10 c.

ISTITUTO NAZIONALE FISICA NUCLEARE

INFN/BE - 79/3
10 Giugno 1979

U. Abbondanno, A. Boiti, A. Giletti and C. Tuniz

LIZA - A FAST COMPUTER CODE FOR AUTOMATIC ROUTINE ANALYSIS OF SOLID STATE DETECTORS SPECTRA

SERVIZIO RIPRODUZIONE DELLA
SEZIONE DI TRIESTE DELL'INFN

LIZA - A FAST COMPUTER CODE FOR AUTOMATIC ROUTINE ANALYSIS OF SOLID STATE DETECTORS SPECTRA

Ugo Abbondanno, Aldo Boiti, Angelo Giletti and Claudio Tuniz

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

ABSTRACT

A computer programme for fast routine automatic analysis of solid state detectors spectra is presented. The peak identification is carried out by means of a pattern recognition method acting directly on the raw spectrum, and the fitting by means of a non-iterative procedure. The programme is able to analyze both single and overlapping peaks by means of a successive stripping procedure. Main features of the programme are a remarkable working speed and a small core memory occupation that makes it suitable for small computers. Some examples of the analysis of γ-ray spectra from Ge(Li) detectors and of charged-particle spectra from Si detectors are shown.

1. - INTRODUCTION

In this report, we present a computer code particularly suitable to fast routine analysis of high-resolution γ-ray spectra from Ge(Li) detectors and charged-particle spectra from Si detectors, which in the last few years we have been gathering in ever increasing quantity in the course of a low-energy nuclear spectroscopy research programme. The large amount of collected data made the development of fast routine analysis techniques indispensable, and non-iterative methods [1,2] were chosen for their simplicity and velocity. Firstly, we used a semi-automatic method [3], then a fully automatic one [4], which combined the second derivative method [5] for peak identification with non-iterative techniques for the fitting.

The necessity of analyzing in a reasonable time batches of some hundreds of spectra on a relatively small computer (the HP 2100 S, 32 K word core memory of the Istituto di Fisica dell'Università di Trieste) showed subsequently that developing another programme even more economical in terms of computer time and memory requirements would be very useful. In fact, the only drawback of the method described in ref. [4] is the rather cumbersome and time-consuming procedure for the identification of peaks, which requires the re-calculation of the second difference of the spectrum after the analysis of each peak and its re-examination to find another one. In order to overcome this slowest step in [4] we adopted a much simpler identification procedure, based on a method of pattern recognition, which could be directly applied to the raw data. Using this new procedure we wrote in FORTRAN IV a computer programme named LIZA that determines position, FWHM and area of the peaks in a spectrum, and achieves a significant improvement in the velocity of the analysis with the following main features:

(1) it is suited to small computers, even without being split into linking segments (the code occupies no more than 8 K, 16-bit words)

(2) it requires very little initial information to perform the analysis

(3) it is satisfactorily accurate with respect both to peak location and peak area determination

(4) it can operate both in a completely automatic and very fast way or, alternatively, in an interactive way with provision for easy visual choice of interesting fitted peaks on a cathode ray tube (CRT) display, for subsequent printout of the desired results only.

In Section 2 the peak and background finding procedure will be presented, in Section 3 the complete structure of the programme will be described. Section 4 will show some examples of the application of this method to the analysis of γ-ray and charged-particle spectra. In Section 5 comments and conclusions will be reported. In Appendix A the complete listing of the interactive version of the programme LIZA is reported.

2. - PEAK AND BACKGROUND SEARCH METHOD

For the purposes of the analysis, the basic assumption we adopt is that a spectrum obtained with semiconductor detectors may be described as the sum of a number of more or less overlapping Gaussian functions, representing the peaks, superimposed on a relatively flat, slowly varying line representing the background.

Although other refined analytic functions, much more complex than a simple Gaussian, have been selected ([6-11]) better to fit the real peak shape (which, as is well known ([2,12-14]), is asymmetric) they all need iterative least-square fitting procedures which are at variance with our aim of economy in time of execution, core memory requirements and input information. Moreover, these methods often require the manual intervention of the user in the determination of the number of peaks to be considered in a particular fitting ([8,11,15]) and also require some initial estimates of the parameters to be optimized. Avoiding iterative least-square methods, then, implies the choice of the simple Gaussian function as the mathematical model of peak shape. This choice is supported

by the fact that the deviation from a pure Gaussian shape is not very significant in the case of low and moderate statistics, as we experienced in our spectra, and the accuracy in the position determination of a peak is not affected by the use of Gaussian shapes. The greater simplicity in the computer code and in the analysis makes up for the slightly less accurate determination of the peak area. Finally, as the number of channels defining each peak is usually small, because of the high resolution obtainable with semiconductor detectors, a straight line was, in relatively narrow zones of the spectrum, a sufficient approximation in describing the background.

The peak and background search method described in this Section was chosen after lengthly testing of two widely used peak search routines (smoothed first and second numerical difference of the data) in order to study their compatibility with our aim of simplicity and velocity. These routines require the following steps:

(1) Smoothing out of the raw data

(2) Differentiation of the smoothed data

(3) Inspection of the variations of the first or second difference.

At this point, some comments are to be made. Step (1), to be really effective, necessitates an approximate knowledge of the value of the FWHM of the peaks. The number of points to be used in the routines which implements steps (1) and (2) is to be evaluated before the analysis, and is then to be given as an external parameter ([10]). Moreover, in step (3) to distinguish real peaks from noise fluctuations, the variations of the derivatives are to be compared with a predetermined threshold, which is usually chosen as the standard deviation of the data multiplied by an appropriate constant factor to be empirically chosen prior to the analysis ([4,15]). Then, further tests are usually applied.

Owing to the above stated incoveniences, which result in lengthening the computer code and increasing the execution time, we decided to adopt a much simpler procedure: the basic principle is to determine the position of all candidate peaks in a very simple way, and to shift all restrictive tests to a fast peak fitting procedure which effectively rejects non Gaussian shapes.

The peak finding and background determination procedures have been based on the same very simple pattern recognition method. The presence of a valley region centred at the channel number J is characterized by the following height pattern

$$D(J-1) > D(J) \quad \text{and} \quad D(J) < D(J+1) \tag{1}$$

where D(I) is the content of the original spectrum at the channel number I. As will be explained in detail in Section 3, selected valley region centres are used to determine the background, which will be indicated as B(I). Then, if the spectrum after background subtraction is described as S(I) = D(I)-B(I), to acknowledge the presence of a peak it is only necessary to find the similar height pattern

$$S(J-1) < S(J) \quad \text{and} \quad S(J) > S(J+1) \tag{2}$$

This channel number J is prospectively accepted as an approximate peak centre. This peak finding procedure may give a number of prospective peak centres larger than the true number, but very small humps are not taken into account with a test on their height and spurious peaks will be discarded in the fitting procedure (see Section 3).

Compared to the more commonly used search routines, this pattern based peak finding procedure enables complete skipping of steps (1) and (2) and replaces step (3) with a simpler and faster examination of the behaviour of the spectrum, working directly on the raw data. A fringe benefit of this simple pattern recognition method is also that it can be coded in such a way that in the programme the same subroutine is alternately used to locate firstly valley regions and then the approximate peak centres.

Since this method is simpler and no less effective than previous ones, we don't see why a sledgehammer should be used when nutcrackers will do.

3. - DESCRIPTION OF THE PROGRAMME

The programme LIZA consists of the following steps:

(1) Input of the data

(2) Background subtraction from the raw spectrum

(3) Peak search on the backgroundless spectrum

(4) Fitting of the peaks in decreasing height order

(5) Subtraction of the calculated Gaussian function from the spectrum. Repetition of steps (4) and (5) until all significant peaks are analyzed

(6) Printout of the results.

