contaminants/'
      +      ' Isotope Bq/gm Pu Uncertainty')
          FCT = 6.26E-5*2.29E9/(.11*AREA(6))

```

```

      CTSFIS(6) = CTSFIS(6) *FCT
      ERFIS(6) = ERFIS(6) *FCT
      DO 66 I = 1,6
      IF (CTSFIS(I) .LE. 0.0) CTSFIS(I) = ERFIS(I)
      CTSFIS(I) = CTSFIS(I)*PCT2(2)/100.
      ERFIS(I) = ERFIS(I)*PCT2(2)/100.
      UPRLMT = .3
      IF (I .EQ. 1 .OR. I .EQ. 2 .OR. I .EQ. 5)UPRLMT = .1
      IF (ERFIS(I)/CTSFIS(I) .GT. UPRLMT) THEN
9202  WRITE (NDEV3,9202) LABLE(I),3.*ERFIS(I)
      FORMAT (2X,A5,1PE11.2,' Upper Limit')
      ELSE
9204  WRITE (NDEV3,9204) LABLE(I),CTSFIS(I),100.*ERFIS(I)/CTSFIS(I)
      FORMAT (2X,A5,1PE11.2,' +/-',OPF5.1,'%')
      ENDIF
      CONTINUE
      ENDIF
C
      IF (ICMP(9) .GT. 0) JRM = JRM + 1
      IF (AREA(15) .GT. 0.0) JRM = JRM + 1
C
      IF (RMSD .GT. 1.05) THEN
          WRITE (NDEV3,9069)
9069  FORMAT (' The quality-of-fit is poor. Data will be printed.')
      ELSE
          !Check to see if long form is requested
          IF (NOPFLG(5) .EQ. 0) GOTO 82
          ENDIF
C
          R4139 = -ACOEFF(JRM,2)/ACOEFF(JRM,4)
          D4139 = (R4139/GRMS(4)-1.) *100.
          WRITE (NDEV3,9071) R4139,D4139
9071  FORMAT (' 241/239 ratio based on 148/129 keV peaks =',F9.6,
      +      ' Diff. =',F7.4,'%')
          IF (NOPFLG(5) .NE. 0 .AND. IPRINT .NE. 'Y')
          +      PAUSE 'Press "Enter" to continue listing of Fit'
          WRITE (NDEV3,9070)
9070  FORMAT (/3X,'keV NETCNTS RES keV NETCNTS RES keV ',
      +      ' NETCNTS RES keV NETCNTS RES')
          NPTS = (FLOAT (NDPTS)+3.) / 4.
          OFFST1 = 92.6
          INDX = 0
          IF (NOPFLG(1) .LE.0)INDX = 3
          XDEV = 0.0
C
          DO 78 K = 1,NDPTS
          IF (K .LE. NPT100) THEN
              ENRG(K) = K * GAIN + OFFST1
          ELSE
              INDX = INDX + 1
              IF ((NOPFLG(1) .GT. 0) .AND. (INDX .LT. 4)) GOTO 79
              IF (INDX .LT. 6) THEN
                  IF (ICMP(9) .GT. 0) THEN
                      IF (INDX .EQ. 4) GOTO 79
                      IF (AREA(15) .GT. 0.0) GOTO 79
                      INDX = 6
                  ELSE
                      INDX = 6
                  END IF
              END IF
          END IF
          IF ((INDX .EQ. 9) .AND. (ICMP(8) .EQ. 0)) INDX = 10
          IF (INDX .EQ. 17 .AND. R6669 .EQ. 0.0) INDX = 18
          ENRG(K) = PKEN(INDX)
          IF (INDX .EQ. 13 .AND. CTS370(7)/CTS370(6) .LE. .15)INDX=INDX+1
          ENDIF
      79

```

```
78      CONTINUE
C
      YNET(NDPTS+1) = 0.0
      YNET(NDPTS+2) = 0.0
      YNET(NDPTS+3) = 0.0
C
      DO 80 I = 1,NPTS
      NXT = 0
C
      DO 84 K = 1,4
      JK = I + NXT
      ENG(K) = ENRG(JK)
      YNT(K) = YNET(JK)
      RMO(K) = RM(JK)
      NXT = NXT + NPTS
84      CONTINUE

      IF (ICMP(8) .LE. 0 .OR. A86 .EQ. 0.) GOTO 80
      IF (I .EQ. 1) ENG(1) = 86.5
      IF (I .EQ. 2 .AND. SPC2ID(1:1) .EQ. ' ') ENG(1) = 300.1
C
80      WRITE (NDEV3,9072) (ENG(K),YNT(K),RMO(K),K=1,4)
9072     FORMAT (4(F6.1,F9.0,F5.1))
C
82      IF (IPRINT .EQ. 'Y') WRITE (NDEV3,9074)
9074     FORMAT ('1')
C
99      ICMP(8) = 0
      RETURN
      END
```

----->

```

SUBROUTINE MGASOL
C
C*****
C
C      This subroutine is called by the MGACAL subroutine of MGA
C      when the NTYPE argument is set for MGA to determine and
C      report the Pu concentration of a solution measurement.
C*****
C
C      First implemented in version 9.0 of MGA in 1/1992
C*****
C
C$INCLUDE:'MGACOM.CMN'
C$INCLUDE:'MGACL.CMN'
C
C      CHARACTER*10 SFILNM
C      CHARACTER*3 STFIL(2)
C      CHARACTER MY
C      DATA SFILNM/'MGASOL.'/
C      DATA STFIL/'OLD','NEW'/
C      DATA NDEV/1/
C
C-----
C
C      1      SFILNM(8:9) = DETSYS(1:2)
C             NERR = 0
C      2      OPEN (UNIT=NDEV,FILE=SFILNM,STATUS=STFIL(NTYPE),ERR=9000)
C             GOTO 10
C      9000   IF (NERR .EQ. 0) THEN
C             NERR = 1
C             ELSE
C             STOP ' Cannot open file in MGASOL'
C             ENDIF
C
C             See if old file already exists
C             NTYPE = 1
C             GOTO 2
C      10      IF (NERR .EQ. 1) THEN
C             WRITE (NDEV,9010)
C      9010   +      FORMAT (' Calibration file already exists. Write over it?',
C                          ' (Y/N): ', $)
C             READ (NDEV,9020) MY
C             FORMAT (A)
C             IF (MY .EQ. 'N' .OR. my .eq. 'n') THEN
C             WRITE (NDEV,9030)
C      9030   +      FORMAT (' Enter new Detector Label: ', $)
C             READ (NDEV,9020) DETSYS(1:2)
C             GOTO 1
C             ELSE
C             NTYPE = 2
C             ENDIF
C             ENDIF
C
C      IF (NTYPE .EQ. 1) THEN
C      C      Read in parameters from calibration file
C      READ (NDEV,1000) SURFC
C      1000  +      FORMAT (F12.7)
C      READ (NDEV,1002) CCNST
C      1002  +      FORMAT (E12.4)
C      READ (NDEV,1000) VOLUME
C      READ (NDEV,1000) XDIST
C      READ (NDEV,1000) SIGCAL
C      READ (NDEV,1002) CNST59
C      READ (NDEV,1002) CDCPM
C      READ (NDEV,1000) CDERR

```

```

      READ (NDEV,1002) CALDAT
      CDFCT = 0.0
      IF (CDCPM .GT. 0. .AND. CPM88 .GT. 0.) THEN
C      C      Decay 109Cd monitor source
C      CDCPM = CDCPM*EXP(.69315*(CALDAT-CDECAY)/462.6)
C      C      Change in efficiency
C      CDFCT = CDCPM/CPM88
C      ENDIF
C      CLOSE(NDEV)
C      ELSE
C      C      Calibration mode. Store parameters in file.
C      WRITE (NDEV,1012) SURFC
C      1012  +      FORMAT (F12.7, ' Sample Area')
C      C      Total grams Pu in sample.
C      WRITE (NDEV,1018) CCNST
C      1018  +      FORMAT (1PE12.4, ' Calibration Factor')
C      WRITE (NDEV,1014) VOLUME
C      1014  +      FORMAT (F12.4, ' Sample Volume = Area * Depth')
C      WRITE (NDEV,1016) XDIST
C      1016  +      FORMAT (F12.7, ' Source-to-Detector Distance')
C      WRITE (NDEV,1019) SIG
C      1019  +      FORMAT (F12.7, ' Error on Calibration Factor')
C      WRITE (NDEV,1020) CNST59
C      1020  +      FORMAT (1PE12.4, ' 59 keV Calibration Factor')
C      IF (CPM88 .LT. 300.) CPM88 = 0.0
C      WRITE (NDEV,1022) CPM88
C      1022  +      FORMAT (1PE12.4, ' 88 keV cpm at Calibration Date')
C      IF (CPM88 .EQ. 0.0) ERR88 = 0.0
C      WRITE (NDEV,1024) ERR88
C      1024  +      FORMAT (1PE12.4, ' 88 keV Uncertainty(%)')
C      WRITE (NDEV,1026) CDECAY
C      1026  +      FORMAT (1PE12.4, ' Calibration Date (days since 1990)')
C      CLOSE(NDEV)
C      SFIL(7:8) = DETSYS(1:2)
C      OPEN (UNIT=NDEV,FILE=SFIL)
C      SHAPC(6) = 0.0
C      SHAPC(12) = ESLP
C      SHAPC(13) = ECURV
C      SHAPC(15) = CDABS
C      WRITE (NDEV,1040) (SHAPC(I),I=1,15)
C      1040  +      FORMAT (1PE15.6)
C      CLOSE(NDEV)
C      ENDIF
C
C      RETURN
C      END

```


SUBROUTINE SETUP

```

C*****
C
C      This subroutine of the MGA program is used to initialize
C      or establish setup parameters for the subsequent runing
C      of the MGA program.
C
C      First introduced with version 9.0 of MGA in 1/1992
C*****
C$INCLUDE: 'MGACOM.CMN'
C$INCLUDE: 'MGACL.CMN'
C
C      CHARACTER*3 STFIL(2)
C      DATA STFIL/'OLD','NEW'/
C      DATA NDEV/1/
C
C-----
C      The following are the default values of the setup parameters
C
C      IPRINT = 'Y' means output to line printer
C      IPRINT = 'N'
C      NOOUT = 0 output to files; ie. no printout.
C      NOOUT = 1
C      NTYPE = 0 NTYPE = Type of analysis to be performed.
C      NOPFLG(3) = 0 NOPFLG(3) = Spectrum format index
C      NUMDET = 1 NUMDET = Number of detectors (1 or 2)
C      SPC2ID(1:1) = ' ' NCHANS = Number of channels in the 1st spectrum
C      NCHANS = 4096
C      Default GAIN, ZERO, and Shaping time constants for the detectors
C      GAIN = .075
C      ZERO = 0.0
C      SHPTM1 = 2.0
C      GAIN2 = .25
C      ZERO2 = 0.0
C      SHPTM2 = 2.0
C      Detector-System label
C      DETSYS = ' '
C      JS = 0
C      JS = JS + 1
C      IF (JS .GT. 2) STOP 'Cannot open SETUP.MGA file'
C
C      OPEN (UNIT=NDEV,FILE='SETUP.MGA',STATUS=STFIL(JS),ERR=10)
C
C      IF (JS .EQ. 1) THEN
C      READ (NDEV,1001) NOOUT
C      FORMAT (I4)
C      READ (NDEV,1001) IPR
C      IPRINT = 'Y'
C      IF (IPR .EQ. 0) IPRINT = 'N'
C      READ (NDEV,1001) NTYPE
C      READ (NDEV,1001) NOPFLG(3)
C      READ (NDEV,1001) NCHANS
C      READ (NDEV,1001) NUMDET
C      IF (NUMDET .EQ. 2) SPC2ID(1:1) = '2'
C      READ (NDEV,1006) GAIN
C      FORMAT (F10.7)
C      READ (NDEV,1006) ZERO
C      READ (NDEV,1010) SHPTM1

```

