ISTITUTO NAZIONALE FISICA NUCLEARE

INFN/BE - 78/3
23 Gennaio 1978

U. Abbondanno, A. Boiti, F. Demanins and M.R. Malisan:

DUMAN - A FORTRAN PROGRAMME FOR THE AUTOMATICAL ANALYSIS OF GAMMA-RAY AND CHARGED-PARTICLE COMPLEX SPECTRA FROM SOLID STATE DETECTORS

SERVIZIO RIPRODUZIONE DELLA
SEZIONE DI TRIESTE DELL'INFN

DUMAN - A FORTRAN PROGRAMME FOR THE AUTOMATICAL ANALYSIS OF GAMMA-RAY AND CHARGED-PARTICLE COMPLEX SPECTRA FROM SOLID STATE DETECTORS

U. Abbondanno, A. Boiti, F. Demanins and M.R. Malisan

Istituto di Fisica dell'Universita', Trieste
Istituto Nazionale di Fisica Nucleare, Sezione di Trieste

1. - INTRODUCTION

In the analysis of spectra of charged-particles or gamma-rays recorded with a solid state detector, the shapes of the peaks often can be assumed to be pure Gaussian functions superimposed to a slowly varying background. In such cases the use of a non-iterative method presents some remarkable advantages with respect to other methods. In fact, it requires a very small computer time if compared to the usual iterative search techniques, a relatively little computer memory, and it doesn't need an initial estimate of the parameters of the peaks, like center and full-width at half-maximum (FWHM).

Non-iterative methods (1) have been succesfully applied both to the analysis of gamma ray spectra from NaI(Tl) scintillation detectors (2) and to the analysis of charged-particle and gamma-ray spectra from solid state detectors (3). Recently, a method of this kind was developed for the automatical analysis of spectra (4), in which the second derivative of the spectrum (5) is used in order to recognize the presence of a peak, and a non-iterative fitting technique is used to extract the physically interesting parameters.

This analysis is performed by the computer programme DUMAN, written in FORTRAN IV for the HP 2100S computer of the Istituto di Fisica dell'Università di Trieste. In this report we describe the structure of the programme (section 2) and the input data (section 3). The complete listing is given in the Appendix. Since the mathematical apparatus and the flowing of the programme have been presented in detail elsewhere (4) for what concerns both the peak finding procedure (5) and the fitting technique (3), in the present report we shall simply refer to the quantities defined in the above mentioned papers.

2. - DESCRIPTION OF THE PROGRAMME

The programme consists of a main programme and of some subroutines in order to achieve a complete modularity.

Main programme:

DUMAN - It controls the whole flow of the analysis and calls the subroutines involved in the calculations.

Subroutines:

TITLE and DATA - They write the name of the programme, read the input data, report them on teleprinter.

RIDIN - It reads the spectrum from punched paper tape according to the reading technique described in ref.(4).

COOK - It smooths the spectrum for the determination of the background intervals.

CIGEI and SECDF - They calculate the second difference S_i and the standard deviation F_i of the spectrum, together with the weighting coefficients C_{ij} described in refs.(4) and (5).

HILLS - It identifies the peaks from the analysis of the mutual behaviour of S_i and F_i.

FLAT - It determines the zones in the spectrum through which the background has to be calculated.

SOTFO - It subtracts the calculated background from the spectrum.

TOP - It determines the points to be used in the non-iterative fit.

FIT2R - It performs the non-iterative fit and determines the center and the standard deviation of the peaks.

PARAM - It writes the parameters of the peaks.

BAKL - It subtracts the calculated peaks from the spectrum.

FIT1R - It determines the area of the peaks.

3. - DESCRIPTION OF THE INPUT DATA

In the version of the programme presented in this report, five input cards are necessary for the analysis of a spectrum.

CARD 1 (FORMAT I5) contains KFORM

KFORM is a parameter concerning the FORMAT (see Appendix) of the punched paper tape to be analyzed.