3.1 Input of the data

At the start of the programme, the subroutine DATA reads the parameters required for the analysis from punched cards or from a teletype. The spectrum is then read (subroutine RIDIN) from punched paper tape or from a binary disc file and it is analyzed in segments of chosen length, according to the partition technique described in detail elsewhere (4). The use of this piecemeal reading technique is enabled by our analyzing method, which doesn't require the simultaneous presence of the whole spectrum in the computer memory. So, even large spectra may be analyzed using vectors of much smaller length, thereby contributing to the reduction of the core memory requirements.

3.2 Background determination and subtraction

The segment of the raw spectrum under analysis is searched for valley regions by means of the above described pattern recognition method (subroutine PREP). When a sequence of three channel contents which obey the condition (1)

$$D(I-1) > D(I) \quad \text{and} \quad D(I) < D(I+1)$$

is found, the channel number I is stored as a prospective "background point" (BP). The centres of eventual flat zones are also accepted as BPs (subroutine FLAT). After all the BPs in the segment under analysis have been identified, the subroutine CHOIX determines the maximum number of equal length parts of the segment each of which contains at least two BPs. Then a selection procedure is applied so that in each part of the segment only the BP with the lowest content is retained (subroutines CHOIX and CLEAR). This condition is usually sufficient to exclude high valley regions between two adjacent peaks. The ends of the ordinates $D(I)+\sqrt{D(I)}$ centred at the remaining BPs are then joined by straight lines, and the background obtained in this way is subtracted from the raw spectrum ([16]) (subroutine FONDO). In Figs. 1, 2, 3, 6 and 7 the BPs are indicated, and the straight lines represent the calculated background to be subtracted from the spectrum.

3.3 Peak search

Due to the fitting method we have adopted, the peak finding procedure is only required to produce approximated centres of the peaks, which will be simply called "peak points" (PP). The exact centres will be calculated in the fitting procedure. The PPs are determined by the same subroutine used in the background determination, provided the appropriate logical comparison function is used in the calling statement. In fact, the peak search is based on the already described pattern recognition method. When the conditions (2)

$$S(I-1) < S(I) \quad \text{and} \quad S(I) > S(I+1)$$

are fulfilled, the channel number I is assumed to be a PP. However, in order that this PP be stored for subsequent use in the fitting procedure, the value of S(I) is also required to be larger than the statistical error associated with the maximum content of the original spectrum in the segment under analysis. With this very simple condition many spurious peaks and humps due to the statistical fluctuations are not even taken into account in the first examination, while no real peak may pass undetected.

3.4 Peak fitting

The basic assumption of pure Gaussian shape peaks allows the use of the non-iterative fitting technique of Zimmermann ([1]) and Mukoyama ([2]), which is very economical as far as both core memory requirements and execution time are concerned.

In the channels near a PP J, the spectrum S(I) is approximately described by the Gaussian function

$$G(I) = \frac{P}{\sigma\sqrt{2\pi}} \exp\left[-(I-I_o)^2/2\sigma^2\right] \qquad (3)$$

where I_o is the centre of the peak, σ its standard deviation and P its area. Then the function

$$f(I) = \ln\left[S(I-1)/S(I+1)\right] \qquad (4)$$

may be fitted by a straight line. In fact, using a property of the Gaussian, it emerges that the quantity

$$\ln\left[G(I-1)/G(I+1)\right] = 2(I-I_o)/\sigma^2 \qquad (5)$$

is a linear function of I and so is approximately f(I), having assumed S(I) to be nearly equal to G(I). With an explicit least squares fit (subroutine FIT2R), the straight line

$$Y(I) = A.I+B \qquad (6)$$

is fitted to the function (4) in the interval between a lower channel number M on the left of the PP and an upper channel number N on the right of the PP. The parameters I_o and σ are then obtained from the coefficients A and B according to the formulae

$$I_o = -B/A \qquad \sigma = \sqrt{(2/A)} \qquad (7)$$

The area P of the peak is then obtained by fitting (subroutine FIT1R)

to the points of the spectrum S(I) the Gaussian function (3), where P is now the only free parameter left. If the calculated value of I_o, which does not need to be an integer number, differs from J for more than one channel, the prospective peak is far from a Gaussian shape and is then rejected (subroutine FIND).

Instead of trying to resolve the structures of the spectrum by fitting the peaks in the order of increasing channel number, in each segment of the spectrum the PPs are previously arranged (subroutine ORDO) according to the decreasing values of the corresponding S(I), so that the highest peak is the first to be analyzed. In every fit the quantity

$$C = \sqrt{\sum_{I=M}^{N} \frac{[Y(I) - f(I)]^2}{(N-M+1)}} \qquad (8)$$

is calculated (subroutine CHSQ) with M = J-1, N = J+1. The value C_o obtained in the fit of the first peak is chosen as standard for clearance of successive fits, since the first fitted peak has the largest statistics. This value C_o is then multiplied by a constant factor F to give the quantity $E = C_o.F$. In each successive fit, if the corresponding C is larger than E, the prospective peak under analysis is discarded (subroutine FIND). It must be pointed out that the constant F is the only input parameter affecting the analysis procedure. Its value is easily determined empirically by testing the programme on a few sample data chosen at random from the set of spectra to be analyzed.

For a peak centred at the PP J, the values M and N of the lower and upper limits of the fitting interval are initially set at J-1 and J+1 respectively. Although this fitting interval may be sufficient for peaks with FWHM circa equal to three channels, the programme includes an option for the automatic addition of further points both to the right and to the left side of the initial three points to deal more effectively with peaks with larger FWHM. The maximum number of points to be added (if required) on each side is supplied by an external

parameter called IOPT. A value of IOPT different from zero causes the programme to start the optimization subroutine OPT3. Further points are tentatively added one by one on the low-energy side of the fitting interval and in each step the quantity C (defined in formula (8)) is recalculated with the appropriate new value of the limit M. If $C < E$ the point is approved. This procedure goes on until getting to a point for which $C > E$ or the number IOPT of added points is reached. The same procedure is then applied to the points of the high--energy side, finally resulting in the best determination of the values of the limits M and N. The coefficients of the straight line (6) fitted to the points of the function (4) in the enlarged interval from M to N are then used to determine the parameters of the peak according formulae (7).

In contrast to methods employing rigid prescriptions to exclude unfavourable data points from the fitting interval (setting for example its limits to the channels whose contents are greater than half maximum ([2,3])) this flexible method is more suitable to produce better estimates of the Gaussian parameters, especially in the case of strongly overlapping peaks.

3.5 Subtraction of the fitted peak and housekeeping

After a fit has been completed, the calculated Gaussian function (3) is subtracted (subroutine STRIP) from the spectrum S(I) and the residuals in the subtraction region are searched in order to detect eventual peaks previously shadowed by the bigger peak which has just been subtracted. If new PPs are found, they are inserted in the right places in the list of the PPs waiting to be processed according to height. The fitting procedure starts again, and continues until all peaks have been analyzed. It can be observed that the successive stripping of the fitted peaks allows satisfactory analysis of even rather convoluted structures of the spectrum. Moreover, this method overcomes the usual limits ([10,15]) on the maximum number of peaks that can be unfolded from an overlapping group of lines, unavoidable in the simultaneous iterative least-squares fitting procedure unless a very large computer is available ([11]).