```

1010  FORMAT (F4.1)
      READ (NDEV,1006) GAIN2
      READ (NDEV,1006) ZERO2
      READ (NDEV,1010) SHPTM2
      READ (NDEV,1018) DETSYS
1018  FORMAT (A2)
      READ (NDEV,1001) ICMPTR
      READ (NDEV,1001) IREPRT
      ELSE
      WRITE (NDEV2,1040)
1040  FORMAT (' Output to a file(0) or to screen/printer(1)? ',%)
      READ (NDEV2,1041) NOOUT
1041  FORMAT (I4)
      IF (NOOUT .EQ. 1) THEN
      WRITE (NDEV2,1042)
1042  FORMAT
      + (' Normal output to screen (=0) or line printer(=1): ',%)
      READ (NDEV2,1041) IPR
      ENDIF
      IPRINT = 'Y'
      IF (IPR .EQ. 0) IPRINT = 'N'
      WRITE (NDEV2,1044)
1044  FORMAT (' 3=ASCII, 4=IAEA binary, 5=ND uMCA, 6=IAEA "PUIS"',
      + ' 7=S100, 8=Ortec Spectran!/'
      + ' Spectrum format index = ',%)
      READ (NDEV1,1041) NOPFLG(3)
      WRITE (NDEV2,1046)
1046  FORMAT (' Number of channels in 1st spectrum = ',%)
      READ (NDEV1,1041) NCHANS
      IF (NCHANS .LT. 4097) THEN
      WRITE (NDEV2,1048)
1048  FORMAT (' One (1) or two (2) spectrum mode? ',%)
      READ (NDEV1,1041) NUMDET
      IF (NUMDET .EQ. 2) SPC2ID(1:1) = '1'
      ENDIF
      WRITE (NDEV2,1050)
1050  FORMAT (' Computer System: PC(=0) or VAX(=1): ',%)
      READ (NDEV1,1041) ICMPTR
      WRITE (NDEV2,1052)
1052  FORMAT (' Standard Output Format (=0)/241Pu ratio format(=1)')
      READ (NDEV1,1041) IREPRT
C
      WRITE (NDEV,1020) NOOUT
      WRITE (NDEV,1021) IPR
      WRITE (NDEV,1022) NTYPE
      WRITE (NDEV,1024) NOPFLG(3)
      WRITE (NDEV,1025) NCHANS
      IF (NCHANS .GT. 4096) NUMDET = 1
      WRITE (NDEV,1026) NUMDET
      WRITE (NDEV,1027) GAIN
      WRITE (NDEV,1028) ZERO
      WRITE (NDEV,1030) SHPTM1
      WRITE (NDEV,1032) GAIN2
      WRITE (NDEV,1034) ZERO2
      WRITE (NDEV,1036) SHPTM2
      WRITE (NDEV,1038) DETSYS
      WRITE (NDEV,1060) ICMPTR
      WRITE (NDEV,1062) IREPRT
1020  FORMAT (I4,' Value <or= 0= output to file; no other output')
1021  FORMAT (I4,' Output to screen = 0, to line printer = 1')
1022  FORMAT (I4,' Type of analysis, 0 = std, 1 = solution')
1024  FORMAT (I4,' Spectrum format index')
1025  FORMAT (I4,' Number of channels in 1st spectrum')
1026  FORMAT (I4,' One or two spectra')
1027  FORMAT (F10.7,' Gain of 1st spectrum')
1028  FORMAT (F10.7,' Zero energy channel position of 1st spectrum')

```

```
1030  FORMAT (F4.1,'      Shaping time-constant for 1st system')
1032  FORMAT (F10.7,' Gain of 2nd spectrum')
1034  FORMAT (F10.7,' Zero energy channel positon of 2nd spectrum')
1036  FORMAT (F4.1,'      Shaping time-constant for 2nd system')
1038  FORMAT (A2,'      Detector-system label')
1060  FORMAT (I4,'      Computer System: PC(=0) or VAX(=1)')
1062  +  FORMAT (I4,'      Standard Printout (=0)/241Pu ratio printout(=1)')
      ENDIF
C
      CLOSE(NDEV)
C
      RETURN
      END
```


Common blocks 'MGACOM', 'MGADAT', and 'MGACHR':

Definition of variables found in COMMON block:

NDPTS	Number of equations representing the spectral regions that are used in the analysis.
NCMP	The number of unknowns or components being determined by the least-squares fitting.
IOCH1	I/O channel for 1st spectrum
IOCH2	I/O channel for 2nd spectrum (Not used in this version)
IDT1	System #1 dead time
SPC11D	Spectrum ID of the 1st (low energy) spectrum.
SPC21D	Spectrum ID of the 2nd (high energy) spectrum (Not used)
ICMP	A flag array indicating which components are present. Component identities are defined in root program (MGA).
NOPFLG1	A flag indicating that the sample is freshly processed or that the 237U is out of equilibrium with its 241Pu parent. A negative value signifies that the 38-59 keV region will NOT be included in the analysis.
NOPFLG2	A flag indicating the approximate intensity of 235U.
NOPFLG3	Lab. spectrum index code - to determine spectrum format
NOPFLG4	Used to fix efficiency curvature
NOPFLG5	A option flag indicating short or long printout.
NOPFLG6	Calibration mode for analyzing solution samples
NOPFLG7	Number of channels included from 100 keV group
NOPFLG8	Flag to indicate repeat of analysis
NOPFLG9	= 1 will fix Cd absorber thickness (CDABS) at input value.
NOPFLG10	= 1 will fix Pu thickness (PUGPSC) at input value.
SMPID	ASCII description of the sample.
SHAPC	Array of peak shape constants for small detector
SFIL	Name of file containing small-detector shape constants
SHAPC2	Array of peak shape constants for large detector
SFIL2	Large detector shape constants file name
ZERO	Channel where energy = 0 keV for first spectrum
GAIN	System gain in keV/channel for 1st (small) detector
ZERO2	Channel where energy = 0 keV for second spectrum
GAIN2	System gain (2nd spectrum)
SHPTM1	Shaping time constant for 1st system
SHPTM2	Shaping time constant for 2nd system
TIME	Counting live time of 1st spectrum (in minutes)
TIM	Counting live time of 2nd spectrum (in minutes)
DLTAT	Time (in days) between count date and zero date.
COLTM	Reactor discharge-to-chem. sep. cooling time. Needed only if 242Pu abundance is to be calculate using algorithm
ALFA	Peak width variable (Related to FWHM)
EXP1	Short-term tailing amplitude variable
EXP2	Short-term tailing slope variable
GSQ	Gain squared
SG	Equals the FWHM **2
ST	Starting channel of 100 Kev region
AVEBG	Average background continuum level in the 100 keV region
AVBG2	Average background continuum level in the 332-345 keV region
AVBG3	Average background continuum level in the 367-382 keV region
PU242	User input of 242Pu abundance
A86	Area of 86 keV peak due to 237Np (if present)
ER86	Uncertainty in 86 keV peak area
HT61	Height of 61 keV peak due to 239Np (if present)
A106	Area of 106 keV peak of 239Np (if present)
ER106	Uncertainty in 106 keV peak area.
CDABS	The amount of Cd absorber present. (g/cm2)
PUGPSC	The approximate Pu sample thickness. (g/cm2) The sample is assumed infinitely thick at 5 g/cm2.
RSLP	Remaining slope in 100 keV region based on 94/111 keV residuals
ESLP	The change in the detector efficiency slope from that of the generic efficiency slope written into the code.
ECURV	The change of the detector efficiency curvature from that of

[illegible]

SUBROUTINE AVE (NS,NE,SUM,AV,Y)

```

C*****
C
C
C      AVE
C      ====
C*****

```

Developed for : Safeguards Technology Program
 Nuclear Chemistry Division
 Lawrence Livermore National Laboratory
 Livermore, CA. 94550

```

C*****
C
C      Title      : AVE
C      Revision   :
C      Author     : R. Gunnink
C      : Revised for the IBM PC by Russell Fleming
C      Date       : 10/15/86
C      Language   : Microsoft Fortran Ver. 4.01
C*****

```

Abstract :
 This routine determines the average count of a group of data values from NS to NE inclusive. It also returns the sum.

```

C*****
C
C      Arguments :
C      INPUT
C      NS      Starting position for group in Y array
C      NE      Ending position for group in Y array
C      Y       Data array
C
C      OUTPUT
C      AV      Average count of data points in group set by NS and NE
C      SUM     Sum of data points in group
C*****

```

Routines
 Called : None

Libraries
 Used : None

```

C*****
C      Revision History ( Include date,revision I.D., and details ) *
C*****
C      Original Version: 19-Jan-1978, Ver. 2.1
C*****

```

```

C*****
C      COMMON and variable declarations
C*****
C
C      DIMENSION Y(*)
C*****

```

```

C      variable initialization declarations *
C*****
C
C      SUM = 0.0
C*****
C
C      SM = NE - NS + 1
C      DO 2 I = NS,NE
C      SUM = SUM + Y(I)
C      AV = SUM / SM
C      RETURN
C      END

```

SUBROUTINE BKGD (NS,NE,AV1,AV2,S2,ND,Y1,Y2)

```

C*****
C
C          BKGD
C          =====
C*****
C
C    Developed for : Safeguards Technology Program
C                  Nuclear Chemistry Division
C                  Lawrence Livermore National Laboratory
C                  Livermore, CA. 94550
C*****
C
C    Title       : BKGD
C    Revision    : 2.1
C    Author      : Ray Gunnink
C    Date        : 1/28/88
C    Language    : Microsoft Fortran Ver 4.01
C*****
C
C    Abstract :
C    To calculate the background continuum under a peak region,
C    to subtract it from the channel counts, and return the net
C    channel and total counts.
C
C    Like NCTS, this subroutine calculates the background distribution
C    for a specified peak region, performs a channel-by-channel
C    subtraction, and returns the net area and net count array.
C    However, the input arguments provide the before and aft average
C    background levels to be used. The routine uses one pass for a
C    positive loping background, and two for a negative slope. The first
C    pass determines a straight line connecting the before and aft
C    background levels. The second pass additionally computes a
C    smoothed-step background which is subtracted from the gross channel
C    counts to obtain the net count array and the integrated net count.
C
C    This routine is similar to the NCTS and BKGRD routines. The
C    BKGRD is more rigorous and also takes into account the entrance and
C    exit slopes of the background.
C*****
C
C    Arguments :
C    INPUT
C    NS = Beginning channel of peak grouping
C    NE = Ending channel of peak grouping
C    AV1= Average background level in front of the peak grouping.
C    AV2= Average background level aft of the peak grouping.
C    Y1 = Array containing gross counts of the peak grouping.
C
C    RETURN
C    ND = Number of channels in peak grouping
C    S2 = Net counts in the peak region.
C    Y2 = Array of net counts generated by the routine.
C
C    OTHER VARIABLES
C    DBG = Change in background level.
C    CHANS = Number of channels in the peak grouping.
C    SLOPE = Average change in background per channel.

```

```

C    AVSIG = A factor related to the statistical variance of the data
C            in the region. It is used to provide smoothing to
C            statistically poor data.
C    SUMY = Variable used to accumulate the net count using
C            the straight line for background determination.
C    SUM = The final value of SUMY following the 1st pass.
C    YN = Net count of a channel (temporary argument).
C    YNT= Positive value only of YN (temporary argument).
C    SLBG = Straight line background value at a given channel.
C*****
C
C    Routines
C    Called      : None
C
C    Libraries
C    Used        : None
C*****
C
C    Revision History ( Include date, revision I.D., and details ) *
C*****
C    Version 2: 12-Oct-84
C*****
C    COMMON and variable declarations *
C*****
C
C    dimension Y1(*),Y2(*)
C*****
C    variable initialization declarations *
C*****
C
C    Change in background level.
C    DBG = AV2 - AV1
C    ND = 0
C    S2 = 0.0
C
C    Total channels in group.
C    CHANS = NE - NS + 1
C
C    Average change in background per channel.
C    SLOPE = DBG / CHANS
C    IF (AV2.LE.0)AV2=1.
C
C    Smoothing factor
C    AVSIG = 5./CHANS*SQRT(AV2*CHANS)
C    NN = 2
C
C    One pass for positive sloping bkgd.
C    IF (SLOPE.GT. 0.0) NN = 1
C
C    DO 30 J = 1,NN
C    SUMY = 0.0
C
C    Initial straight line bkgd value set at AV1.
C    SLBG = AV1
C
C    DO 20 I = NS,NE
C    New straight line bkgd value.
C    SLBG = SLBG + SLOPE
C
C    Calc. net count.
C    YN = Y1(I) - SLBG
C    YNT = YN
C
C    Only integrate positive values.
C    IF (YN.LT. 0.0) YNT = 0.0
C
C    Running sum of net counts.
C    SUMY = SUMY + YNT + AVSIG

```



```
C  Straight line only on + slope
      IF (SLOPE .GT. 0.0) GOTO 10
C  Straight line only on 1st pass
      IF (J .EQ. 1) GOTO 20
C  Calc. smoothed-step bkgrd value.
      YN = Y1(I) -AV1 -DBG*SUMY/SUM

10      ND = ND + 1
C  Store net channel count.
      Y2(ND) = YN
C  Integrate net channel counts.
      S2 = S2 + YN
20      CONTINUE

C  Store total net counts using straight line bkgrd.
      SUM = SUMY
C  Get out if SUM is zero or less
      IF (SUM .LE. 0.)RETURN
30      CONTINUE

      RETURN
      END
```


SUBROUTINE BKGRD (NS,NE,BGLO,BGHI,SLPLO,SLPHI,NPTS,SMY,Y,YNET)

```

C *****
C                                     *
C                                     *
C                                     *
C                                     *
C                                     *
C *****

```

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

```

C *****
C Title       : BKGRD
C Revision    :
C Author      : Ray Gunnink
C              Revised for the IBM PC by Russell Fleming
C Revised     : Set SMY = 0 if SMY < 0.
C Date        : 10/15/86
C Language    : Microsoft Fortran Ver. 4.01
C *****

```

Abstract :

This subroutine calculates the background distribution for a given peak region, subtracts it, and returns the net count array. Like NCTS and BKGD, this routine generates an accurate step-shaped background. However, in these routines, BKGRD takes into consideration the sloping backgrounds surrounding the region of interest. If the entrance and exit slopes differ, the routine assumes that the change from one slope to the other is monotonic.

Arguments :

INPUT

NS = starting channel of the region
NE = ending channel of the region.
BGLO = background count in channel next to low E side of peak region.
BGHI = background level next to high energy side of region.
SLPLO = background slope on low side (in counts/channel)
SLPHI = background slope on high side (in counts/channel)
Y(I) = array containing original spectral data.

RETURN

NPTS = number of channels in YNET array.
SMY = sum of net counts returned to calling program
YNET(I) = array into which net counts are placed (may be same as Y).

OTHER VARIABLES

DBG = Change in background level.
SIG & AVSIG = Factors related to the statistical variance of the data in the region. They are used to provide a measure of smoothing to statistically poor data.
SLPBG = Contribution to the bkgrd due to the slope of the bkgrd.
SLBG = Straight line background value at a given channel.
YN = Net count of a channel.
YNT = Positive value only of YN.
SUMY1 = Integrated net area as obtained using a straight line bkgrd.