CARD 2 (FORMAT 36A2) contains ITITLE

ITITLE is a label for the spectrum

CARD 3 (FORMAT 16I5) contains LU, NZETA, NPASSO, NSM, I53, NHIL, NSING, NDOUB

LU is the number of the logic unit for a detailed printing of the results (such as the spectrum N_i, the calculated background B_i, the spectrum after background subtraction, the second difference S_i and its standard deviation F_i) which allows one to follow the analysis step by step. LU is usually taken equal to the dummy output, and only the parameters of the analyzed peaks are written in the routine operating of the programme. NZETA is the number of the smoothing iterations in the calculation of the second difference, as explained in ref.(5).

NPASSO is the smoothing step of the NZETA iterations.

NSM is the smoothing window used in the subroutine COOK for the smoothing of the spectrum in the search of the background zones.

I53 is the minimum distance between the points I_3 and I_5 (4) that is acceptable for the identification of a peak.

NHIL is the maximum acceptable distance between I_3 (or I_5) and the nearest channel i for which $S_i > F_i$.

NSING, NDOUB. If a peak is single and the channel nearest to the center is I, the non-iterative fit is performed from the channel (I-NSING) to tne channel (I+NSING). If a peak is overlapped to another one, the fit is performed from the channel (I±NSING) (the sign plus or minus is taken according to wether the contaminant peak lies, respectively, to the left or to the right of the considered peak) to the channel (I∓NDOUB).

CARD 4 (FORMAT 16I5) contains LENF, NPART, NARR, LSG, IOVF

LENF is the length of the spectrum

NPART, NARR identify the first and the last channel of the zone of the spectrum that has to be analyzed.

LSG is the length of the segment of the spectrum considered in the reading routine RIDIN.

IOVF is the number of channels of the overlapping zone between two consecutive segments.

CARD 5 (FORMAT 3 (F10.3)) contains ZERO, CONV, FACT

ZERO is the value (in keV) of the energy scale at channel zero.

CONV is the conversion factor of the scale (keV/channel).

FACT is the percentage of the maximum acceptable asymmetry between the positive maxima M_1 and M_2 of the smoothed second difference S_i: if the condition

$$(1\text{-FACT}) \leq (M_1/M_2) \leq (1\text{+FACT})$$

is fulfilled, the peak is analyzed as single; in the contrary case, the peak is analyzed as multiple.

After the complete analysis of a spectrum, the programme begins the analysis of another spectrum, by reading CARD 1; a value of KFORM=99 causes the programme to stop.

REFERENCES

[1] W. Zimmermann, Rev. Sci. Instr. 32, 1063 (1961).

[2] T. Mukoyama, Nucl. Instr. and Meth. 125, 289 (1975).

[3] U. Abbondanno, A. Boiti and F. Demanins, Nucl. Instr. and Meth., 142, 605 (1977)

[4] U. Abbondanno, A. Boiti, F. Demanins and M.R. Malisan, Nucl. Instr. and Meth., in press.

[5] M.A. Mariscotti, Nucl. Instr. and Meth. 50, 309 (1967).