3.6 Printout of the results

The programme may run either in a completely automatic manner or, alternatively, in an interactive way. In the automatic way of operation, the only requested output device is a line printer, and the printout consists of two parts:

(1) the list of the parameters of each analyzed peak

(2) an alphanumeric show of the results on the line printer, at the end of the analysis of a segment of the spectrum.

In the plot the fitted peaks, the background and the spectrum reconstructed by summing the peaks and the background (subroutine SOMGS) over the whole segment are plotted against the raw data (subroutines FARBE and PLOT). The points of D(I) are represented by the symbol "*", those of B(I) by the symbol ":". The profiles of the fitted Gaussian functions are outlined by a different letter for each peak. The reconstructed spectrum points are indicated by the symbol "+". A typical example of this kind of plot is shown in Fig. 4. The plot, if not required, may be omitted.

The second version also requires an interactive CRT display. In fact, steps (1) and (2) are initially executed on the display instead of on the line printer. When the plot is shown on the CRT, a feature of the programme allows the user to select the peaks to be printed simply by pressing on the CRT keyboard the letters corresponding to the desired peaks. This situation is illustrated in Fig. 5.

4. - EXAMPLES

Some examples of the application of our programme will be shown in this Section.

Fig. 1 describes in detail the analysis of a γ-peak (the 662 keV γ-ray from a ^{137}Cs calibrated source) recorded with a 50 cc. Ge(Li) detector ([17]). Part A shows the original spectrum. The BPs and the PP are indicated. Part B shows the behaviour of the function f(I):

the limit points M and N of the fit of the straight line (6) are indicated. In part C, the points represent the spectrum S(I), the full line the calculated Gaussian curve. A value of IOPT = 2 (maximum seven channels for the fit) was adopted, and the programme performed the fit on six channels.

Fig. 2 shows the analysis of a zone of a spectrum from the measurement of the γ-rays following the inelastic scattering of 2.5 MeV neutrons in ^{127}I, recorded with the same Ge(Li) detector. The BPs are indicated, the PPs are replaced by the arrows indicating the energies (in keV) of the observed γ-rays. A value of F = 10 and IOPT = 1 was adopted (maximum five points fitted for every peak). The points represent the original spectrum, the straight lines the calculated background, the curves are the fitted Gaussian shapes.

Fig. 2 shows a zone of spectrum with well-resolved peaks; Fig. 3, on the other hand, emphasizes the performance of the programme in fitting overlapping peaks. The data is taken from the $^{93}Nb(n,n'\gamma)^{93}Nb$ reaction [18]. The points represent the spectrum D(I), the thin curves the single components of the calculated spectrum, the thick curve their sum. The BPs, PPs and the calculated background are indicated as in Fig. 2, together with the energies of the observed γ-rays. It can be interesting to observe the fit of the multiple peaks, such as the 473, 477, 482 keV triplet and the 507-511, 538-541 and 568-572 keV doublets. The FWHM of the detector, obtained, for example, from the data of Fig. 1, was about 2.4 keV.

Fig. 4 reproduces directly the line printer output of a part of the same zone of the spectrum, printed as described in detail in Subsection 3.6: the letters from A to L identify the single components of the spectrum.

For the sake of completeness, Fig. 5 shows the situation in the case of the use of the interactive version of the programme, on the same zone of spectrum again. The picture directly presents the CRT display, on which the user can choose the interesting peaks by means of their labels.

Figs. 6 and 7 show the analysis of two spectra of elastically scattered protons recorded with a 1000 μm Si detector ([19]). The data is taken from a study of the $^{68}Zn(p,p)^{68}Zn$ reaction.

Fig. 6 shows the analysis of part of a spectrum that presents, from the left to the right side, the peaks from the $^{12}C(p,p)^{12}C$, $^{16}O(p,p)^{16}O$, and $^{68}Zn(p,p)^{68}Zn$ reactions. In fact, the target consisted of a thin layer of ZnO evaporated on a 10 $\mu g/cm^2$ carbon backing. The PPs and BPs are indicated, even though the background is too small to be appreciated on the scale. The points represent the spectrum D(I), the curves the calculated Gaussians.

Fig. 7 shows a peak from the elastic scattering of protons from ^{68}Zn, in which the target presented a contamination from a lighter element, resulting in a small peak on the left side of the peak due to the $^{68}Zn(p,p)^{68}Zn$ reaction. The thin lines represent the two components, the thick one their sum. As in Fig. 6, the background is too low to be appreciated on the scale. The PPs and BPs are indicated.

5. - COMMENTS AND CONCLUSIONS

The performance of the programme described in this report may be judged by inspecting the fits shown in the figures. It is to be stressed that these results have been obtained in an appreciably short execution time and with a small core memory occupation (less than 8K, 16-bit word). For example, on a HP 2100 S computer, which has a base-cycle of 980 ns, the automatic analysis of a 4096 channel spectrum with about 100 peaks requires 120 s of central processor time. This short execution time may suggest using this programme as a first step to obtain the preliminary information (such as a good approximation of the position, FWHM and height of each peak) necessary to a successive least-square iterative fit, using when necessary more refined line shapes. Moreover, the time usually necessary to set up the programme for the analysis of a batch of similar spectra is remarkably reduced by the fact that the sensitivity of the analysis is controlled by a single parameter (F).

The results obtained with the programmes LIZA and DUMAN ([4]) have been extensively compared: a good general agreement was found.

The authors would like to thank Prof. F. Demanins for his continuous interest in this work.

REFERENCES

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

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

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

(4) U. Abbondanno, A. Boiti, F. Demanins and M.R. Malisan, Nucl. Instr. and Meth. 148 (1978) 577.

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

(6) R.G. Helmer, R.L. Heath, M. Putnam and D.H. Gipson, Nucl. Instr. and Meth. 57 (1967) 46.

(7) J.T. Routti and S.G. Prussin, Nucl. Instr. and Meth. 72 (1969) 125.

(8) L. Varnell and J. Trischuk, Nucl. Instr. and Meth. 76 (1969) 109.

(9) D.C. Robinson, Nucl. Instr. and Meth. 78 (1970) 120.

(10) W. Teoh, Nucl. Instr. and Meth. 109 (1973) 509.

(11) N. Sasamoto, K. Koyama and S.I. Tanaka, Nucl. Instr. and Meth. 125 (1975) 507.

(12) D.C. Camp, Report UCRL-50156 (March 1967).

(13) R. Gunnink, R.A. Meyer, J.B. Niday and R.P. Anderson, Nucl. Instr. and Meth. 65 (1968) 26.

(14) J. Kern, Nucl. Instr. and Meth. 79 (1970) 233.

(15) K.S. Basu and A.P. Patro, Nucl. Instr. and Meth. 126 (1975) 115.

(16) W.W. Bowman, Nucl. Instr. and Meth. 96 (1971) 135.

(17) Supplied by Canberra, model 721 K.

(18) U. Abbondanno, F. Demanins, A. Giletti, M.R. Malisan, C. Tuniz and G. Nardelli, Lett. Nuovo Cimento 21 (1978) 409.

(19) Supplied by Ortec, model A-018-050-1000.

FIGURE CAPTIONS

Fig. 1 -

Analysis of the 662 keV γ-ray from a ^{137}Cs source recorded with a Ge(Li) detector. In part A the original spectrum D(I) (points) and the calculated background B(I) (straight line) are shown. In part B the values of the function f(I) and the fitted straight line Y(I) are plotted: the limit points of the fit are labeled M and N. In part C, the points represent the spectrum S(I) after background subtraction, the line the calculated Gaussian function.