```

C SUMY2 = Integrated net area as obtained using a step-function bkgrd.
C YSUM1 = Variable used to accumulate straight-line net area.
C YSUM2 = Variable used to accumulate step-function net area.
C YS1,YS2,YS3,YS4 = Accumulator variables used to account for histogram
C                   nature of the data. ie use halves of 2 adjacent channels.
C *****

```

Routines

Called : None

Libraries

Used : None

Revision History (Include date,revision I.D., and details) *

```

C Version: 1b
C Author: Ray Gunnink
C Date: 12-Dec-83
C Last revision: 20-Sep-84 (R.G.)
C *****

```

COMMON and variable declarations *

DIMENSION Y(*),YNET(*)

variable initialization declarations *

```

C *****
C Number of channels in grouping.
C   NPTS = NE-NS+1
C   XPTS = NPTS
C Level change in bkgrd
C   DBG = (BGHI-BGLO) -0.5 *(SLPLO+SLPHI)*(XPTS+1.)
C Average bkgrd slope in counts/channel
C   SLOPE = DBG / (XPTS + 1.)
C Starting (low-side) bkgrd level.
C   SLBG = BGLO
C   SUMY1 = 0.0
C   YSUM2 = 0.0
C   IF (BGLO.LE.0.) BGLO = 1.
C Variable related to statistical level of bkgrd
C   SIG = 5.*SQRT(BGLO)
C   AVSIG = SIG/XPTS
C
C   L = 0
C   SLPBG = 0.0

```

C Get 1st estimate of net area of peak(s).

DO 10 I = NS,NE

C Fractional distance through grouping.

FRAC = L / XPTS

C Amt due to sloping bkgrd.

SLPBG = SLPBG+SLPLO*(1.-FRAC)+SLPHI*FRAC

C Straight line background level.

SLBG = SLBG + SLOPE

L = L + 1

C Net count in channel using straight line

YN = Y(I) -SLBG -SLPBG

C Only accumulate positive values.

```

      IF(YN .LT. 0.) YN = 0.
C Approximate area using straight line.
10      SUMY1 = SUMY1 + YN
C Add value related to statistical variance.
      SUMY1 = SUMY1 + SIG
      NN = 2
C Only 1 pass if slope is positive.
      IF (DBG .GE. -10.0) NN = 1

C 1st uses straight line. 2nd also uses step.
      DO 30 J = 1,NN
      L = 0
C Start at initial bkgd level.
      SLBG = BGLO
      YSUM1 = 0.0
      SLPBG = 0.0
C Data is histogram. Use half of 2 adjacent channels.
      YS1 = 0.5 * BGLO
      YS3 = YS1
      SMY = 0.0

      DO 20 I = NS,NE
C Straight line background.
      SLBG = SLBG + SLOPE
C Fractional distance through grouping
      FRAC = L / XPTS
C Background slope.
      SLPBG = SLPBG + SLPLO *(1.-FRAC)+SLPHI *FRAC
C Net count using straight line.
      YN = Y(I) - SLBG - SLPBG
      YNT = YN
C Only integrate positive values.
      IF (YN .LT. 0.) YNT = 0.
      L = L + 1
C Use straight line only for + step.
      IF (DBG .GE. -10.) THEN
        IF (J .EQ. 1 .AND. NN .EQ. 2) GOTO 15
        GOTO 14
      ENDIF
C Partial area up to channel "I".
      YSUM1 = YSUM1 + YNT + AVSIG
C Other half.
      YS2 = 0.5 * (BGLO + DBG * YSUM1/SUMY1)
C Net count using straight line bgrd.
      YN = Y(I) - YS1 - YS2 - SLPBG
C Skip following on first iteration.
      IF (J .EQ. 1) GOTO 15

C Partial area to channel "I" using step bgrd.
      YSUM2 = YSUM2 + YN + AVSIG
C Other half.
      YS4 = 0.5 * (BGLO + DBG * YSUM2/SUMY2)
C Net count resulting from refined step.
      YN = Y(I) - YS3 - YS4 - SLPBG
      YS3 = YS4
C Transfer to YNET array on last iteration.
14      YNET(L) = YN
C Net counts (or area).
15      SMY = SMY + YN

      YS1 = YS2
      CONTINUE

20
C New value of SUMY2 for 2nd pass.
      IF (SMY .LT. 0.)SMY = 0.
      SUMY2 = SMY + SIG

```

```

30      CONTINUE

      RETURN
      END

```

SUBROUTINE BWF (X)

```

C*****
C
C          BWF
C          =====
C*****
C
C    Developed for : Safeguards Technology Program
C                   Nuclear Chemistry Division
C                   Lawrence Livermore National Laboratory
C                   Livermore, CA. 94550
C*****
C
C    Title       : BWF
C    Revision    : 1.0
C    Author      : Ray Gunnink
C                 Revised for the IBM PC by Russell Fleming
C    Date        : 10/15/86
C    Language    : Microsoft Fortran Ver. 4.01
C*****
C
C    Abstract :
C*****
C
C    Arguments : X
C*****
C
C    Routines
C    Called    : None
C
C    Libraries
C    Used      : None
C*****
C
C    Revision History ( Include date,revision I.D., and details ) *
C*****
C
C    COMMON and variable declarations
C*****
C
C    DIMENSION CA(5),CB(6)
C*****
C
C    variable initialization declarations
C*****
C
C    CA(1) = .729281
C    CA(2) = .0344557
C    CA(3) = .127639
C    CA(4) = -.0325239
C    CA(5) = .00642257
C    CB(1) = .933301
C    CB(2) = -4.45603
C    CB(3) = 1.16191

```

```

CB(4) = -.141124
CB(5) = -.0352464
CB(6) = .00744387

```

```

C*****
C
C    IF (X - 2.2) 2,10,10
C    2    XX = X * X
C         AX = 1.
C         SUM = 0.0
C         DO 4 I = 1,5
C             AX = AX * XX
C             SUM = SUM + CA(I) * AX
C         X = SUM * EXP (-XX)
C    4
C         GO TO 20
C    10    XLOG = LOG (X)
C         SUM = CB(1)
C         DO 12 I = 2,6
C             SUM = SUM + CB(I) * XLOG ** (I-1)
C    12    X = EXP (SUM)
C    20    RETURN
C         END

```

SUBROUTINE BWF2 (WL,X,EXPNT,EXPN1)

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

BWF2

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

Version: 1

Author: Ray Gunnink

Date: 29-OCT-82

```

*****

```

PURPOSE

To evaluate certain series involved in the calculation of X ray peak profiles.

Reference: Ray Gunnink, "An Algorithm for Fitting Lorentzian-Broadened K-Series X-Ray Peaks of the Heavy Elements", Nucl. Inst. & Methods, 143, p145, (1977)

```

*      *      *      *      *      *      *      *

```

ARGUMENTS

INPUT

WL Variable related to the Lorentzian line width.

EXPNT =ALPHA*DLXSQ

EXPN1 =EXP(EXPNT)

RETURN

X ANSWER

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

DIMENSION CA(5),CB(6)

DATA CA/.729281,.0344557,.127639, -.0325239,.00642257/

DATA CB/.933301,-4.45603,1.16191,-.141124,-.0352464,.00744387/

C1 = 1. - .5642*WL

C2 = .15915 * WL

C3 = .25 * WL**2

C4 = .6366 * WL

RT = C1 + C3

X = ABS (X)

IF (X .GE. 2.2) GOTO 10

XX = X * X

AX = 1.

SUM = 0.0

DO 4 I = 1,5

AX = AX * XX

SUM = SUM + CA(I) * AX

CONTINUE

X = SUM * EXP (-XX)

GO TO 20

XLOG = LOG (X)

SUM = CB(1)

DO 12 I = 2,6

SUM = SUM + CB(I) * XLOG ** (I-1)

```

20      X = EXP (SUM)
      AA = (C1 - C2*EXPNT + C3*(1. +2.*EXPNT))*EXPN1 +C4*X
      X = AA / RT

```

RETURN

END

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE C2JDAT(MONTH,NDAY,NYR,DDAY)

```

```

C
C      This routine converts calendar month and day to Julian date.

```

```

C      MONTH = Calendar month
C      NDAY  = Calendar day
C      NYR   = Year
C      DDAY  = Julian day of year

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

C      REAL*8 DDAY
C      DIMENSION NDAYS(12)
C      DATA NDAYS/31,28,31,30,31,30,31,31,30,31,30,31/

```

```

C      DDAY = NDAY
C      NMONTH = MONTH - 1
C      IF(NMONTH.EQ.0)GOTO 20
C      DO 10 I = 1,NMONTH
C      DDAY = DDAY + NDAYS(I)
C      IF(MONTH.LT.3)RETURN
C      IF(MOD(NYR,4).EQ.0)DDAY = DDAY + 1.
C      RETURN
C      END

```

```

---->

```

```

C      FUNCTION ER
C      FUNCTION CALERR(XY,ALPHA,NTERMS,CHISQR)
C      DOUBLE PRECISION ALPHA(3,3)

```

```

C      YR=0.
C      DO 36 I=1,NTERMS
C      F1 = 1.
C      IF(I.EQ.1)GOTO 30
C      F1=XY**(I-1)
C      DO 36 J=1,NTERMS
C      F2 = 1.
C      IF(J.EQ.1)GOTO 34
C      F2=XY**(J-1)
C      YR=YR+ALPHA(I,J)*F1*F2
C      CONTINUE
C      CALERR=SQRT(CHISQR*ABS(YR))
C      RETURN
C      END

```

```

C
C      SUBROUTINE DATARD (NDEV,NTYPE,NS,NE,TIME,RLTIME,DATA,NYR,
C      +      MONTH,NDAY,NHOUR,MINUTE,sampid)
C
C-----
C      This is a subroutine for reading spectrum data of various formats
C      from files that have been previously opened. The routine will
C      also read date and time information if it is available in the
C      header or in the spectrum AND IF the argument "MONTH" has a value
C      of 0 entering the routine.
C
C      A spectrum format index, NTYPE, is used to select the file format.
C      The following are currently available in this routine.
C
C      1 = LLNL spectrum format
C      2 = Not available (Was used for old Canberra formats)
C      3 = ASCII format (Sequential file of integer counts)
C      4 = IAEA (Silen analyzer) binary format
C      5 = Nuclear Data uMCA format.
C      6 = Not available
C      7 = Canberra S100 PC format
C      8 = ORTEC spectrum master format
C      9 = Not available
C
C-----
C      CHARACTER*8 STRDTE
C      CHARACTER*1 STRTIME(4),STRSEC(2)
C      CHARACTER*3 CMONTH(12)
C      CHARACTER*512 HDRBYT
C      CHARACTER*26 SAMPID
C      INTEGER*2 TYPE,MCA,STRCH,LNGTDT,SEG,IOFF,JBUF(256)
C      INTEGER*4 IY,RTIME,LVTIM,ISTIM,IBUF(256)
C      EQUIVALENCE (JBUF,HDRBYT)
C
C      REAL*8 DSTIM
C      DIMENSION DATA(*)
C      DATA CMONTH/'Jan','Feb','Mar','Apr','May','Jun','Jul','Aug',
C      +      'Sep','Oct','Nov','Dec'/
C
C-----
C      GOTO (1,2,3,4,5,2,7,8,2) NTYPE
C
C      1111111111111111The following is for LLNL spectrum format 111111111111
C
C      CALL RDCHNS (DATA,NS,NE,NDEV)
C
C      IREC = 1
C      CALL RDBLK(256,JBUF,IREC,NDEV)
C      !Get time parameters
C      CALL GETIME (JBUF,TIME,RLTIME,DSTIM,NYR)
C      DAY = DSTIM
C
C      Get sample ID information from spectrum header
C
C      SAMPID(1:26)=HDRBYT(271:296)
C
C      GOTO 90
C
C      111111111111111111111111111111111111111111111111111111111111111111111111
C
C      write (*,9001) ntype
C      format(' Data format type',i2,' not implemented')
C      NTYPE = 0
C      RETURN