APPENDIX

```
FTN4,L
      PROGRAM DUMAN
      DIMENSION SMU(560),Q(200)
      INTEGER C(45)
      COMMON AN(560),S(560),F(560)
      COMMON BACK(200),SPUL(200),ESP(200)
      COMMON ROW1(200),ROW2(200)
      CALL TITLE
306   READ(7,105) KFORM
105   FORMAT (4I5)
      IF(KFORM.EQ.99) GO TO 999
      CALL DATA(LU,NZETA,NPASSO,LENF,NPART,NARR,LSG,I53,NHIL,
     1IOVF,NSING,NDOUB,NSM,ZERO,CONV,FACT,KNOK,NSG,LSS,LUS)
      IF(NPART.EQ.1) GO TO 303
      NPAR =NPART-1
      CALL RIDIN(.TRUE.,KFORM,NPAR,IK)
303   NPE=0
      IF (KNOK.EQ.0) GO TO 306
      NZONE=0
      DO 202 LAV=1,NSG
      IF(LAV.EQ.NSG) LSS=LUS
      IF(LAV.GT.1) GO TO 304
      NOOR=NPART+LSS-1
      NPAR = LSS
      IK=1
      GO TO 305
304   NPART=NOOR-IOVF+1
      NOOR=NPART+LSS-1
      NPAR =LSS-IOVF
      IK=IOVF+1
      IF(IK.EQ.1) GO TO 305
      DO 204 I=1,IOVF
204   AN(I)=AN(I+LSG)
305   CALL RIDIN(.FALSE.,KFORM,NPAR,IK)
      WRITE (6,118) NPART,NOOR
118   FORMAT(5X,"SEGMENT FROM CHANNEL",I5," TO CHANNEL",I5)
      WRITE(6,120)
120   FORMAT(5X,"****************************************",/)
      CALL COOK (LSS,NSM,SMU)
      NPAR = NPART-1
      NLB1=1
      NHB1=1
      NHB2=1
205   M=NZETA*NPASSO
      MPIU=M+1
      MOOR=M+2
      CALL CIGEI(NZETA,NPASSO,M,C)
      CALL SECDF(LSS,0,MOOR,C)
209   CALL HILLS(NHB2,LSS,I53,MI3,MI5,NED,NHIL)
      IF(NED.EQ.1) GO TO 202
      MIO3 = MI3+NPAR
      MIO5 = MI5+NPAR
      CALL FLAT(MI3,MI5,NLB1,NLB2,NHB1,NHB2,LSS,SMOD1,SMOD2,MPIU,IV,SMU)
      IF (IV.EQ.0) GO TO 801
      NHB2 = MI5+1
      GO TO 209
801   NLOB1=NLB1+NPAR
      NLOB2=NLB2+NPAR
      NLOB3=NHB1+NPAR
      NLOB4=NHB2+NPAR
      NZONE=NZONE+1
      NEP=0
      WRITE(6,121) NZONE
121   FORMAT(5X,"ANALYSIS OF THE ZONE NUMBER",I5)
      WRITE(6,122)
122   FORMAT(5X,"*******************************",/)
      WRITE(6,123) NLOB1,NLOB2
123   FORMAT(5X,"LOW ENERGY BACKGROUND CALCULATED FROM CHANNEL",I6,"   TO
     1 CHANNEL",I6)
      WRITE(6,124)NLOB3,NLOB4
124   FORMAT(5X,"HIGH ENERGY BACKGROUND CALCULATED FROM CHANNEL",I6,"   T
     1O CHANNEL",I6,/)
```

```
      MU = NHB2-NLB1+1
      MOM1=NLB1-MPIU
      IF (MOM1.LT.1) MOM1 = 1
      MOM2=NHB2+MPIU
      IF (MOM2.GT.LSS) MOM2 = LSS
      MOM = MOM2-MOM1+1
      IF (NLB1.LE.MPIU) MPIU = NLB1-1
      MUM = MU+MPIU
      CALL SOTFO(NLB1,NLB2,NHB1,NHB2,SMOD1,SMOD2,MPIU,MOM1,MOM2,MOM)
 777  CALL SECDF(MOM,MOM1,MOOR,C)
      WRITE(LU,131)
 131  FORMAT(1X,"CHANNEL",8X,"AN(I)",8X,"BACK(I)",8X,"SPUL(I)",8X,"S(I)"