Fig. 2 -

Analysis of a zone of spectrum taken from the $^{127}I(n,n'\gamma)^{127}I$ reaction, recorded with a Ge(Li) detector. The points represent the spectrum D(I), the straight lines the calculated background B(I), the curves are the Gaussian shapes fitted to the peaks. The energies of the observed γ-rays are also indicated.

Fig. 3 -

Analysis of a zone of spectrum taken from the $^{93}Nb(n,n'\gamma)^{93}Nb$ reaction, recorded with a Ge(Li) detector. The points represent the spectrum D(I), the straight line the calculated background B(I). The thin curves are the single components of the analyzed spectrum, the thick one their sum. The energies of the observed γ-rays are also indicated.

Fig. 4 -

Direct computer line printer output of a part of the zone of spectrum of Fig. 3. The values of the raw spectrum are labeled "*", those of the calculated background are labeled ":". The single components of the calculated spectrum are indicated by the letters from "A" to "L". The spectrum reconstructed by adding the calculated Gaussians to the background is labeled "+".

Fig. 5 -

CRT display of the same zone of Figs. 3 and 4. By inspecting this display, the user can select the peaks to be printed on line printer simply by pressing the corresponding letter on the keyboard.

Fig. 6 -

Analysis of a zone of spectrum from the $^{68}Zn(p,p)^{68}Zn$ reaction, recorded with a Si detector. The points represent the original spectrum D(I), the curves the calculated Gaussians.

Fig. 7 -

Analysis of a double peak from the $^{68}Zn(p,p)^{68}Zn$ reaction, recorded with a Si detector. The points represent the spectrum D(I), the thin curves the single components, the thick curve the reconstructed spectrum.

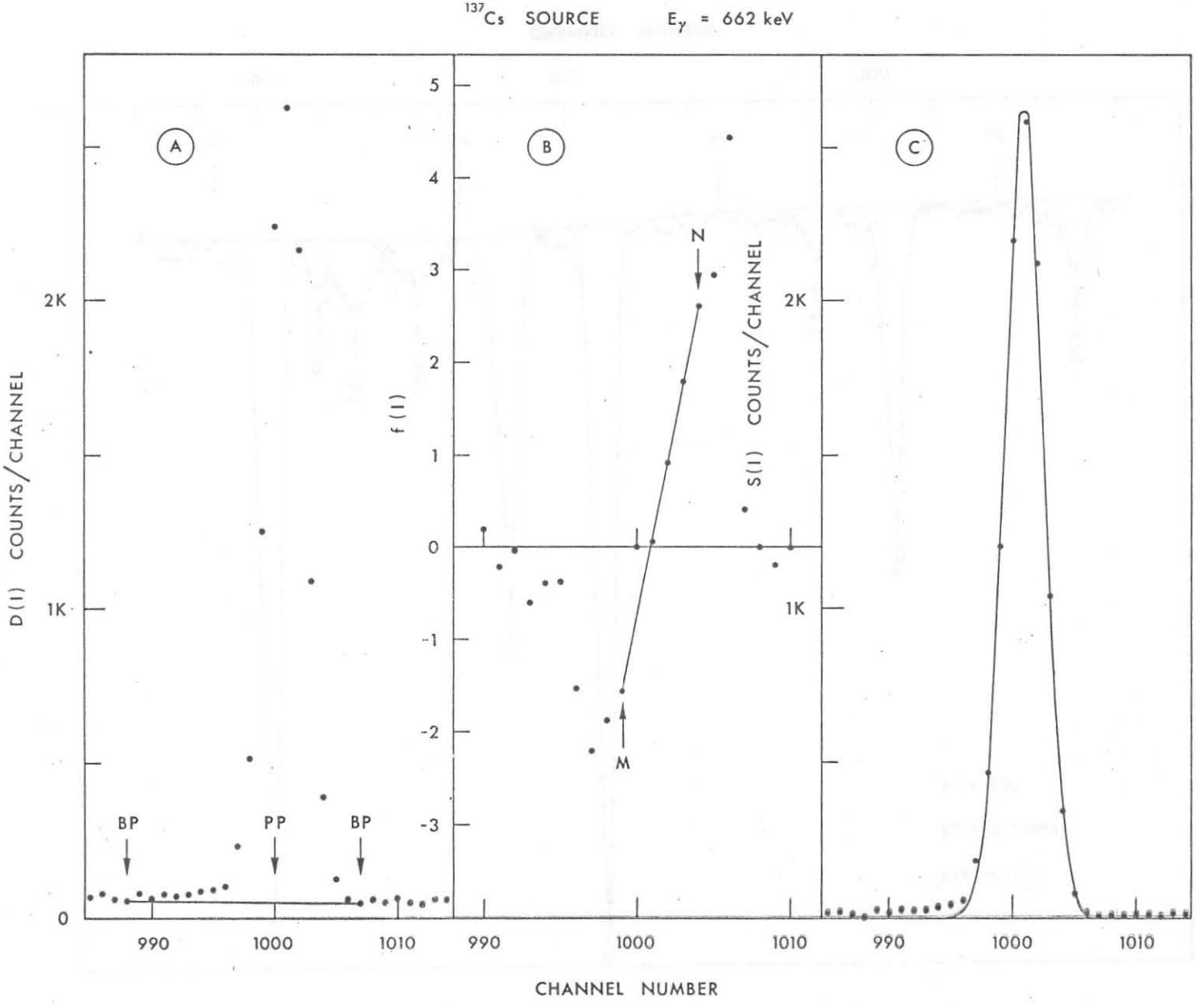
^{137}Cs SOURCE E_γ = 662 keV
A
B
C
D(I) COUNTS/CHANNEL
f(I)
S(I) COUNTS/CHANNEL
2K
1K
0
5
4
3
2
1
0
-1
-2
-3
BP
PP
BP
N
M
990
1000
1010
CHANNEL NUMBER