```

```

C
C      333333333333333333333333333333333333333333333333333333333333333333333333
C
C      NFORMT = 0
C      L = 0
C      DO 32 I = 1,NE
C      IF (I .GE. NS) THEN
C      L = L + 1
C      IF (NFORMT .EQ. 0) THEN
C      9005 READ (NDEV,9005,err=34) IY
C      FORMAT (I15)
C      DATA(L) = IY
C      ELSE
C      9006 READ (NDEV,9006,err=36) DATA(L)
C      FORMAT (E15.4)
C      ENDIF
C      ENDIF
C      CONTINUE
C      TIME = DATA(1)
C      RLTIME = DATA(2)
C      Start time (seconds), converted to fraction of day
C      FCTDAY = DATA(3)/86400.
C      Start date since 1900, converted to since 1970.
C      IF (DATA(4) .GT. 0.) THEN
C      ISTIM = DATA(4) - 25568. + FCTDAY
C      Start date since 1970 converted to seconds
C      ISTIM = ISTIM * 86400.
C      CALL SEC2J (ISTIM,NYR,DAY)
C      CALL J2CDAT (DAY,NYR,MONTH,NDAY,NHOUR,MINUTE)
C      ENDIF
C      RETURN
C
C      34 NFORMT = 1
C      REWIND NDEV
C      GOTO 30
C      36 WRITE (*,*) ' Cannot read this ASCII file.'
C      NTYPE = 0
C      RETURN
C
C      333333333333333333333333333333333333333333333333333333333333333333333333
C
C      4444444444444444 IAEA BINARY DATA FORMAT 444444444444444444444444444444444444
C
C      4 NUMCHS = NE-NS+1
C      L = 0
C      DO 42 I = 1,NE
C      READ(NDEV) IY
C      IF (I .GE. NS) THEN
C      L = L+1
C      DATA(L) = DBLE(IY)
C      ENDIF
C      42 CONTINUE
C      RETURN
C
C      5555555555555555 Nuclear Data uMCA FORMAT 555555555555555555555555555555555555
C
C      5 CALL RDHDR(IBUF,NDEV)
C      CALL GETND(IBUF,IOFF,TIME,RLTIME,DSTIM,NYR)
C      NUMCHS = NE-NS+1
C      CALL RNDAT(DATA,NS,NUMCHS,IOFF,NDEV)
C      DAY = DSTIM
C      GOTO 90
C
C      555555555555555555555555555555555555555555555555555555555555555555555555
C
C      7777777777777777 !Canberra S100 PC format 777777777777777777777777777777777777

```

[illegible]

```

C      SUBROUTINE DATE (DUM)
C
C      This is a dummy routine in the PC environment
C      character*9 dum
C
C      dum(1:9) = '          '
C      return
C      end

```


CC
 C FLTBF converts DOUBLE PRECISION binary spectral data to REAL*4
 C Conversion may be into the same buffer area. NWDS is word count.
 C (9-SEP-81) Force treatment of 24-bit data as always positive.

SUBROUTINE FLTBF(JBUF,FBUF,NWDS)

INTEGER*2 BUFPT,NWDS
 INTEGER*4 JBUF(1)
 REAL*4 FBUF(1)

DO 20, K=1,NWDS/2
 FBUF(K) = JBUF(K)
 IF(FBUF(K).LT.0)FBUF(K)=FBUF(K)+16777216.
 CONTINUE
 RETURN
 END

20

SUBROUTINE GETCB(IBUF,LIVTIM,RLTIME,ISTIM)

Subroutine that determines time information from header
 information stored in the first 128 byte record of a Canberra
 S-100 spectral data file.

INTEGER*2 ITC(4)
 INTEGER*4 ibuf(32),ISTIM,IDUM(2),istmp

REAL*4 LIVTIM
 REAL*4 RLTIME

EQUIVALENCE (IDUM,ITC(1)),(ISTMP,ITC(2))

LIVTIM = IBUF(13)/100.
 RLTIME = IBUF(14)/100.
 IDUM(1)= ibuf(10)
 IDUM(2)= ibuf(11)
 istim = istmp

RETURN
 END

```

C      SUBROUTINE GETND(IBUF,IOFF,LIVTIM,RLTIME,DSTTIM,ISTYR)
C
C      Subroutine that determines time information from header
C      information stored in the first 256 records of a Nuclear
C      Data "ASAP" spectral data file.
C
C      INTEGER*2 ISTR, IDAY, IT(2), IOFF
C      INTEGER*4 IBUF(256), LTIM, CTIM, ISTIM, IDUM
C      REAL*8 DSTTIM
C      REAL*4 LIVTIM
C
C      EQUIVALENCE (IT(1), IDUM)
C
C      LTIM = IBUF(13)
C      CTIM = IBUF(15)
C      idum = ibuf(7)
C      IOFF = it(2)/4
C      IDUM = IBUF(21)
C      IDAY = IT(2)
C      ISTR = IT(1)
C      ISTIM = IBUF(22)
C      LIVTIM = FLOAT(LTIM)
C      RLTIME = FLOAT(CTIM)
C      DSTTIM = IDAY + FLOAT(ISTIM)/86400.
C
C      RETURN
C      END

```

```

      SUBROUTINE GETIME(IDBUF,LIVTIM,RLTIME,DSTTIM,ISTYR)
C
C      SUB FOR PICKING LIVE, REAL, & START TIMES OUT OF STANDARD
C      LLL FILE HEADER.
C
C      REAL*4 LIVTIM, RLTIME, DBUF(10)
C      REAL*8 DSTTIM
C      INTEGER*2 IDBUF(1), IDECBF(20)
C
C      EQUIVALENCE (DBUF, IDECBF)
C
C      ISTR = IDBUF(83)
C      IDECBF(1) = IDBUF(85)
C      IDECBF(2) = IDBUF(86)
C      LET FLOATING BE AUTOMATIC:
C      DSTTIM = IDBUF(84)
C      DSTTIM = DSTTIM + DBUF(1)
C
C      DO 25, J=1,4
C      IDECBF(J) = IDBUF(J+90)
C
C      RLTIME = DBUF(1)
C      LIVTIM = DBUF(2)
C
C      RETURN
C      END

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      Generalized GFIT
C      25-MAR-80; REV 23-JUL-82 by W.D.Ruhter
C      Edited 28-Feb-83 by J.B.Niday (removed index locn to caller)
C      and 24-May-83(prevent crash on neg. Y values)
C      Edited 25-Jun-87 by R. Gunnink (Added out-of-bound check)
C
C      Fit Gaussian to upper half of a peak after stripping out the
C      tailing contribution. Return peak posn, height, and width. Return
C      negative error flag if no valid peak in region.
C      L      Approx index of peak posn in data array Y
C      M      No. of points beyond L to include
C      ALFA    Peak width parameter calculated (an approx received)
C      EXP1    Short term tail amplitude
C      EXP2    Short term tail slope
C      PKHT    Peak height to be returned.
C      ER      Uncertainty in peak position; unless its value is -.
C              Then ER is a flag (-1= Parabola nearly degenerate.
C              -2= Peak position out of bounds.)
C
C      Y      Data array received
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      SUBROUTINE GFIT(L,M,ALFA,EXP1,EXP2,PKPOS,PKHT,ER,Y)
C
C      DIMENSION Y(*)
C
C      Initializations
C
C      !In case of early return
C      PKPOS = L
C      ER = 0
C      SY = 0.0
C      SYY=0.
C      SXY=0.
C      SXXY=0.
C      C=M*(M+1)/3.
C      Y1 = Y(L+1)
C      Y2 = Y(L-1)
C      TPOS = .5 * (Y2-Y1) / (Y1 - 2. * Y(L) + Y2)
C      Y0 = Y(L)
C
C      Strip out the short term tail contribution to peak
C      I=-M
C      J=L+I
C      XL=I
C      DLX = XL - TPOS
C      Y1 = Y(J)
C      IF (DLX .GE. 0) GO TO 34
C      Y1=Y1-Y0 * EXP1* EXP(EXP2 * DLX) * (1. -EXP(.4*ALFA*DLX*DLX))
C      IF(Y1 .GT. 0)GO TO 37
C
C      WRITE (*,1034)(Y(K),K=L-M,L+M)
C1034  FORMAT(' A negative data value was incountered in GFIT.'/
C      + ' Its value will be set to 0.1. Data Values ='/1P(5E12.4))
C
C      Y1=Y(J)
C      IF(Y1 .LE. 0.)Y1 = 0.1
C
C      Take natural log of net 'Y' data. Result should be a parabola.
C      Fit to a parabola to get peak height, position, and width.
C
C37  Y1=LOG(Y1)
C      SY=SY+Y1
C      SYY=SYY+Y1**2
C      SXY=SXY+XL*Y1

```

```

SXXY=SXXY+(XL**2-C)*Y1
I=I+1
IF(I.LE.M)GOTO 30
C
C      COMPUTE THE DIAGONAL ELEMENTS OF X'X
C      XPX11=2*M+1.
C      XPX22=XPX11*C
C      XPX33=XPX22*(4*M*(M+1)-3)/15.
C      COMPUTE LEAST SQUARES ESTIMATES OF COEFFICIENTS
C      A0=SY/XPX11
C      A1=SXY/XPX22
C      ALFA=SXXY/XPX33
C      IS THE PARABOLA NEARLY DEGENERATE
C      !Default value
C
C      ER = -1.
C      IF(ABS(A0) .LT. 1.E-06) GOTO 50
C
C      COMPUTE THE COORDINATES OF THE VERTEX
C      PKPOS=-0.5*A1/ALFA
C      PKHT=A0+0.5*A1*PKPOS-ALFA*C
C      PKPOS = PKPOS + L
C
C      !Check bounds
C      IF ((PKPOS .LT. (L-M)) .OR. (PKPOS .GT. (L+M))) GOTO 40
C      ER=SY-A0*SY-A1*SXY-ALFA*SXXY
C      IF(M.GT.1)ER=ER/(2*(M-1))
C      PKHT=EXP(PKHT)
C      GOTO 60
C
C      !Set flag
C40  ER = -2.
C      !Return starting position
C50  PKPOS = L
C      !Return default height
C
C      PKHT = Y(L)
C      WRITE (*,9000) ER
C9000  FORMAT (' ***Error in GFIT*** Error flag = ',F3.0)
C
C60  RETURN
C      END

```

```
C=====
C      SUBROUTINE J2CDAT (DAY,NYR,MONTH,NDAY,NHOUR,MINUTE)
C
C      This routine converts Julian date to calender month, day, hour and
C      minute.
C      DAY = Julian day of year.
C      NYR = Year
C      NDAY = Calender day
C      NHOUR = Hour of the day.
C      MINUTE = Minute of the hour
C
C      Version 1A      31-May-84      R. Gunnink
C=====
C
C      DIMENSION NDAYS(12)
C      DATA NDAYS /31,28,31,30,31,30,31,31,30,31,30,31/
C
C      NDAY = DAY
C      HOUR = (DAY - FLOAT(NDAY)) * 24.
C      NHOUR = HOUR
C      MINUTE = (HOUR - FLOAT(NHOUR)) * 60.
C      IF (.25*FLOAT(NYR) - FLOAT(NYR/4) .EQ. 0) NDAYS(2) = 29
C
C      DO 10 MONTH = 1,12
C      IF (NDAY .LE. NDAYS(MONTH)) GOTO 20
C      NDAY = NDAY - NDAYS(MONTH)
10    CONTINUE
20    RETURN
C      END
```

SUBROUTINE MAXVAL (NS,NE,JMAX,YMAX,Y)

MAXVAL
=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

```
Title      : MAXVAL
Revision   : 1.0
Author     : R. Gunnink
           : Revised for the IBM PC by Russell Fleming
Date       : 10/15/86
Language   : Microsoft Fortran Ver. 4.01
```

Abstract :
This routine returns the position and value of the maximum value in a group of data points from NS to NE inclusive.