     1,8X,"F(I)",/)
      K = MPIU
      DO 234 J=NLB1,NHB2
      K=K+1
      L=J+NPAR
 234  WRITE(LU ,132)(L,AN(J),BACK(K),SPUL(K),S(K),F(K))
 132  FORMAT(3X,I5,3X,5(F10.5,5X))
      NK = MOOR
 409  CALL HILLS(NK,MUM,I53,MI3,MI5,NED,NHIL)
      IF(NED.EQ.0) GO TO 410
      NHB2=NHB1
      GO TO 205
 410  CONTINUE
      WRITE (LU,105) MI3,MI5
      CALL TOP(MI3,MI5,FACT,NSING,NDOUB,I1,I2,ISTAR)
      DO 709 I=I1,I2
      ROW1(I)=1.
      ROW2(I)=I
      X =(SPUL(I-1)/SPUL(I+1))
 709  ESP(I) = ALOG(X)
      CALL FIT2R(I1,I2,AK1,AK2,ROW1,ROW2,ESP)
      IF(AK2.GT.0.) GO TO 406
      NK = MI5
      GO TO 409
 406  SIGMA=SQRT(2./AK2)
      CENTR =-AK1/AK2
      ENF=2.*SIGMA*CONV*1.17741
      CONTR = CENTR+NLOB1-1-MPIU
      IF (CONTR.GE.FLOAT(NLOB1)) GO TO 756
      ISTAR = 0
      GO TO 755
 756  EN=ZERO+CONV*CONTR
      DO 716 I=I1,I2
      X=FLOAT(I)
 716  ROW1(I)=(EXP(-(X-CENTR)**2/(2*SIGMA**2)))/(2.506628275*SIGMA)
      CALL FIT1R(I1,I2,AK1,ROW1,SPUL)
      IB1=ISTAR-IFIX(2.3*SIGMA)
      IB2=ISTAR+IFIX(2.3*SIGMA)
      IF (IB2.GT.MUM+MPIU) GO TO 755
      SUMB=0.
      DO 751 J1=IB1,IB2
 751  SUMB=SUMB+SPUL(J1)
      SBAC=0.
      DO 752 J2=IB1,IB2
 752  SBAC=SBAC+BACK(J2)
      FOND=SQRT(SUMB+SBAC)
      NPE=NPE+1
      NEP=NEP+1
      CALL PARAM(NPE,CONTR,EN,ENF,AK1,FOND,SBAC,NEP)
 755  CONTINUE
      CALL BAKL(MUM,ISTAR,I1,I2,AK1,CENTR,SIGMA,SPUL)
      GO TO 777
 202  CONTINUE
      IF(NOOR.EQ.LENF) GO TO 306
      NPAR =LENF-NOOR
      CALL RIDIN(.TRUE.,KFORM,NPAR,IK)
      GO TO 306
 999  WRITE(6,998)
 998  FORMAT(20X,"DUMAN IS A FASCINATING BLACK DOG",///)
      STOP
      END

** NO ERRORS**    PROGRAM = 02787       COMMON = 05360
```

```
      SUBROUTINE TITLE
      WRITE(6,101)
101   FORMAT("1",///)
      WRITE(6,102)
102   FORMAT(40X,"**********************************")
      WRITE(6,103)
103   FORMAT(40X,"*                                *")
      WRITE(6,103)
      WRITE(6,104)
104   FORMAT(40X,"*   P R O G R A M   D U M A N    *")
      WRITE(6,103)
      WRITE(6,103)
      WRITE(6,102)
      WRITE(6,101)
      RETURN
      END

** NO ERRORS**    PROGRAM = 00136      COMMON = 00000


      SUBROUTINE DATA(LU,NZETA,NPASSO,LENF,NPART,NARR,LSG,I53,NHIL,
     1IOVF,NSING,NDOUB,NSM,ZERO,CONV,FACT,KNOK,NSG,LSS,LUS)
      DIMENSION ITITLE(36)
      DO 201 LS=1,36
201   ITITLE(LS)=2H
      READ(7,106) (ITITLE(LS),LS=1,36)
106   FORMAT(36A2)
      WRITE(6,107) (ITITLE(LS),LS=1,36)
107   FORMAT(30X,36A2)