fig 1

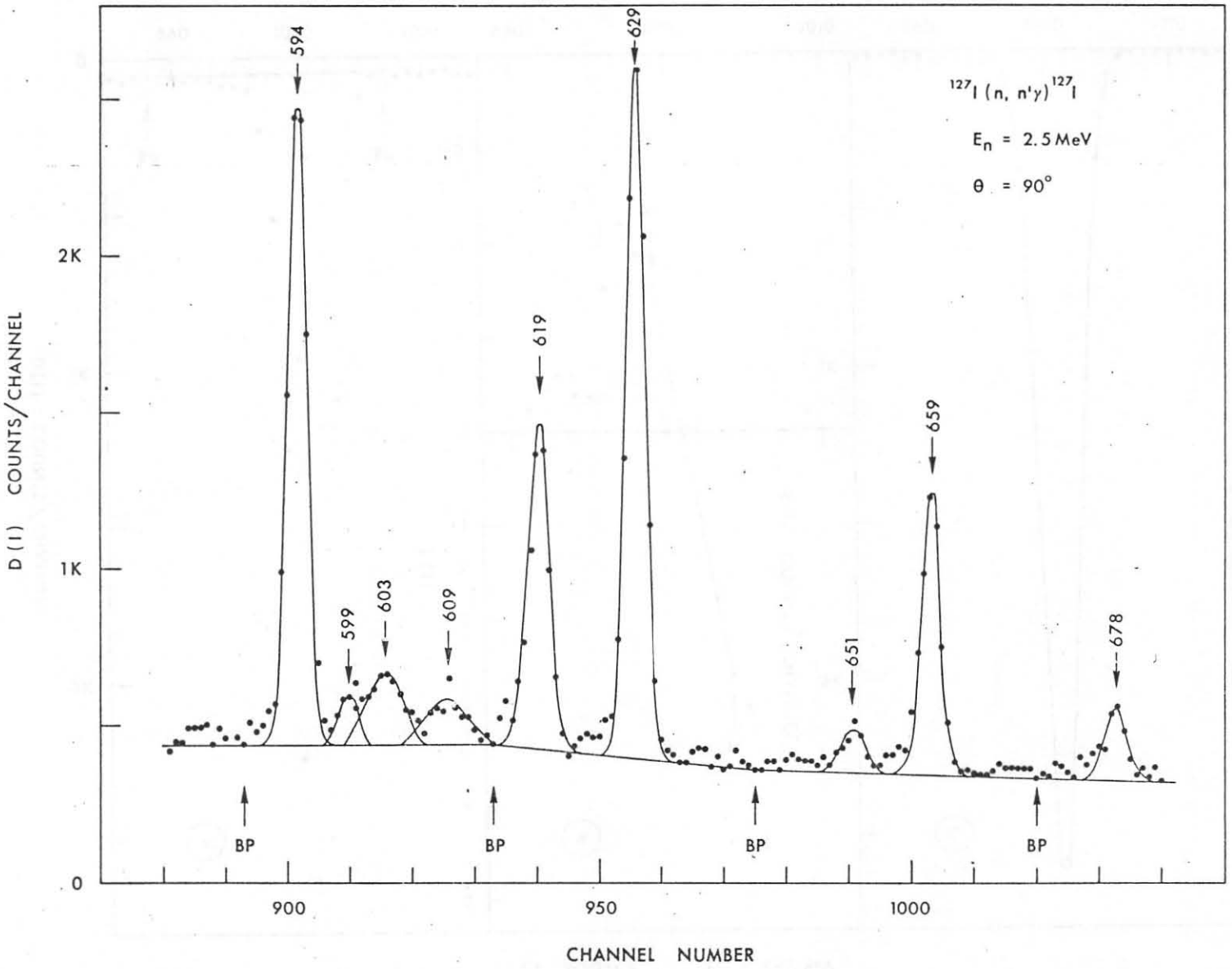
$^{127}I\,(n, n'\gamma)^{127}I$
E_n = 2.5 MeV
θ = 90°
D (I) COUNTS/CHANNEL
2K
1K
0
594
599
603
609
619
629
651
659
678
BP
BP
BP
BP
900
950
1000
CHANNEL NUMBER

fig. 2

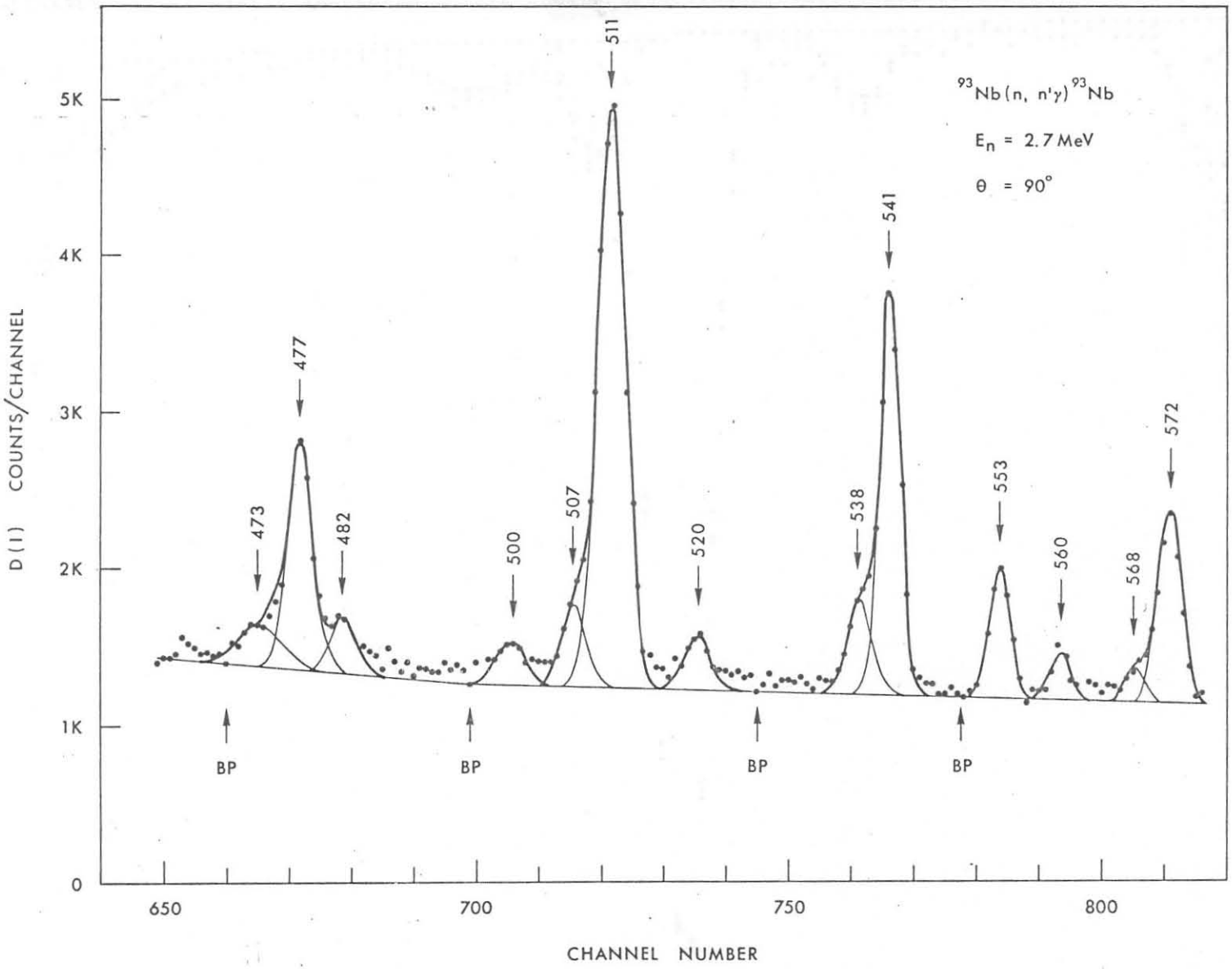
$^{93}Nb(n, n'\gamma)^{93}Nb$
E_n = 2.7 MeV
θ = 90°
D(I) COUNTS/CHANNEL
CHANNEL NUMBER
5K
4K
3K
2K
1K
0
650
700
750
800
473
477
482
500
507
511
520
538
541
553
560
568
572
BP
BP
BP
BP