Arguments :

C	INPUT	
C	NS	starting position for group in Y array
C	NE	ending position for group in Y array
C	Y	data array
C	OUTPUT	
C	JPK	position in array Y of maximum data value
C	YMAX	maximum data value in group set by NS and NE

```
Routines
Called      : None

Libraries
Used        : None
```

Revision History (Include date,revision I.D., and details)

```
C Version: 2.1
C Author: R. Gunnink
C Date: 19-JAN-78
```

```
COMMON and variable declarations
```

DIMENSION Y(*)

```
variable initialization declarations
```

YMAX = 0.0
JMAX = NS

```
DO 2 I = NS,NE-1
  IF (Y(I) - YMAX) 2,2,4
  YMAX = Y(I)
  JMAX = I
CONTINUE
RETURN
END
```

SUBROUTINE NCNTS (N1,N2,M1,M2,ND,S1,S2,ER,AVEBG,Y1,Y2)

```
C*****
C*****
C*****
C      NCNTS
C     =====
C*****
```

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

```

C*****
C
C      Title      : NCNTS
C      Revision   : 1.0
C      Author     : Ray Gunnink
C      Date       : 11/9/87
C      Language   : Microsoft Fortran Version 4.01

```

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

Arguments : N1,N2,M1,M2,ND,S1,S2,ER,AVEBG,Y1,Y2

Routines
Called : AVE, BKGD

Libraries
Used : None

Revision History (Include date, revision I.D., and details) *

NCNTS VERSION 0 19-JAN-78 AT 12:30

COMMON and variable declarations

```

DIMENSION Y1(*),Y2(*)

```

variable initialization declarations

```
format declarations
```

```

C
C*****
C
      CALL AVE (N1,N2,S1,AV1,Y1)
      CALL AVE (M1,M2,S2,AV2,Y1)
      S1 = S1 + S2
      CALL BKGD (N2+1,M1-1,AV1,AV2,S2,ND,Y1,Y2)
      CHANS = ND
      XPTS = M2 + N2 - M1 - N1 + 1
      AVEBG = 0.5 * (AV1 + AV2)
      IF (S2 .LT. 0.0) S2 = 0.0
      ER = SQRT (S2 + AVEBG * CHANS * (1. + CHANS / XPTS))
      RETURN
      END

```

SUBROUTINE MATINV(ARRAY,MX,NORDER,DET)

```
C*****  
C*****  
C  
C      MATINV  
C     =====  
C*****
```

Developed for : Nuclear Chemistry Division
Safeguards Technology Program
Lawrence Livermore National Laboratory
Livermore, CA 94550

```

Title       :      MATINV
Revision    :      1.00
Author      :      Wayne D. Ruhter, modified by
                  Russell Fleming and William Buckley
Date        :      12 May 1986
Language    :      MS FORTRAN Ver. 4.01 (MS/PC-DOS)

```

Abstract:

This subroutine was adapted from the matrix inversion subroutine given in the book, "Data Reduction and Error Analysis for the Physical Sciences" by Philip R. Bevington. This subroutine inverts a square symmetric matrix and calculates its determinant. The input variables are ARRAY and NORDER, and the output variable is DET. The inverted matrix is returned in ARRAY. MX must be equal to the size of the array in the calling program.

DICTIONARY OF VARIABLES

```

ARRAY  = MATRIX WHOSE INVERSE IS TO BE EVALUATED
MX     = ARRAY DIMENSION
NORDR  = DEGREE OF THE MATRIX
DET    = VALUE OF THE DETERMINANT

```

Compiler Directives -- Metacommands

\$NOFLOATCALLS

\$NOLIST

Routines called : NONE

Called by : IAEAPU

Libraries used : 8087.LIB

COMMON and local variable declarations

C

```
C
C*****
C      Variable and parameter initialization                                *
C*****
C
```

```
DOUBLE PRECISION ARRAY(MX,MX)
DIMENSION IK(50),JK(50)
```

```
C*****
C*****
C      Format declarations
C*****
```

```
DET = 1.  
DO 100 K =1,NORDER
```

FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX

AMAX = 0.

```

21      DO 30 I =K,NORDER
22          DO 30 J =K,NORDER
23              IF (ABS(AMAX)-ABS(ARRAY(I,J)))24,24,30
24              AMAX = ARRAY(I,J)
25              IK(K) = I
26              JK(K) = J

```

CONTINUE

```
INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)
```

```

31      IF(AMAX)41,32,41
32      DET = 0.
      GOTO 140
41      I = IK(K)
      IF(I-K)21,51,43
43      DO 50 J = 1,NORDER
          SAVE = ARRAY(K,J)
          ARRAY(K,J) = ARRAY(I,J)
50      ARRAY(I,J) = -SAVE
51      J = JK(K)
      IF(J-K) 21,61,53
53      DO 60 I = 1,NORDER
          SAVE = ARRAY(I,K)
          ARRAY(I,K) = ARRAY(I,J)
60      ARRAY(I,J) = -SAVE

```

ACCUMULATE ELEMENTS OF INVERSE MATRIX

```

61      DO 70 I =1,NORDER
           IF(I-K)63,70,63
63      ARRAY(I,K) = -ARRAY(I,K)/AMAX
70      CONTINUE
71      DO 80 I =1,NORDER
           DO 80 J =1,NORDER
           IF(I-K)74,80,74
74      IF(J-K)75,80,75
75      ARRAY(I,J)= ARRAY(I,J) + ARRAY(I,K) * ARRAY(K,J)
80      CONTINUE
81      DO 90 J =1,NORDER
           IF(J-K)83,90,83
83      ARRAY(K,J) = ARRAY(K,J)/AMAX
90      CONTINUE
           ARRAY(K,K) = 1. / AMAX
100     CONTINUE

```

RESTORE ORDERING OF MATRIX

C


```
101 DO 130 L =1,NORDER
      K = NORDER - L + 1
      J = IK(K)
      IF(J-K)111,111,105
105 DO 110 I =1,NORDER
      SAVE = ARRAY(I,K)
      ARRAY(I,K) = -ARRAY(I,J)
      ARRAY(I,J) = SAVE
111 I = JK(K)
      IF(I-K)130,130,113
113 DO 120 J =1,NORDER
      SAVE = ARRAY(K,J)
      ARRAY(K,J) = -ARRAY(I,J)
      ARRAY(I,J) = SAVE
120 CONTINUE
130 RETURN
140 END
```

```

SUBROUTINE PKFIT (NPKS,NDPTS,IFLG,NYFLG,NXFLG,REFEN,GAIN,AVEBG,
+ EN,HIGHT,GAMA,YNET,DELY,RM,SHAPC,SUM,ER)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C This is a general purpose subroutine for fitting peak groupings.
C The maximum size of the peak grouping is determined by the size
C of the YNET array specified in the calling routine. The maximum
C number of peaks is currently set at 20. For each peak, one can
C specify whether the position and the area are to be determined.
C The peak width, the short- and long-term tailing amplitudes and
C slopes can also be determined in the fitting process. However,
C this routine is currently dimensioned to handle only 25 degrees
C of freedom. If all 5 shape parameters are being determined,
C and the peak positions and peak areas of all peaks are also being
C determined, then only 10 peaks can be handled. However, more
C peaks can be fit if some of the positions or areas are fixed
C relative to other peaks. This is done by using the NYFLG, NXFLG,
C and HIGHT variables as follows:
C NYFLG - +1 = peak area is to be determined.
C          0 = peak area is fixed at input value given by HIGHT(J)
C          -n = peak area is fixed relative to area of the nth peak.
C             The area ratio is given by -HIGHT(J)
C NXFLG - +1 = energy (position) is to be determined
C          0 = energy (position) is to be frozen at input value.
C          -n = energy (position) is fixed relative to nth peak
C The peak shape parameters can be freed or fixed at the input
C values using the NFLG variable as follows:
C NFLG(1) - FWHM peak width parameter (0 = fixed, 1 = free)
C NFLG(2) - Short-term tailing amplitude (0 = fixed, 1 = free)
C NFLG(3) - Short-term tailing slope (0 = fixed, 1 = free)
C NFLG(4) - Long-term tailing amplitude (0 = fixed, 1 = free)
C NFLG(5) - Long-term tailing slope (0 = fixed, 1 = free)
C Other flags are:
C NFLG(6) - Printout flag (0 = none, +n = printout)
C NFLG(7) - Device number of printer.
C NFLG(8) - Number of additional peaks that may be added.
C
C Please note by looking at the initial program instructions that
C the values as obtained in the SHAPC array will be treated in one
C of two ways, depending on the sign of the value stored in
C SHAPC(1). If its value is negative, then it is understood that
C the explicit values for ALPHA, EXP1, EXP2, EXP3, and EXP4 are
C stored in the SHAPC array and these will be used in the fitting process.
C (ALPHA = -2.7726/(FWHM*GAIN**2)). If SHAPC(1) is positive,
C the values stored in the SHAPC file are assumed to be the shape
C constants from which peak fitting parameters can be calculated.
C
C This routine can also fit x ray peaks. If the Jth peak is an
C an x ray, the x ray line-width, in eV, must be stored in the
C corresponding Jth value of GAMA. Alternatively, a negative (-1)
C value will also signal the routine that a peak is an x ray, and
C an approximate line-width will be calculated based on the energy
C of the peaks in this region.
C
C This routine was taken from the FIT routine used in the GRPANL
C program.

```

Ray Gunnink 20-OCT-88

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

COMMON Block for subroutine PKFIT.

```

+ DIMENSION IFLG(*),NYFLG(*),NXFLG(*),EN(*),HIGHT(*),
+ GAMA(*),PKPOS(20),PKHT(20),DM(25),DEL(25)

```

```

+ DIMENSION SUM(*),ER(*),SHAPC(*),YNET(*),DELY(*),RM(*),
+ BG(2),ERR(7),NFLG(8)
DOUBLE PRECISION APHA(25,25),XSUM(25),AX(25)
C
C -----
C
NP = NPKS
MXFP = 25
NFP = 0
INSRT = 0
IF (SHAPC(1) .LT. 0.) THEN
C Read in shape parameters from shapc file
ALPHA = SHAPC(1)
EXP1 = SHAPC(2)
EXP2 = SHAPC(3)
EXP3 = SHAPC(4)
EXP4 = SHAPC(5)
S1 = -2.7726/(ALPHA*GAIN**2) -.0019*EN(1)
S2 = .0019
ELSE
C IF (SHAPC(1) .EQ. 0) STOP ' SHAPC(1) = 0.0 in PKFIT'
C Calculate shape parameters from file values
CALL SPRAM (EN(1),GAIN,SHAPC,SG,ALPHA,F12M,EXP1)
EXP2 = SHAPC(5) * GAIN
EXP3 = SHAPC(7)
EXP4 = SHAPC(8) * GAIN
S1 = SHAPC(1)
S2 = SHAPC(2)
ENDIF
AOLD = ALPHA
DELOLD = EXP1
EXP1OLD = EXP1
EXP2ST = EXP2
EXP3ST = EXP3
CHGSLP = .5*GAIN
C
DO 8 I = 1,8
NFLG(I) = IFLG(I)
C
DO 18 J = 1,NP
IF (EN(J) .GT. 130.) GAMA(J) = 0.0
IF (GAMA(J) .NE. 0.0) THEN
IF (GAMA(J) .LT. 0.0) THEN
C Calculate the x ray line width
GAMA(J) = -.4+.4*EN(J)
IF (EN(J) .GT. 28.) GAMA(J)=GAMA(J)-27.22+.972*EN(J)
ENDIF
C Determine GAMA in channel units
GAMA(J) = GAMA(J)*.001/GAIN
ENDIF
18 CONTINUE
C
DO 20 J = 1,NP
C Determine peak position relative to start of group
PKPOS(J) = (EN(J) - REFEN)/GAIN
NPK = PKPOS(J) + 0.5
C Obtain estimate of peak height.
PKHT(J) = YNET(NPK)
IF (HIGHT(J) .GT. 0.) THEN
C Peak area is prescribed by input
PKHT(J) = HIGHT(J) / SQRT(-3.14/ALPHA)
WL = GAMA(J) * SQRT(-ALPHA)
C Check if it an x ray
IF (WL .GT. 0.) PKHT(J) = PKHT(J)*(.1-.5642*WL+.25*WL**2)
ENDIF
C Adjust relative intensities for local variation of shapes

```

```

      IF (HIGHT(J) .GE. 0.) GOTO 20
      K = -NYFLG(J)
      HIGHT(J)=HIGHT(J)*SQRT((S1+S2*EN(K))/(S1+S2*EN(J)))
      RATIO = 0.0
      IF ((GAMA(K) .EQ. 0.) .AND. (GAMA(J) .EQ. 0.)) GOTO 20
      IF ((GAMA(K) .GT. 0.) .AND. (GAMA(J) .GT. 0.)) GOTO 20
C     X ray and Gamma Ray are being related. Make height adjustment.
      WL = (GAMA(J)+GAMA(K))*SQRT(-ALPHA)
C     !One of GAMA's = 0.0
      RATIO = 1.-.5642*WL +.25*WL**2
C     !X ray/Gamma peak height ratio.
      IF (GAMA(K) .GT. 0.0) RATIO = 1. / RATIO
      HIGHT(J) = HIGHT(J) * RATIO
20    CONTINUE
C
C     FIX THE INTENSITY RELATIONSHIPS THAT HAVE BEEN SPECIFIED
C     IN THE INPUT. FIRST RE-ADJUST PEAK HEIGHT
C     ESTIMATES FOR INTERFERENCES FROM NEIGHBORING PEAKS.
C
      DO 25 J=1,NP
      IF((NYFLG(J) .GT. 0) .OR. (HIGHT(J) .GE. 0)) GOTO 25
      K = -NYFLG(J)
      DX = PKPOS(J) - PKPOS(K)
      FCT = -HIGHT(J) * EXP(ALPHA * DX**2)
      IF (HIGHT(K) .LE. 0.) PKHT(K) = PKHT(K) / (1. + FCT)
      PKHT(J) = -HIGHT(J) * PKHT(K)
25    CONTINUE
C
      DO 30 J = 1,NP
      IF (NYFLG(J) .LT. 0) GOTO 30
      L = J - 1
      CONST = 0.
      IF (L .LE. 0) GOTO 29
C     Is neighbor related?
27    IF (NYFLG(L) .LT. 0) THEN
      IF (-NYFLG(L) .EQ. J) GOTO 29
      ENDIF
      DX = PKPOS(J) - PKPOS(L)
      EXPN = EXP(ALPHA * DX**2)
      PKHT(J) = PKHT(J) - PKHT(L) * EXPN * (1.- CONST * EXPN)
      IF (PKHT(J) .LE. 0.0) PKHT(J) = -1.0
29    IF (L .GT. J) GOTO 30
      CONST = 0.5
      L = J + 1
      IF (L .LE. NP) GOTO 27
30    CONTINUE
C
C     FIX THE ENERGY RELATIONSHIPS THAT HAVE BEEN
C     SPECIFIED IN THE INPUT.
C
      DO 39 J=1,NP
      IF(NXFLG(J) .GE. 0)GO TO 39
      L = -NXFLG(J)
      PKPOS(J) = PKPOS(L) -(EN(L) - EN(J))/GAIN
39    CONTINUE
C     !Max iterations
      MAXITR = 10
      NFLG2 = NFLG(2)
      NFLG3 = NFLG(3)
      ASLP = S2/GAIN
      ITER = 1
      ICNT2 = 0
      ICNT3 = 0
      LFLG2 = 0
      LFLG4 = 0
C
C     !INIT ITERATION COMPLETION FLAG