      WRITE(6,108)
108   FORMAT("1",/)
302   READ(7,105)LU,NZETA,NPASSO,NSM,I53,NHIL,NSING,NDOUB
      READ(7,105)LENF,NPART,NARR,LSG,IOVF
105   FORMAT(16I5)
      IF (NZETA.GT.7) NZETA = 7
      IF (NPASSO.GT.3) NPASSO = 3
      NPART=(NPART/8)*8+1
      LSG=(LSG/8)*8
      NARR=(NARR/8)*8
      IOVF=(IOVF/8+1)*8
      READ(7,109)           ZERO,CONV,FACT
109   FORMAT(8F10.3)
      WRITE(6,110) LENF
110   FORMAT(5X,"SPECTRUM OF ",I5," CHANNELS",/)
      WRITE(6,111) LSG
111   FORMAT(5X,"SUBGROUPS OF  ",I5," CHANNELS")
      WRITE(6,112) IOVF
112   FORMAT(5X,"OVERLAP BETWEEN THE GROUPS OF",I4," CHANNELS")
      WRITE(6,113) NPART,NARR
113   FORMAT(5X,"ANALYSIS FROM CHANNEL ",I5," TO CHANNEL   ",I5,/)
      WRITE(6,115) NZETA,NPASSO
115   FORMAT(5X,I1,"  SMOOTHING ITERATIONS WITH A STEP OF ",I1,/)
      WRITE(6,116) ZERO,CONV
116   FORMAT(5X,"ENERGY SCALE: ",F10.4," KEV=0 ",F10.4," KEV/CHANNEL")
      WRITE (6,118) NSM,I53,NHIL
118   FORMAT (5X,"NSM =",I2," I5MI3 =",I2," NHILLS =",I2,//)
      KNOK=NARR-NPART+1
      NSG=(KNOK-1)/LSG+1
      LSS=LSG+IOVF
      IF(LSS.LE.560) GO TO 301
      WRITE(6,117)
117   FORMAT("        TOO LONG INTERVAL. RE-ENTER DATA    ")
      PAUSE
      GO TO 302
301   LUS=KNOK-(NSG-1)*LSG
      RETURN
      END

** NO ERRORS**    PROGRAM = 00570      COMMON = 00000
```

```
      SUBROUTINE RIDIN(GARB,KFORM,KANAL,IK)
      DIMENSION AA(8)
      COMMON AN(560),S(560),F(560)
      LOGICAL GARB
      NOTT = KANAL/8
      DO 1 II=1,NOTT
      IF (KFORM.EQ.0) GO TO 2
      IF (KFORM.EQ.1) GO TO 5
      GO TO 3
 5    READ (5,9) (AA(I),I=1,8)
      GO TO 3
 2    READ (5,8) (AA(I),I=1,8)
 3    IF (GARB) GO TO 1
      DO 4 K=1,8
      AN(IK) = AA(K)
 4    IK = IK+1
 1    CONTINUE
 8    FORMAT (7X,8(F6.0,1X))
 9    FORMAT (8(F6.0,1X))
      RETURN
      END

** NO ERRORS**   PROGRAM = 00163      COMMON = 03360


      SUBROUTINE COOK (LSS,NSM,SMU)
      DIMENSION SMU(1)
      COMMON AN(560)
      IF (NSM.EQ.0) RETURN
      D = 2*NSM+1
      L = LSS-NSM
      IN = NSM+1
      DO 1 I=IN,L
      LI = I-NSM
      LS = I+NSM
      S = 0.
      DO 2 J=LI,LS
 2    S = S+AN(J)
 1    SMU(I) = S/D
      RETURN
      END

** NO ERRORS**   PROGRAM = 00102      COMMON = 01120


      SUBROUTINE CIGEI(NZETA,NPASSO,M,C)
      INTEGER C(1),B(45),FUC(45)
      DO 100 K1 = 1,45
  100 C(K1) = 0
      C(M+1) = 1
      C(M+2) = -2
      C(M+3) = 1
      IF(NZETA .EQ.0) GO TO 302
      NMAX=2*M+3
      DO 301 L = 1,NZETA
      DO 300 IL = 1,NMAX