fig. 3

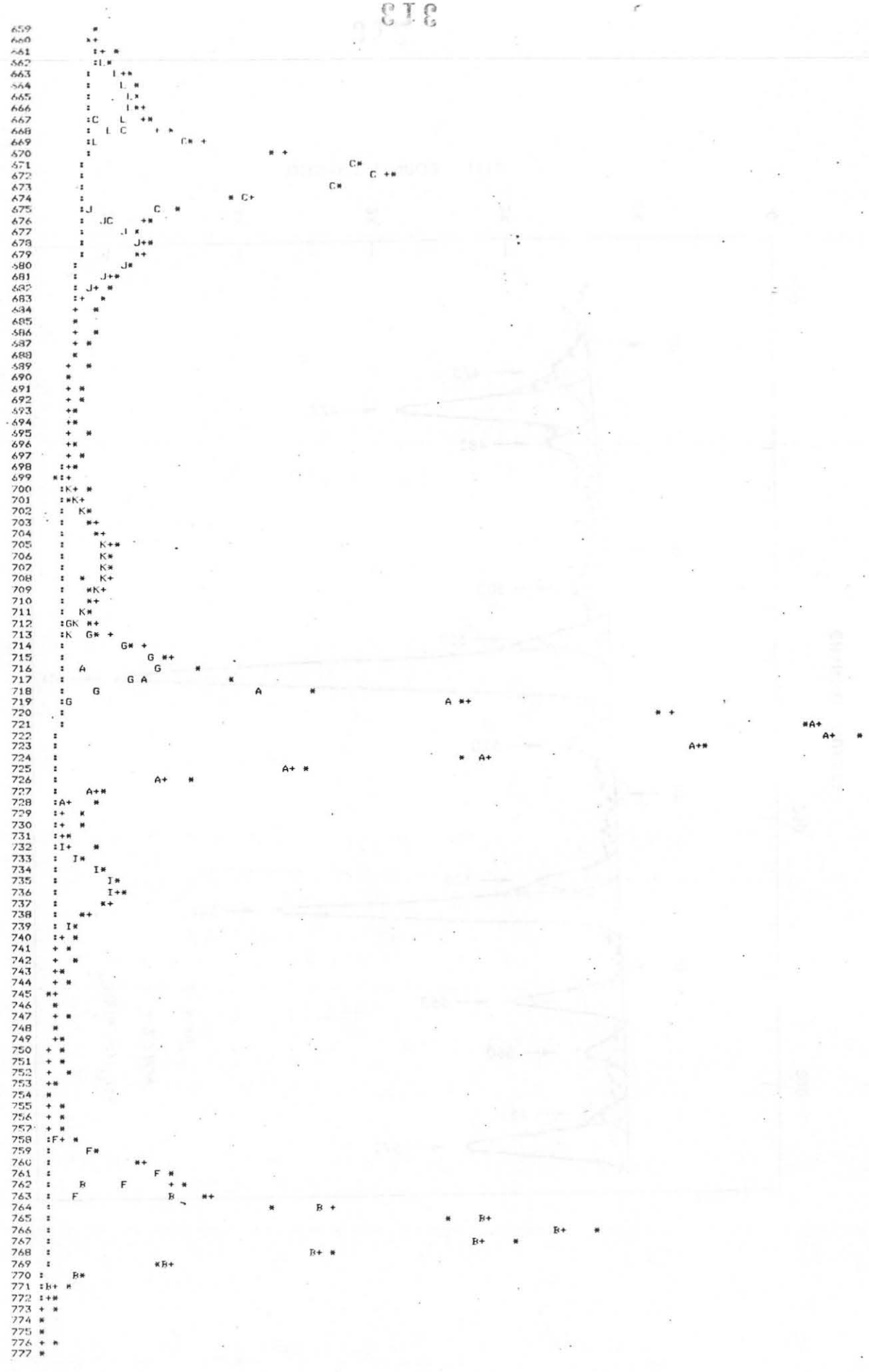

fig. 4

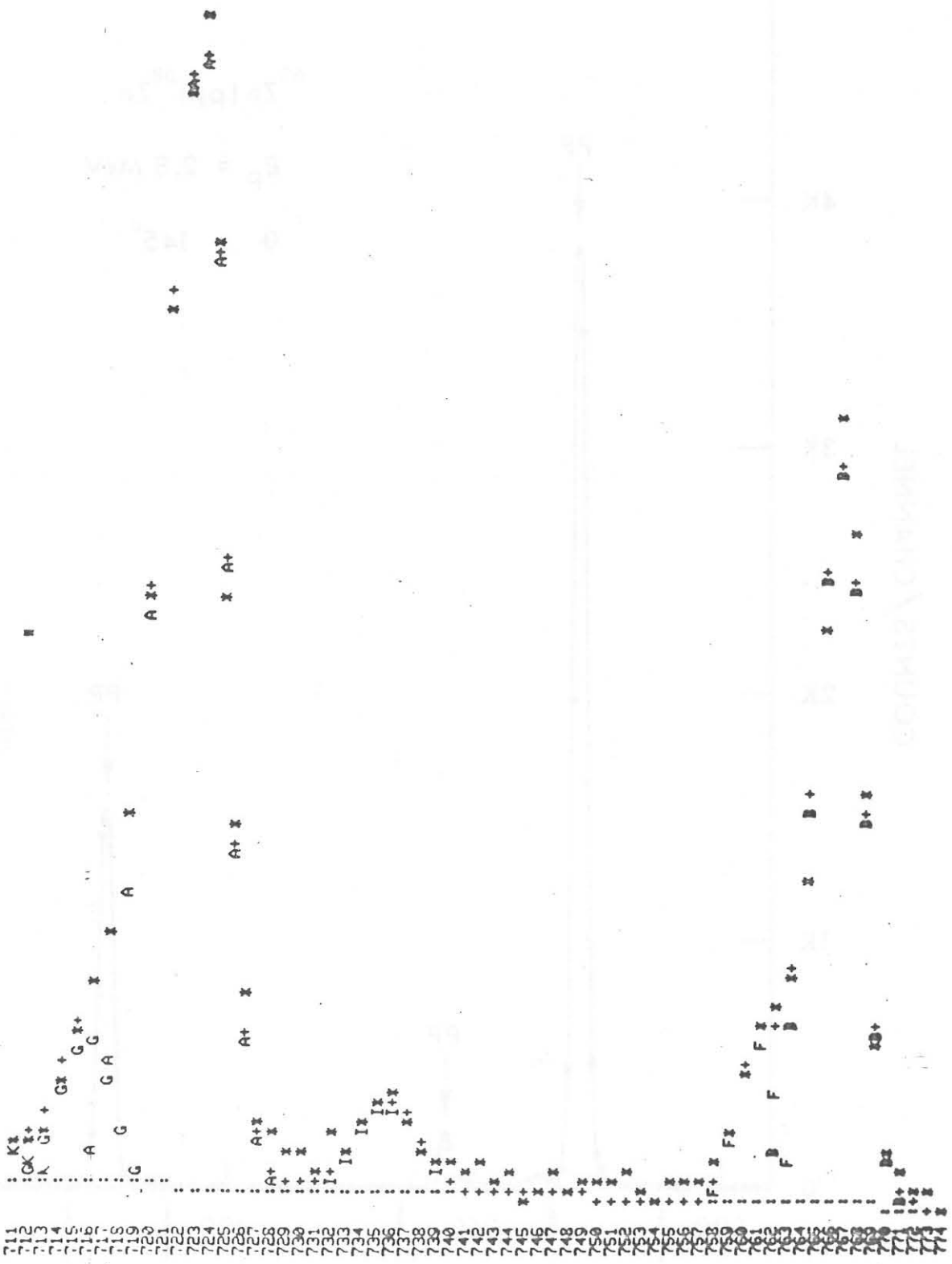

fig. 5

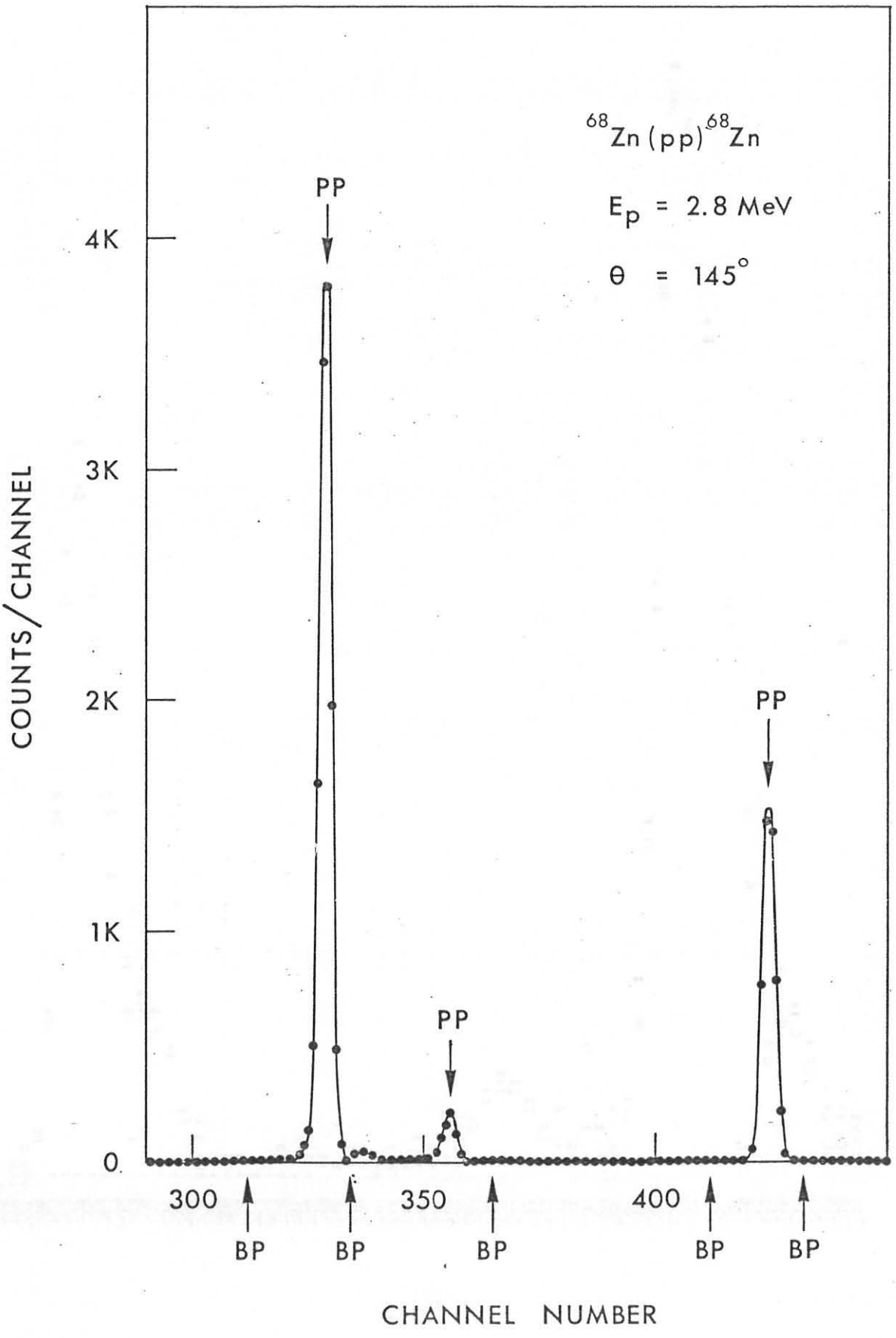
$^{68}Zn(pp)^{68}Zn$
E_p = 2.8 MeV
θ = 145°
PP
PP
PP
4K
3K
2K
1K
0
COUNTS/CHANNEL