```

```

      ITFLG=0
C
C     The Five NFLG's determine whether FWHM, short-term tail size &
C     slope, and long-term tail size & slope, should held fixed or be
C     determined during the fitting process.
C     Iteration comes back to here.
C
45    KK = NFLG(1) + NFLG(2) + NFLG(3) + NFLG(4) + NFLG(5)
C
      SG = -2.7726 / ALPHA
      CX = SQRT(-ALPHA)
      ES2 = (.7*EXP2/GAIN)**2
      XPTS = NDPTS
C
C     CALCULATE PEAK TAILING INTENSITIES AT THE GROUP BOUNDARIES.
C     THESE VALUES USED LATER TO ADJUST NET COUNTS OF PEAK
C     GROUPING.
C
      BG(1) = 0.
      BG(2) = 0.
C
      DO 55 J = 1,NP
      AMP1 = EXP1
      AMP2 = EXP3
      XI = PKPOS(J)
C
      DO 50 L = 1,2
      CC = 0.0
      IF (GAMA(J) .EQ. 0.) GOTO 52
      WL = GAMA(J) * CX
      CC = CX * XI
      EXPNT = ALPHA * XI * XI
      EXPN1 = 0.
      IF(EXPNT .GT. -14.)EXPN1 = EXP(EXPNT)
      CALL BWF2 (WL,CC,EXPNT,EXPN1)
52    BG(L)=BG(L)+PKHT(J)*(CC+AMP1*EXP(-EXP2*X1)
      * (1.- EXP(.4*ALPHA*XI*XI)) + AMP2*EXP(-EXP4*X1))
      AMP1 = 0.
      AMP2 = 0.
      XI = XPTS - PKPOS(J)
50    CONTINUE
55    CONTINUE
56    YP = 0.
      DBG = BG(2) - BG(1)
      CALL AVE(1,NDPTS,YNSUM,AV,YNET)
C
C     Zero out coefficient array
C
      DO 60 J=1,MXFP
      XSUM (J) = 0.
      DO 60 I=1,MXFP
      APHA(I,J)=0.
60    DO 61 J=1,NP
      SUM(J) = 0.0
      CHISQ = 0.
C
C     Form matrix of linear equations
C
      DO 200 I = 1,NDPTS
      DO 65 JK = 1,MXFP
      AX(JK) = 0.
      XI = I
      FX = 0.0
      DALFA = 0.0
      DBETA = 0.0
      DGAMA = 0.0
      DDELT = 0.0
65

```

```

DCHI = 0.0
YP = YP + YNET(I)
DELY(I) = YNET(I) + BG(1) + YP*DBG/YNSUM
RM(I) = DELY(I)
TYNET = YNET(I)
IF (TYNET .LE. 0) TYNET = TYNET + 1.
EY = ABS(TYNET) + AVEBG + 1.0
IF (EY .LT. 0.) EY = 1.0
C                                     Channel width undertain
EY = SQRT(EY) + 0.001*EY
C                                     Determine weighting factor
WT = 1.0 / EY
KPK = KK
C
C                                     Calculate coefficients for each of "NP" peaks
DO 170 J = 1, NP
  DLTAX = XI - PKPOS(J)
  DX = DLTAX
  DLX = DLTAX
  DLXSQ = DLTAX * DLTAX
  FX2 = 0.0
  FX3 = 0.0
  CB1 = 0.
  CB2 = 0.
  EXPN1 = 0.0
  PHT = PKHT(J)
  IF (PHT .LT. 0.0) PHT = -1.
  ALFA = -2.7726 / (ASLP * PKPOS(J) + SG)
  EXPNT = ALFA * DLXSQ
  EXPN3 = 0.
  IF (EXPNT .GT. -14.) EXPN1 = EXP(EXPNT)
C                                     Calculate Gaussean component of peak
  IF (DLTAX .LT. 0) EXPN3 = EXP (0.4 * EXPNT)
  FX1 = EXPN1
  IF (GAMA(J) .GT. 0) THEN
C                                     Compute modifies shape for x ray peak
    CX = SQRT(-ALFA)
    WL = GAMA(J) * CX
    CC = CX * DLTAX
    CALL BWF2 (WL, CC, EXPNT, EXPN1)
    FX1 = CC
  ENDIF
  IF (NFLG(1) .GT. 0) THEN
C                                     !Peak width parameter free
    DALFA = DALFA + PHT * DLXSQ * EXPN1
  ENDIF
  IF (DLTAX .GE. 0) GOTO 130
  EXPN = EXP2 * DLTAX
  IF (EXPN .LT. -10.) GOTO 110
  EXPN2 = EXP (EXPN)
  CB = PHT * EXPN2 * (1.0 - EXPN3)
  CB1 = EXP1 * EXP2 * CB
  FX2 = EXP1 * CB
C                                     !Short term tailing amplitude free
  IF (NFLG(2) .GT. 0) DBETA = DBETA + CB
C                                     Short term tailing slope free
  IF (NFLG(3) .GT. 0) DGAMA = DGAMA + DLTAX * CB
C
C110 IF (EXP3 .LE. 0) GOTO 120
110 EXPN = EXP4 * DLTAX
  IF (EXPN .LT. -10.) GOTO 120
  CB = PHT * EXP(EXPN) * (1. - EXPN3)
  CB2 = EXP3 * EXP4 * CB
  FX3 = EXP3 * CB
C                                     !Long term tailing amplitude free
  IF (NFLG(4) .GT. 0) DDELT = DDELT + CB

```

```

C                                     !Long term tailing slope free
  IF (NFLG(5) .GT. 0) DCHI = DCHI + DLTAX * CB
C
120 IF (NFLG(3)+NFLG(5) .EQ. 0) THEN
  IF (NFLG(2) .EQ. 0) FX = FX + FX2
  IF (NFLG(4) .EQ. 0) FX = FX + FX3
ELSE
  FX = FX + FX2 + FX3
ENDIF
C
130 IF (NYFLG(J) .GT. 0) THEN
C                                     !Peak size free. Count free parameters
    KPK = KPK + 1
    L = KPK
    VALU = FX1 * WT
    GO TO 140
  ENDIF
C                                     Peak size is not free. [NYFLG is not positive.]
  IF (HIGHT(J) .GE. 0.) THEN
C                                     !Peak size is fixed absolutely.
    FX = FX + FX1 * PHT
    GO TO 160
  ENDIF
C                                     Peak size is fixed relative to another peak.
C                                     !Find index of primary peak.
    LPK = -NYFLG(J)
    IF (HIGHT(LPK) .GT. 0.) THEN
C                                     !Primary is fixed also.
      FX = FX - HIGHT(J) * PKHT(LPK) * FX1
      GOTO 160
    ENDIF
C                                     VALU = -HIGHT(J) * FX1 * WT
    L = KK
    IF (LPK .GT. 1) THEN
C                                     !Skip stg. space already used up.
      DO 135 M = 1, LPK - 1
        IF (NYFLG(M) .GT. 0) L = L + 1
        IF (NXFLG(M) .GT. 0) L = L + 1
      CONTINUE
135 ENDIF
C                                     !Locn. indexing profile of primary .
      L = L + 1
C
140 AX(L) = VALU + AX(L)
C
C                                     Peak position free?
160 IF (NXFLG(J) .GT. 0) THEN
C                                     !Yes. Count the free parameter
      KPK = KPK + 1
      AX(KPK) = (-2. * ALFA * DX * EXPN1 * PHT - CB1 - CB2) * WT
    ENDIF
C
    SUM(J) = SUM(J) + FX1 * PHT + FX2
    RM(I) = RM(I) - FX1 * PHT - FX2 - FX3
170 CONTINUE
C
    L = 0
    IF (NFLG(1) .GT. 0) THEN
      L = L + 1
      AX(L) = DALFA * WT
    ENDIF
    IF (NFLG(2) .GT. 0) THEN
      L = L + 1
      AX(L) = DBETA * WT

```

```

      ENDIF
      IF (NFLG(3) .GT. 0) THEN
        L = L + 1
        AX(L) = DGAMA * EXP1 * WT
      ENDIF
      IF (NFLG(4) .GT. 0) THEN
        L = L + 1
        AX(L) = DDELT * WT
      ENDIF
      IF (NFLG(5) .GT. 0) THEN
        L = L + 1
        AX(L) = DCHI * EXP3 * WT
      ENDIF
      DELY(1) = (DELY(1)-FX) * WT
      DO 192 KM = 1,KPK
        XSUM(KM) = XSUM(KM) + AX(KM)*DELY(1)
      DO 190 KL = 1,KPK
        APHA(KM,KL) = APHA(KM,KL) + AX(KM)*AX(KL)
      CONTINUE
      RM(1) = RM(1) * WT
      CHISQ = CHISQ + RM(1)**2
      CONTINUE
      Matrix is now loaded.
      IF (ITER .EQ. 1) GOTO 210
      RCHISQ = CHISQ/(NDPTS-KPK)
      IF (NFLG(6) .GT. 0) WRITE (NFLG(7),1208) RCHISQ
      FORMAT(' REDUCED CHI-SQUARE VALUE IS ',F7.1)
      1208 C
      210 IF (ITFLG .EQ. 1) GOTO 500
      CALL MATINV(APH, MXFP, KPK, DET)
      C
      IF (DET .EQ. 0.) STOP 'Matrix is singular in PKFIT'
      C
      C      !For FWHM at mid-energy
      POS = NDPTS/2
      C
      C      !At peak position
      IF (NP .EQ. 1) POS = PKPOS(1)
      FWHM = GAIN * SQRT(ASLP*POS + SG - .462)
      EX2 = EXP2/GAIN
      EX4 = EXP4/GAIN
      IF (NFLG(6) .GT. 0) THEN
        WRITE (NFLG(7),9040)
        9040 FORMAT(/, ' ITERATION FWHM EXP1 EXP2 EXP3 EXP4')
        WRITE (NFLG(7),9030) ITER,FWHM,EXP1,EX2,EXP3,EX4
        9030 FORMAT(I3,6X,F9.5,4F8.4)
        WRITE (NFLG(7),9010)
        9010 FORMAT(/, ' CHANNEL KEV PEAK HEIGHT AREA %ERR')
        WRITE (NFLG(7),9020) (PKPOS(J),PKPOS(J)*GAIN+REFEN,PKHT(J),
        + SUM(J),100.*ER(J)/(SUM(J)+1.),J=1,NP)
        9020 FORMAT(2F9.2,2F12.2,F7.1)
      ENDIF
      C
      DO 310 L = 1,KPK
        DEL(L) = 0.
      C
      DM(L) = SQRT(ABS(APH(L,L)))
      DO 310 J=1,KPK
        DEL(L) = DEL(L) + XSUM(J)*APH(L,J)
      310 C
      C      INCREMENT THE TRIAL VALUES AND CHECK FOR COMPLETION OF ITERATIVE
      C      PROCESS.