      K3 = IL-NPASSO
      IF (K3.LE.0) K3 = 1
      K4 = IL + NPASSO
      IF (K4.GT.NMAX) K4 = NMAX
      B(IL) = 0
      DO 400 KO = K3,K4
  400 B(IL) = B(IL) + C(KO)
 300  FUC(IL)=B(IL)
      DO 99 N = 1,NMAX
 99   C(N)=FUC(N)
  301 CONTINUE
  302 RETURN
      END

** NO ERRORS**   PROGRAM = 00266      COMMON = 00000
```

```
          SUBROUTINE SECDF(NEST,KRUK,IO,C)
          INTEGER C(1)
          COMMON AN(560),S(560),F(560)
          COMMON BACK(200),SPUL(200),ESP(200)
          DO2 I=1,200
          S(I) = 0.
    2     F(I) = 0.
          DO 12 I=IO,NEST
          JO=I-IO+1
          JF=I+IO-1
          IF (JF.GT.NEST) GO TO 11
          SUMS=0.
          SUMF=0.
          DO 22 L=JO,JF
          K=L-I+IO
          IF(KRUK.NE.0) GO TO 101
          DSUMS= C(K)*AN(L)
          DSUMF =   (C(K)**2)*AN(L)
          GO TO 20
  101     LIP=L+KRUK-1
          DSUMS= C(K)*SPUL(L)
          DSUMF=(C(K)**2)*AN(LIP)
   20     SUMS=SUMS+DSUMS
   22     SUMF=SUMF+DSUMF
          S(I)=SUMS
   12     F(I)=SQRT(SUMF)
   11     RETURN
          END
```

```
**  NO ERRORS**     PROGRAM = 00236       COMMON = 04560
```

```
          SUBROUTINE HILLS (IP,IA,I5MI3,I3,I5,NED,NHIL)
          COMMON AN(560),S(560),F(560)
          NED = 0
          KU = IA-1
          I5 = IP
   11     KP = I5+1
          IF (KP.GT.KU) GO TO 7
          DO 1 I=KP,KU
          IF (S(I).LT.-F(I)) GO TO 2
    1     CONTINUE
    7     NED = 1
          RETURN
    2     I3 = I
          KS = I3+1
          DO 3 I=KS,IA
          IF (S(I).GE.-F(I)) GO TO 4
    3     CONTINUE
    4     I5 = I-1
          IF (I5-I3.LT.I5MI3) GO TO 11
          I = I3
    6     I = I-1
          IF (I.LT.IP) GO TO 5
          IF (S(I).LT.F(I)) GO TO 6
          IF (I3-I.GT.NHIL) GO TO 5
          RETURN
    5     I = I5
    8     I = I+1
          IF (I.GT.IA) GO TO 7
          IF (S(I).LT.F(I)) GO TO 8
          IF (I-I5.GT.NHIL) GO TO 11
          I6 = I
          RETURN
          END
```

```
**  NO ERRORS**     PROGRAM = 00208       COMMON = 03360
```

```
          SUBROUTINE FLAT(MIM3,MIM5,NLB1,NLB2,NHB1,NHB2,LSS,SMOD1,SMOD2,MPIU
         1,IV,AN)
          DIMENSION AN(1)
          NUM=10
          NEM=10
          IV = 0
    10    NUM=NUM-1
          NAM=NUM
    1     NAM=NAM-1
          IF(NAM.GE.4) GO TO 11
          IV = 1
          GO TO 6
    11    NOB1 = MIM3-NAM-4
          NOB2=MIM3-4
          IF(NOB1.LT.NHB1) GO TO 1
    4     NOB1=NOB1-1
          NOB2=NOB2-1