300
350
400
BP
BP
BP
BP
BP
CHANNEL NUMBER

fig. 6

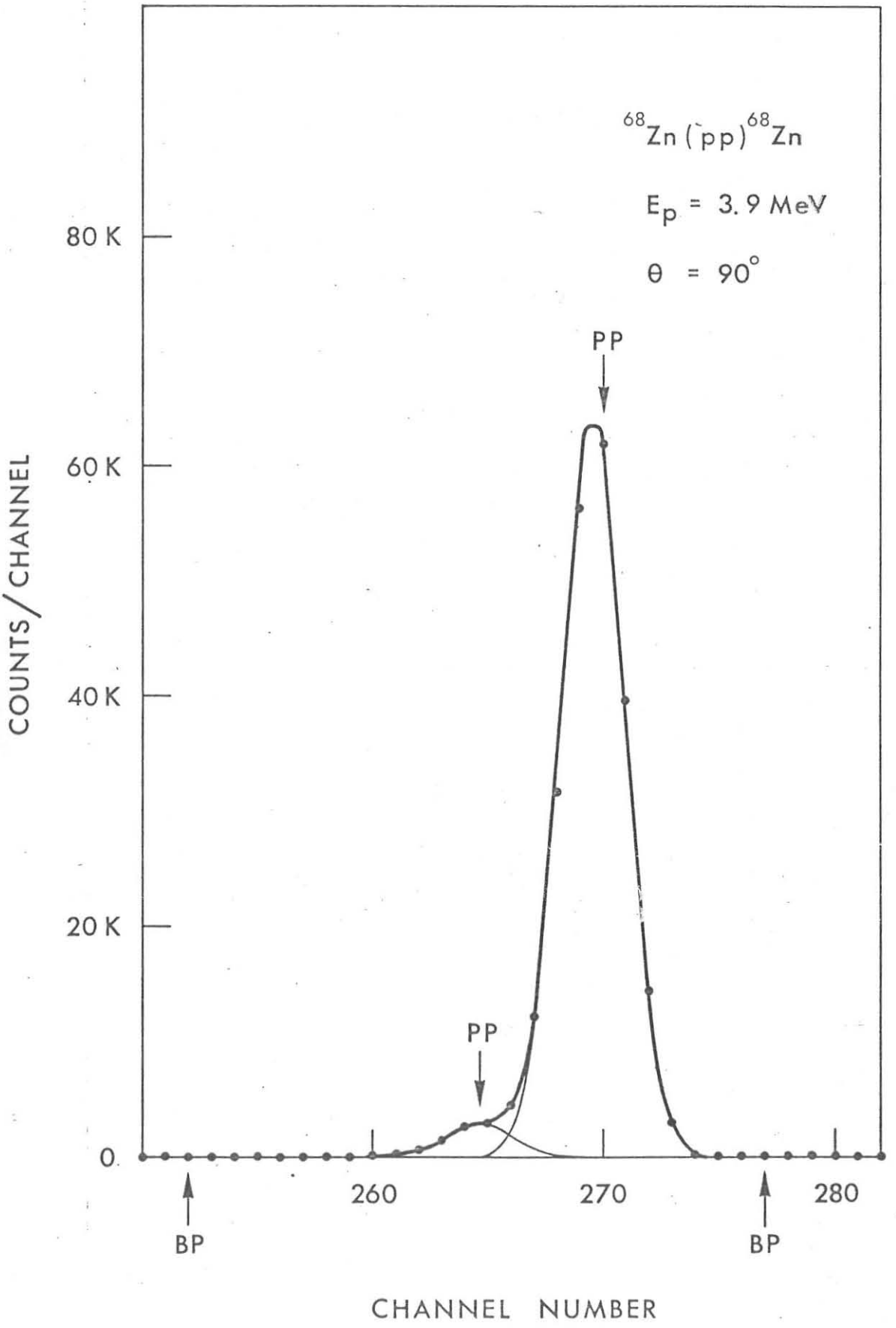
$^{68}Zn(\dot{p}p)^{68}Zn$
E_p = 3.9 MeV
θ = 90°
80 K
60 K
40 K
20 K
0
COUNTS / CHANNEL
PP
PP
BP
BP
260
270
280
CHANNEL NUMBER