```

```

      DO 320 J = 1,5
      IF (NFLG(J) .LE. 0) ERR(J+2) = 0.
      320 C      CONTINUE
      L = 0
      LFLG = 0
      IF (NFLG(1) .GT. 0) THEN
        C
        C      Increment peak width parameter
        L = L + 1
        ERR(3) = DM(L)
        ADEL = ABS(DEM(L))
        ALPHA = ALPHA + DEL(L) * (1. + ADEL/(ALPHA - ADEL))
        IF ((ADEL + .01 * ALPHA) .GT. 0) LFLG = 1
        IF (AOLD/ALPHA .GT. 3. .OR. ALPHA/AOLD .GT. 3.) THEN
          C      Apply limits to width. Remember alpha is -.
          NFLG(1) = 0
          IF (ALPHA .LT. AOLD) THEN
            ALPHA = AOLD*3.
          ELSE
            ALPHA = AOLD/3.
          ENDIF
        ENDIF
        ENDIF
        IF (NFLG(2) .LE. 0) GO TO 425
        L = L + 1
        ERR(4) = DM(L)
        C
        C      Increment short term tailing amplitude
        IF (NFLG(3)+NFLG(5) .NE. 0) THEN
          IF ((EXP1 + DEL(L)) .LE. 0) DEL(L) = -.5 * EXP1
          ADEL = ABS(DEM(L))
          EXP1 = EXP1 + DEL(L) * (1. - ADEL/(EXP1 + ADEL))
          IF (ADEL .GT. (DM(L) + .02 * EXP1)) LFLG = 1
        ELSE
          IF (DEL(L) .LT. 0.) THEN
            IF (NFLG(6) .GT. 0) WRITE (NFLG(7),1419)
            1419 FORMAT(' SHORT TERM TAILING AMPLITUDE NEGATIVE--SET TO .0001')
            DEL(L) = 0.0001
            LFLG2 = LFLG2 + 1
            IF (LFLG2 .GT. 4) THEN
              NFLG(2) = 0
              LFLG = 1
            ENDIF
          ENDIF
          IF (ABS(EXP1 - DEL(L)) .GT. DM(L)) LFLG = 1
          DEL(L) = .5*(DELOLD + DEL(L))
          DELOLD = DEL(L)
          EXP1 = DEL(L)
        ENDIF
        IF (EXP1 .GT. ES2) THEN
          EXP1 = ES2
          ICNT2 = ICNT2 + 1
        ENDIF
        C
        425 C      IF (NFLG(3) .GT. 0) THEN
        C
        C      Increment short term tailing slope
        L = L + 1
        ERR(5) = DM(L)
        IF ((EXP2 + DEL(L)) .LE. 0) DEL(L) = -.5 * EXP2
        ADEL = ABS(DEM(L))
        EXP2 = EXP2 + .5*DEL(L) * (1. - ADEL / (EXP2 + ADEL))
        IF (ADEL .GT. (DM(L) + .02 * EXP2)) LFLG = 1
        DIF = EXP2 - EXP2ST

```

```

      IF (DIF .GT. CHGSLP) EXP2 = EXP2ST + CHGSLP
      IF (DIF .LT. -CHGSLP) EXP2 = EXP2ST - CHGSLP
      IF (ABS(DIF) .GT. CHGSLP) ICNT3 = ICNT3 + 1
    ENDIF
    IF (NFLG(4) .LE. 0) GOTO 440
  C
  C      Increment long term tailing amplitude
  L = L + 1
  ERR(6) = DM(L)
  IF (NFLG(3)+NFLG(5) .NE. 0) THEN
  C
  C      IF ((EXP3 + DEL(L)) .LE. 0) DEL(L) = -.5 * EXP3
  ADEL = ABS(DEL(L))
  EXP3 = EXP3 + DEL(L) * (1. - ADEL/(EXP3 + ADEL))
  IF (ADEL .GT. (DM(L) + .05 * EXP3)) LFLG = 1
  GOTO 440
  ENDIF
  C
  IF (DEL(L) .LT. -.01) THEN
  C
  C      IF (NFLG(6) .GT. 0) WRITE (NFLG(7),1437)
  1437 FORMAT(' LONG TERM TAILING AMPLITUDE NEGATIVE--SET TO -.01')
  DEL(L) = -0.01
  LFLG4 = LFLG4 + 1
  IF (LFLG4 .GT. 4) THEN
  C
  C      NFLG(4) = 0
  LFLG = 1
  ENDIF
  ENDIF
  IF (ABS(EXP3 - DEL(L)) .GT. DM(L)) LFLG = 1
  IF (ABS(EXP3 - DEL(L)) .GT. DM(L)+.00001) LFLG = 1
  EXP3 = DEL(L)
  440 IF (NFLG(5) .GT. 0) THEN
  C
  C      Increment long term tailing slope
  L = L + 1
  ERR(7) = DM(L)
  IF (EXP4 .LE. -DEL(L)) DEL(L) = -0.5 * EXP4
  ADEL = ABS(DEL(L))
  EXP4 = EXP4 + DEL(L)*(1.-ADEL/(EXP4+ADEL))
  IF (ADEL .GT. (DM(L) + 0.02*EXP4)) LFLG = 1
  ENDIF
  K = L
  C
  DO 460 J = 1,NP
  C
  C      If fixed height
  IF (NYFLG(J) .GT. 0) THEN
  C
  C      Get new peak heights
  K = K + 1
  PHT = DEL(K)
  IF (SUM(J) .LE. 0.0) PHT = -1.
  ER(J) = DM(K)*SUM(J)/PHT
  PKHT(J) = DEL(K)
  IF (DEL(K) .LT. 0.0) PKHT(J) = -1.0
  ENDIF
  IF (NXFLG(J) .GT. 0) THEN
  C
  C      Increment peak positions
  K = K + 1
  ADEL = ABS(DEL(K))
  PKPOS(J) = PKPOS(J) + DEL(K) * (1. - ADEL/(1.+ADEL))
  IF (ADEL .GT. (DM(K) + .01)) LFLG = 1
  ENDIF
  460 CONTINUE
  C
  RE-ESTABLISH PEAK ENERGY AND INTENSITY RELATIONSHIPS
  SPECIFIED IN INPUT.

```

```

      DO 470, J = 1,NP
      IF (NYFLG(J) .GE. 0) GOTO 465
      IF (HIGHT(J) .GT. 0) GOTO 464
      IF (HIGHT(J) .EQ. 0) THEN
  C
  C      !Eliminated peak.
      PKHT(J) = 0
      GO TO 465
    ENDIF
  C
  L = -NYFLG(J)
  PKHT(J) = -HIGHT(J) * PKHT(L)
  GO TO 465
  464 PKHT(J) = PKHT(J) * HIGHT(J) / SUM(J)
  C
  465 IF (NXFLG(J) .GE. 0) GOTO 470
  L = -NXFLG(J)
  PKPOS(J) = PKPOS(L) - (EN(L) - EN(J))/GAIN
  CONTINUE
  470 IF (LFLG .EQ. 0) ITFLG = 1
  IF (ITER .GE. MAXITR) ITFLG = 1
  IF (ITFLG .EQ. 0) ITER = ITER + 1
  IF (ICNT2 .GT. 3) NFLG(2) = 0
  IF (ICNT3 .GT. 2) NFLG(3) = 0
  IF (ITER+2 .EQ. MAXITR) THEN
  C
  C      NFLG(3) = 0
  NFLG(5) = 0
  IF (EXP1 .EQ. ES2) NFLG(2) = 0
  ENDIF
  C
  C      Iterate calculation again
  GOTO 45
  C
  500 IF (ITER .GE. MAXITR .AND. NFLG(6) .GT. 0) WRITE (NFLG(7),1500)
  1500 FORMAT(' Maximum Iteration Reached in PKFIT.')
  C
  C      Check for missing peaks. If found, set flags & retry fitting.
  K = KK
  DO 530, J=1,NP
  IF (PKHT(J) .LT. 0) THEN
  C
  C      HIGHT(J) = 0.
  PKHT(J) = 0.
  NYFLG(J) = -J
  NXFLG(J) = 0
  SUM(J) = 0.
  ITER = 6
  ITFLG = 0
  ENDIF
  IF (NYFLG(J) .GT. 0) THEN
  C
  C      K = K + 1
  ER(J) = DM(K)*SUM(J)/PKHT(J)
  ENDIF
  IF (NXFLG(J) .GT. 0) K = K + 1
  530 CONTINUE
  DO 540 J = 1,NP
  K = J
  IF (NYFLG(J) .LT. 0) K = -NYFLG(J)
  ER(J) = ER(K)
  540 CONTINUE
  IF (ITFLG .EQ. 0) GOTO 40
  C
  IF (NFLG(6) .GT. 0) WRITE (NFLG(7),999) (I,REFEN+I*GAIN,
  + YNET(I),RM(I),I=1,NDPTS)
  999 FORMAT (I4,F8.2,F8.0,F8.2)
  C
  C      Check for significant unaccounted peaks by examining residuals.

```

```
C
  IF (NFP .GE. MXFP-1) RETURN
  RMIN = 15.
  RMAX = 0.0
  DO 550 I = 2,NDPTS-1
    AVRM = RM(I-1)+RM(I)+RM(I+1)
    IF (AVRM .GT. RMIN) THEN
      IF (AVRM .GT. RMAX) THEN
        IRM = I
        RMAX = AVRM
      ENDIF
    ENDIF
550  CONTINUE
    IF (RMAX .GT. 0.0) THEN
      NFP = KPK + INSRT*2
      IF ((NP .LT. NPKS+NFLG(8)) .AND. (NFP .LT. MXFP-1) .AND.
+       (ABS(IRM-IRMOLD) .GT. 1)) THEN
9100  IF (NFLG(6) .GT. 0)WRITE (NFLG(7),9100) IRM*GAIN+REFEN,RMAX/3.
      FORMAT (' Spurious peak detected at',F7.1,' keV. Ave. ',
+       'Residual =',F5.0/' A peak will be inserted at this energy',
+       ' and the region refitted')
      IRMOLD = IRM
      NP = NP + 1
      INSRT = INSRT + 1
      PKPOS(NP) = IRM
      PKHT(NP) = .5*YNET(IRM)
      NYFLG(NP) = 1
      NXFLG(NP) = 1
      HIGHT(NP) = 0.
      GAMA(NP) = 0.
      NFLG(2) = NFLG2
      NFLG(3) = NFLG3
      ALPHA = AOLD
      EXP1 = EXP1OLD
      DELOLD = EXP1
      EXP2 = EXP2ST
      EXP3 = EXP3ST
      GOTO 40
    ENDIF
  ENDIF
C
  REFEN = ALPHA
  RETURN
C
END
```



```

C
C      SUBROUTINE RDBDAT(NDEV,NS,NE,IBSIZ,DATA)
C-----
C      This routine reads spectrum data for a specified number of
C      spectrum channels for a block-structured file. (E.g. CB S-100
C      and ND accuSpec files.) NDEV = device
C      NS = starting channel, NE = ending channel, IBSIZ = channels per
C      block, and DATA = destination of data.
C-----
C
C      INTEGER*4 BUFF(32)
C      DIMENSION DATA(*)
C
C      First block is a header block; therefore use offset.
C      DATA INBK/1/
C
C      NUM = NE - NS + 1
C      ISTBLK = 1 + (NS-1)/IBSIZ
C      LASTBK = 1 + (NE-1)/IBSIZ
C      ISTD = NS - IBSIZ*(ISTBLK-1)
C      NDXD = 1
C      NWDCT = 0
C      NDX1 = 1
C      DO 15 I = ISTBLK, LASTBK
C      NWDS = IBSIZ+1 - ISTD
C      IF ((NWDCT+NWDS) .GT. NUM) NWDS = NUM - NWDCT
C      IBK = I + INBK
C      READ (NDEV,REC=IBK) (BUFF(K),K=1,IBSIZ)
C      DO 10 J = 1,NWDS
C      K = NDXD + J + 1
C      DATA(K) = FLOAT(BUFF(ISTD+J-1))
C      CONTINUE
10    NWDCT = NWDCT + NWDS
C      NDXD = NDXD + NWDS
C      ISTD = 1
15    CONTINUE
C
C      RETURN
C      END

```

```

      SUBROUTINE RDBLK(ISIZE,IBUFF,IBLK,ICHAN)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      READ ONE BLOCK OF DIRECT ACCESS FILE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      DIMENSION IBUFF(1)
C
C      READ(ICHAN,REC=IBLK,ERR=95)(IBUFF(I),I=1,128)
C      RETURN
C
C 95    WRITE (*,100) IBLK
100    FORMAT (' ERROR READING BLOCK:',I5)
      IF (ISIZE .LE. 256) STOP ' ATTEMPTED .GT. 256'
      STOP ' ATTEMPT TO PASS EOF'
      END

```

10

```
C*****
C
C
C*****
C
      SUBROUTINE RDCBDAT (DAT,ISTCHN,NUMCHS,INPDEV)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   A UTILITY SUB TO READ DATA IN SPECIFIED CHANNELS FROM A
C   PREVIOUSLY OPENED CANBERRA S-100 SPECTRAL FILE, CONVERT TO REAL NUMBERS,
C   AND RETURN THEM TO CALLER'S BUFFER. CALL WITH ADDRESS OF DESTINATION
C   BUFFER, STARTING CHANNEL, # OF CHANS, AND DEVICE NUMBER ASSIGNED
C   TO THE SOURCE FILE. CALLING PROGRAM IS RESPONSIBLE FOR ADEQUATE SIZE
C   OF DESTINATION BUFFER & FOR IDENTIFICATION OF SOURCE FILE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       INTEGER*4 DATA(32)
C       DIMENSION DAT(*)
C
C   first block is header block; use offset
C   DATA INBK/1/
C
C       LASTCH = ISTCHN + NUMCHS - 1
C       ISTBLK = 1 + (ISTCHN-1)/32
C       LASTBK = 1 + (LASTCH-1)/32
C       ISTD = ISTCHN - 32*(ISTBLK-1)
C       NDXD = 1
C       NWDDCT = 0
C       NDX1 = 1
C       DO 15 II = ISTBLK,LASTBK
C         NWDS = 33 - ISTD
C         IF ((NWDDCT+NWDS).GT.NUMCHS)NWDS = NUMCHS - NWDDCT
C         IBK = II + INBK
C         READ(INPDEV,REC=IBK)DATA
C         DO 10 J = 1,NWDS
C           KK = NDXD+J-1
C           DAT(KK)= FLOAT(DATA(ISTD+J-1))
10        CONTINUE
C         NWDDCT = NWDDCT + NWDS
C         NDXD = NDXD + NWDS
C         ISTD = 1
15        CONTINUE
C
C       RETURN
C       END
```