          IF(NOB1.LT.NHB1) GO TO 1
          SMED1=0.
          DO 2 IFOT=NOB1,NOB2
    2     SMED1=SMED1+AN(IFOT)
          SMOD1=SMED1/(NAM+1)
          SM = SMOD1+3.
          ST = 0.15*ALOG(SM)*SQRT(SM)
          SI=SMOD1-ST
          SS=SMOD1+ST
          DO 3 KFOT=NOB1,NOB2
          IF(AN(KFOT).LT.SI.OR.AN(KFOT).GT.SS) GO TO 4
    3     CONTINUE
          NEM=NEM-1
          NOM=NEM
    5     NOM=NOM-1
          IF(NOM.GT.4) GO TO 12
          IV = 1
          GO TO 6
    12    NOB3=MIM5+4
          NOB4=MIM5+4+NOM
          IF(NOB4.GT.LSS) GO TO 5
    8     NOB3=NOB3+1
          NOB4=NOB4+1
          IF(NOB4.GT.LSS) GO TO 5
          SMED2=0.
          DO 7 LFOT=NOB3,NOB4
    7     SMED2=SMED2+AN(LFOT)
          SMOD2=SMED2/(NOM+1)
          SM = SMOD2+3.
          ST = 0.15*ALOG(SM)*SQRT(SM)
          SI=SMOD2-ST
          SS=SMOD2+ST
          DO 9 NFOT=NOB3,NOB4
          IF(AN(NFOT).LE.SI.OR.AN(NFOT).GE.SS) GO TO 8
    9     CONTINUE
          IF((NOB4-NOB1).GE.(199-2*MPIU)) GO TO 10
          NLB1=NOB1
          NLB2=NOB2
          NHB1=NOB3
          NHB2=NOB4
    6     CONTINUE
          RETURN
          END
```

```
**  NO ERRORS**     PROGRAM = 00387       COMMON = 00000
```

```
          SUBROUTINE SOTFO(L1,L2,N1,N2,S1,S2,MF,M1,M2,MM)
          COMMON AN(560),S(560),F(560)
          COMMON BACK(200),SFUL(200),ESP(200)
          COMMON ROW1(200),ROW2(200)
          IX1 = (L1+L2)/2
          IX2 = (N1+N2)/2
          CA = (S2-S1)/(IX2-IX1)
          K = 0
          DO 1 J=M1,M2
          K = K+1
```

```
1      BACK(K) = S1+CA*(J-IX1)
       DO 2 K=1,MM
       ROW1(K)=0.
       ROW2(K)=0.
2      ESP(K)=0.
       IK = L1-MP-1
       DO 3 K=L1,L2
       I = K-IK
       ROW1(I)=1.
       ROW2(I)=I
3      ESP(I)=AN(K)
       LI = L2+1
       LS = N1-1
       DO 4 K=LI,LS
       I = K-IK
       IF (AN(K).GT.BACK(I)) GO TO 4
       ROW1(I)=1.
       ROW2(I)=I
       ESP(I)=AN(K)
4      CONTINUE
       DO 5 K=N1,N2
       I = K-IK
       ROW1(I)=1.
       ROW2(I)=I
5      ESP(I)=AN(K)
       CALL FIT2R(1,MM,AK1,AK2,ROW1,ROW2,ESP)
       K=0
       DO 6 J=M1,M2
       K=K+1
       BACK(K)=AK1+K*AK2
       SPUL(K)=AN(J)-BACK(K)
       IF(SPUL(K).LE.0.) SPUL(K)=0.5
6      CONTINUE
       RETURN
       END

**  NO ERRORS**     PROGRAM = 00401        COMMON = 05360
```

```
       SUBROUTINE TOP(MI3,MI5,FACT,NSING,NDOUB,I1,I2,ISTAR)
       COMMON AN(560),S(560),F(560)
       COMMON BACK(200),SPUL(200),ESP(200)
       DO 701 I=MI3,MI5