SUBROUTINE RDCHNS (DAT,ISTCHN,NUMCHS,INPDEV)

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   A UTILITY SUB TO READ DATA IN SPECIFIED CHANNELS FROM A
C   PREVIOUSLY OPENED STANDARD SPECTRAL FILE, CONVERT TO REAL NUMBERS,
C   AND RETURN THEM TO CALLER'S BUFFER. CALL WITH ADDRESS OF DESTINATION
C   BUFFER, STARTING CHANNEL, # OF CHANS, AND DEVICE NUMBER ASSIGNED
C   TO THE SOURCE FILE. CALLING PROGRAM IS RESPONSIBLE FOR ADEQUATE SIZE
C   OF DESTINATION BUFFER & FOR IDENTIFICATION OF SOURCE FILE.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

INTEGER*2 ISTCHN,NWDCT,IDAT(256)

INTEGER*4 JDAT(128)

EQUIVALENCE (IDAT,JDAT)

REAL*4 DAT(1)

DATA INBK/1/

```

C   NOTE THAT BLOCK 1 CONTAINS HEADER INFORMATION BLOCK
C   CALCULATE LAST CHANNEL # AND FIRST & LAST DATA BLOCK NEEDED
      LASTCH = ISTCHN + NUMCHS - 1
      ISTBLK = 1 + (ISTCHN - 1)/128
      LASTBK = 1 + (LASTCH - 1)/128
C   CALCULATE FIRST ADDRESS(CHANNEL&WORD) IN FIRST BLOCK
      ISTC = ISTCHN - 128*(ISTBLK-1)
      NDX1 = 2*ISTC - 1
C   INIT. INDEX OF REAL OUTPUT & RUNNING WORD COUNT
      NDXD = 1
      NWDCT = 0
C   NOTE THAT NWDS = # OF WORDS TO BE PROCESSED IN CURRENT BUFFER,
C   AND NWDCT = RUNNING SUM OF NWDS.

```

DO 30, IBK = ISTBLK, LASTBK

NWDS = 257 - NDX1

IF((NWDCT+NWDS).GT.(NUMCHS*2)) NWDS = (NUMCHS*2)-NWDCT

IB = INBK + IBK

CALL RDBLK (256,IDAT,IB,INPDEV)

CALL FLTBF (IDAT(NDX1),DAT(NDXD),NWDS)

NWDCT = NWDCT + NWDS

NDX1 = 1

NDXD = NDXD + NWDS/2

30 CONTINUE

RETURN

END

```

      SUBROUTINE RDHDR(IBUFF,ICHAN)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      READ ONE BLOCK OF DIRECT ACCESS FILE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER*2 ODAT(2),NDAT(2)
      INTEGER*4 IBUFF(256),DATA,Y,ISIZE
      EQUIVALENCE (DATA,ODAT(1)),(Y,NDAT(1))
C
      DO 10 I=1,256
      READ(ICHAN,REC=I)DATA
      NDAT(1)=ODAT(2)
      NDAT(2)=ODAT(1)
10    IBUFF(I)=Y
      RETURN
      END

```

```

C*****
C
C      NOTE: THIS SUBROUTINE SWITCHES THE ORDER OF THE HIGH AND LOWER
C      ORDER BYTES!
C      TO BE USED FOR DATA TAKEN DIRECTLY FROM A NUCLEAR DATA UMCA
C      USING THE "ASAP" SOFTWARE
C*****
C
      SUBROUTINE RDNDAT (DAT,ISTCHN,NUMCHS,IOFF,INPDEV)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      A UTILITY SUB TO READ DATA IN SPECIFIED CHANNELS FROM A
C      PREVIOUSLY OPENED STANDARD SPECTRAL FILE, CONVERT TO REAL NUMBERS,
C      AND RETURN THEM TO CALLER'S BUFFER. CALL WITH ADDRESS OF DESTINATION
C      BUFFER, STARTING CHANNEL, # OF CHANS, AND DEVICE NUMBER ASSIGNED
C      TO THE SOURCE FILE. CALLING PROGRAM IS RESPONSIBLE FOR ADEQUATE SIZE
C      OF DESTINATION BUFFER & FOR IDENTIFICATION OF SOURCE FILE.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER*2 INPDEV,ISTCHN,NWDCT,NDAT(2),ODAT(2),IOFF
      INTEGER*4 DATA,Y
      DIMENSION DAT(*)
      EQUIVALENCE (DATA,ODAT(1)),(Y,NDAT(1))
C
      ISTCHN = ISTCHN + IOFF
      LSTCHN = ISTCHN + NUMCHS
C
      K = 0
      DO 15 II = ISTCHN,LSTCHN
      READ(INPDEV,REC=II)DATA
      NDAT(1) = ODAT(2)
      NDAT(2) = ODAT(1)
      K = K + 1
15    DAT(K) = FLOAT(Y)
C
      RETURN
      END

```

```
      SUBROUTINE RDORTC(DATA,IOCH1)
C
C Subroutine to read ORTEC 92X Spectrum Master spectral data files.
C Data stored in 32-byte records, corresponding to 8 channels of data.
C
      INTEGER*4 SPCIN(32)
      DIMENSION DATA(*)
C
C Read 4096 channels skipping the first record
C
      JJ = 0
      DO 100 J = 2,513
      READ (IOCH1,REC=J)(SPCIN(K),K=1,8)
      DO 50 KK = 1,8
      JJ = JJ + 1
      DATA(JJ)= FLOAT(SPCIN(KK))
50
100  CONTINUE
      RETURN
      END
```

```

SUBROUTINE SEC2J (SECS,YR,JDAY)
*****
C
C
C          SEC2J
C          -----
C
C *****
C
C   Developed by :   W.M. Buckley
C                   LLNL ASD
C
C   Developed for : LLNL NCD
C
C *****
C
C   Title       :   sec to_jday
C   Revision    :   1.00
C   Author      :   W.M. Buckley
C   Date        :   7-July-1989
C   Language    :   MS FORTRAN 4.10
C
C *****
C
C   Abstract    :   This routine converts Canberra S-100 time format ( seconds
C                   since 00:00:00 1970 GMT ) to year and julian day with
C                   fractional day
C
C *****
C
C   Arguments:  secs   I*4   (I) S-100 seconds
C               yr     I*2   (O) year
C               jday   R*4   (O) julian day with fractional day
C
C *****
C
C   Compiler Directives -- Metacommands (special commands)
C *****
C
C   Routines called :
C
C   Routines called by :
C
C   Libraries used :
C
C *****
C
C   Revision History (Include date, revision ID, and details)
C *****
C
C   DATE      NAME      COMMENTS
C   -----
C
C   7-7-89    WMB       1st release
C
C *****
C
C   COMMON and Local Variable Declarations
C *****
C
C   INTEGER*4   secs
C   INTEGER*4   sec_yr
C   INTEGER*4   sec_day
C   INTEGER*4   adj_secs
C   INTEGER*4   num_yrs
C   INTEGER*4   num_lyrs
C   INTEGER*2   yr
C   INTEGER*4   day

```

```

INTEGER*4   sec
INTEGER*4   cvt_gmt
INTEGER*4   rem_sec

REAL*4      frac
REAL*4      jday

sec_yr = 3.1536E+07
sec_day = 8.6400E+04
cvt_gmt = 2.5200E+04

C ...estimate years elapsed
num_yrs = secs / sec_yr

C ...number of leap years passed
num_lyrs = num_yrs / 4

C ...adjust, if this is a leap year
IF(MOD(num_yrs,4).EQ.0)THEN
    num_lyrs = num_lyrs - 1
ENDIF
adj_secs = num_lyrs * sec_day

C ...recalculate number of years
num_yrs = (secs - adj_secs) / sec_yr

C ...calculate year
yr = 1970 + num_yrs

C ...calculate day
rem_sec = MOD((secs-adj_secs),sec_yr)
day = rem_sec / sec_day
sec = MOD(rem_sec,sec_day)

C ...adjust for GMT
sec = sec - cvt_gmt
IF(sec.LT.0)THEN
    day = day - 1
    sec = sec + sec_day
IF(day.LT.0)THEN
    yr = yr - 1
    day = day + 365
ENDIF

ENDIF
frac = FLOAT(sec) / FLOAT(sec_day)
jday = FLOAT(day) + frac

RETURN
END

```

```

C
C      SUBROUTINE SPLIT (A,XM,IE)
C*****
C              SPLIT
C              ====
C*****
C      Author: Ray Gunnink
C      Date: 3/1/89
C*****
C      This subroutine is used to separate the matissa and the exponent
C      of a number and return this values to the calling routine.
C*****
C      INTEGER IE
C      LOGICAL SIGN
C
C      XM = 0.
C      IE = 0
C      IF (A .EQ. 0) RETURN
C      SIGN = .FALSE.
C      IF (A .LT. 0) THEN
C        SIGN = .TRUE.
C        A = -A
C      ENDIF
C      IE = INT(ALOG10(A))
C      XM = 10**(ALOG10(A)-IE)
C      IF (SIGN) XM = -XM
C      RETURN
C      END

```

```

C*****
C      Author :Ray gunnink
C      Version 1.0
C      Date   :19-Jan-78
C*****
C      SPRAM
C
C      THIS SUBROUTINE CALCULATES SOME OF THE PARAMETERS USED TO
C      CHARACTERIZE THE SHAPE OF A PEAK
C
C      SUBROUTINE SPRAM (EN,GAIN,SHAPC,SG,ALFA,F12M,EXP1)
C
C      DIMENSION SHAPC(*)
C
C      GSQ = GAIN * GAIN
C      SG = (SHAPC(1) + EN * SHAPC(2)) / GSQ + 0.462
C      ALFA = -2.7726 / SG
C      F12M = SQRT (SG)
C      EXP1 = EXP (EN * SHAPC(4) + SHAPC(3))
C      RETURN
C      END

```


SUBROUTINE STRIP1 (NS,NE,PKPOS,PKHT,ALFA,EXP1,EXP2,WL,Y)

STRIP1

=====

Developed for : Safeguards Technology Program
Nuclear Chemistry Division
Lawrence Livermore National Laboratory
Livermore, CA. 94550

Title : STRIP1
Revision : 1.0
Author : R. Gunnink
Date : Revised for the IBM PC by Russell Fleming
Date : 10/15/86
Language : Microsoft Fortran Ver. 4.01

Abstract :
This subroutine is used to strip a peak or part of a peak from
a continuum. STRIP1 can be used on high-energy x-rays.

Arguments :

INPUT

NS = Starting position in data array from which peak is to be
stripped
NE = Ending position in data array
PKPOS = Position in array of peak to be stripped
PKHT = Peak height
ALFA = Full-width-half-maximum parameter of peak to be stripped
EXP1 = Short-term tail amplitude of peak
EXP2 = Short-term tail slope of peak
WL = Lorentzian width of peak, if peak is an x-ray
Y = Data array containing data from which peak is stripped

Routines
Called : None

Libraries
Used : None

Revision History (Include date,revision I.D., and details)

Version: 2.1
Author: R. Gunnink
Date: 20-JAN-78

COMMON and variable declarations

DIMENSION Y(*)

variable initialization declarations

format declarations

IF (WL) 1,2,1

1 CX = SQRT (-ALFA)
GAMA = WL * CX
C1 = 1. - .5642 * GAMA
C2 = .15915 * GAMA
C3 = .25 * GAMA * GAMA
C4 = .6366 * GAMA
RT = C1 + C3

2 DO 10 I = NS,NE

XI = I
DX = XI - PKPOS
DDX = DX * DX
ADX = ALFA * DDX
FXX = EXP (ADX)
FX = FXX

3 IF (DX) 3,4,4
EXP2 = EXP (EXP2 * DX)
FXX = FXX + EXP1 * EXP2 * (1. - EXP (.4 * ADX))
DX = -DX

4 IF (WL) 6,9,6

6 CC = CX * DX
CALL BWF (CC)
FXX = FXX - FX + (C1 - C2 * ADX + C3 * (1.+2.*ADX)) * FX + C4 * CC
FXX = FXX / RT
9 Y(I) = Y(I) - PKHT * FXX

10 CONTINUE
RETURN
END

```
C
C      SUBROUTINE TRNSFR (NDPTS,YDATA,Y)
C-----
C
C      This routine transfers data from one array to another.
C
C      DIMENSION YDATA(*),Y(*)
C-----
C
C      DO 10 I = 1,NDPTS
10    Y(I) = YDATA(I)
C
C      RETURN
C      END
```