       IF(S(I+1).GE.S(I)) GO TO 702
701    CONTINUE
702    ISTAR=I
       J=ISTAR
704    J=J-1
       IF (J.EQ.1) GO TO 703
       IF(S(J-1).LE.S(J)) GO TO 703
       GO TO 704
703    IMAX1=J
       J=ISTAR
705    J=J+1
       IF (J.EQ.200) GO TO 706
       IF(S(J+1).LE.S(J)) GO TO 706
       GO TO 705
706    IMAX2=J
       RAP=S(IMAX1)/S(IMAX2)
       IF(RAP.GE.(1+FACT).OR.RAP.LE.(1-FACT)) GO TO 707
       I1=ISTAR-NSING
       I2=ISTAR+NSING
       RETURN
707    IF(RAP.GE.(1+FACT)) GO TO 711
       I1=ISTAR-NDOUB
       I2=ISTAR+NSING
       RETURN
711    I1=ISTAR-NSING
       I2=ISTAR+NDOUB
       RETURN
       END

**  NO ERRORS**     PROGRAM = 00220        COMMON = 04560
```

```
      SUBROUTINE FIT2R(I1,I2,AK1,AK2,ROW1,ROW2,Q)
      DIMENSION ROW1(1),ROW2(1),Q(1)
      AK1=0.
      AK2=0.
      C1=0.
      C2=0.
      C3=0.
      A=0.
      B=0.
      DO 1 I=I1,I2
      C1=C1+ROW1(I)**2
      C2=C2+ROW1(I)*ROW2(I)
      C3=C3+ROW2(I)**2
      A=A+ROW1(I)*Q(I)
1     B=B+ROW2(I)*Q(I)
      DELTA=C1*C3-C2**2
      DX1=A*C3-B*C2
      DX2=B*C1-A*C2
      AK1=DX1/DELTA
      AK2=DX2/DELTA
      RETURN
      END

**  NO ERRORS**     PROGRAM = 00223        COMMON = 00000


      SUBROUTINE PARAM(NPE,CONTER,EN,ENF,A,FOND,SBAC,NEP)
      IF(NEP.EQ.1) WRITE(6,712)
712   FORMAT(7X,"PEAK",2X,"CENTER",4X,"ENERGY",7X,"FWHM",4X,"AREA",6X,"E
     1RROR",4X,"BACKGROUND",/)
      WRITE(6,713) NPE,CONTER,EN,ENF,A,FOND,SBAC
713   FORMAT(5X,I5,6(F10.2),/)
      RETURN
      END

**  NO ERRORS**     PROGRAM = 00103        COMMON = 00000


      SUBROUTINE BAKL(MUM,ISTAR,I1,I2,A,CENTR,SIGMA,SPUL)
      DIMENSION SPUL(1)
      IK1=I1+1
      IK2=I2-1
      IF(ISTAR.EQ.0) GO TO 741
      SKUK=SQRT(SPUL(ISTAR))
      B = 2*SIGMA**2
      C = 2.506628275*SIGMA
      DO 720 I=1,MUM
      X=FLOAT(I)
      X = A*EXP(-(X-CENTR)**2/B)/C
      SPUL(I)=SPUL(I)-X
      IF(SPUL(I).LT.1.) SPUL(I)=1.
720   CONTINUE
      DO 740 KUK=IK1,IK2
      IF(SPUL(KUK).LE.SKUK) GO TO 741
740   CONTINUE
      RETURN
741   DO 743 KEK=IK1,IK2
743   SPUL(KEK)=1.
      RETURN
      END

**  NO ERRORS**     PROGRAM = 00192        COMMON = 00000
```

```
          SUBROUTINE FIT1R(I1,I2,AK1,ROW1,ESP)
          DIMENSION ROW1(1),ESP(1)
          AK1=0.
          C1=0.
          C2=0.
          DO 1 I=I1,I2
          C1=C1+ROW1(I)*ESP(I)
    1     C2=C2+ROW1(I)**2
          AK1=C1/C2
          RETURN
          END

**  NO ERRORS**     PROGRAM = 00082       COMMON = 00000
```