fig. 7

APPENDIX

```
0001  FTN4,L
0002  C                    PROGRAM   LIZA
0003  C
0004  C               DESCRIPTION OF INPUT DATA
0005  C
0006  C      TWO CARDS ARE NEEDED FOR EVERY SPECTRUM TO BE ANALYZED
0007  C
0008  C      THE FIRST CARD CONTAINS :
0009  C      LU,KFORM,KTAP,IFILE,ITITLE
0010  C      [ FORMAT (3I5,3A2,4X,15A2) ]
0011  C
0012  C      THE SECOND CARD CONTAINS :
0013  C      LENF,NPAR,NARR,LSG,IOVF,KP,IOPT,ZERO,CONV,F,T
0014  C      [ FORMAT (7I5,5X,4E10.0) ]
0015  C
0016  C      DESCRIPTION OF THE PARAMETERS
0017  C
0018  C      LU  CONTROLS THE OUTPUT IN STEPS OF INCREASING COMPLETENESS
0019  C      LU = 0   IS THE COMMON WAY OF OPERATION :
0020  C               THE OUTPUT PRESENTS ONLY THE DATA OF THE CHOSEN PEAKS ,
0021  C               HOWEVER PLOTS OF PARTS OF THE SPECTRUM MAY BE OBTAINED
0022  C               PRESSING 1 ON THE KEYBOARD OF THE VIDEODISPLAY
0023  C      LU = 1   THE ZONE OF THE SPECTRUM WHICH CONTAINS THE CHOSEN PEAKS
0024  C               IS PLOTTED ON THE LINE PRINTER
0025  C      LU = 2   THE CONTENTS OF THE SPECTRUM , THE BACKGROUND AND THE
0026  C               RECONSTRUCTED SPECTRUM ARE PRINTED OUT
0027  C      LU = 3   EVERY WHOLE ZONE OF THE SPECTRUM IS PLOTTED
0028  C      LU = 4   THE CONTENTS OF THE SPECTRUM AFTER SUBTRACTION OF THE BACK-
0029  C               GROUND AND THE VALUES OF THE FUNCTION LOG(S(I-1)/S(I+1))
0030  C               ARE ALSO PRINTED
0031  C      LU = 99 DETERMINES THE PROGRAMME TO STOP
0032  C
0033  C      KFORM  CONTROLS THE FORMAT OF THE INPUT SPECTRUM
0034  C      KFORM = 0 : FORMAT (7X,8(F6.0,1X))    INPUT FROM PAPER TAPE
0035  C      KFORM = 1 : FORMAT (8(F6.0,1X))       INPUT FROM PAPER TAPE
0036  C      KFORM = 2 : INPUT FROM BINARY DISC FILES
0037  C
0038  C      KTAP ≠ 0 IMPLIES THE PERFORATION OF THE DATA OF THE PEAKS
0039  C               ON PAPER TAPE
0040  C
0041  C      IFILE  IS THE NAME OF THE EVENTUAL FILE CONTAINING THE SPECTRUM
0042  C
0043  C      ITITLE  ALLOWS TO MARK EACH ANALYZED SPECTRUM WITH A TITLE
0044  C
0045  C      LENF  IS THE LENGTH OF THE SPECTRUM
0046  C
0047  C      NPAR,NARR  IDENTIFY THE FIRST AND LAST CHANNEL OF THE ZONE
0048  C                 OF THE SPECTRUM TO BE ANALYZED
0049  C
0050  C      LSG  IS THE LENGTH OF THE SEGMENT OF THE SPECTRUM READ BY
0051  C           THE SUBROUTINE  RIDIN
0052  C
0053  C      IOVF  IS THE NUMBER OF CHANNELS OF THE ZONE OF OVERLAP BETWEEN
0054  C            TWO CONSECUTIVE SEGMENTS
0055  C
0056  C      THE MAXIMUM VALUE OF  LSG + IOVF  HAS BEEN SET TO  176
0057  C
0058  C      NPAR,NARR,LSG,IOVF  ARE RECALCULATED MODULO 8 IN THE SUBROUTINE DATA
0059  C
0060  C      KP  IS THE LIMIT SET ON THE NUMBER OF PEAKS WHICH MAY BE EXTRACTED
0061  C          FROM EACH SEGMENT OF  LSG + IOVF  CHANNELS
0062  C          THE MAXIMUM VALUE OF  KP  IS  26
0063  C
0064  C      IOPT  CONTROLS THE OPTIMIZATION OPTIONS
0065  C      IOPT = 0  THE OPTIMIZATION ROUTINE IS NOT ACTIVATED
0066  C      IOPT = N  THE OPTIMIZATION ROUTINE IS ACTIVATED WITH A MAXIMUM
0067  C                NUMBER OF POINTS EQUAL TO  N
0068  C
0069  C      ZERO  (KEV)  IS THE VALUE OF THE ENERGY SCALE AT THE CHANNEL
0070  C                   ZERO OF THE SPECTRUM
0071  C
```

```
0072  C     CONV  (KEV/CHANNEL)  IS THE CONVERSION FACTOR OF THE ENERGY SCALE
0073  C
0074  C     F  IS A FACTOR WHICH AFFECTS THE ACCEPTABILITY OF PEAKS RELATED
0075  C        TO THE FIRST FIT IN EACH SEGMENT OF THE SPECTRUM
0076  C        F ( )1 ) SHOULD BE DETERMINED EMPIRICALLY  ( SEE SECTION 3.4 )
0077  C
0078  C     T  IS THE HEIGHT BELOW WHICH PEAKS ARE REJECTED
0079  C
0080  C
0081        PROGRAM LIZA
0082        EXTERNAL GAUSS,IGT,ILT,GTN,LTN
0083        INTEGER P,R,V,W,TAPE,ROW2
0084        DIMENSION SI(26),AK(26),CK(26),L(176),S(176),Q(176)
0085        DIMENSION CC(26),EN(26),FW(26),SA(26),ER(26),AS(26),LP(26)
0086        DIMENSION IKK(26),INN(26),KK(26),NN(26),IP(60),IC(30)
0087        COMMON AN(176),ROW1(176),ROW2(176)
0088        COMMON LENF,NPAR,LSG,IOVF,NOK,NSG,LSS,LUS,KFORM
0089        COMMON IOPT,LU,KP,ZERO,CONV,F,T,IFILE(3),KTAP
0090        DATA IC/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,
0091       C1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H-,1H:,1H+,1H*/
0092        DATA H1/-1.E37/,HH/1.E38/
0093        DATA V/1/,R/5/,W/6/,TAPE/7/,P/4/
0094  3     CALL DATA(R,W)
0095        IF (KFORM.EQ.2) GO TO 1
0096        IF (NPAR.EQ.0) GO TO 1
0097        CALL RIDIN(.TRUE.,NPAR,IK,TAPE)
0098  1     IF (NOK.EQ.0) GO TO 3
0099        KP = MINO(KP,26)
0100        PK = 0.
0101        DO 5 LAV=1,NSG
0102        IF (LAV.EQ.NSG) LSS = LUS
0103        IF (LAV.GT.1) GO TO 6
0104        NARR = NPAR+LSS
0105        KANAL = LSS
0106        IK = 1
0107        GO TO 7
0108  6     NPAR = NARR-IOVF
0109        NARR = NPAR+LSS
0110        KANAL = LSS-IOVF
0111        IK = IOVF+1
0112        IF (IK.EQ.1) GO TO 7
0113        DO 8 I=1,IOVF
0114  8     AN(I) = AN(I+LSG)
0115  7     CALL RIDIN(.FALSE.,KANAL,IK,TAPE)
0116        MP = NPAR+1
0117        CALL SALTO(V)
0118        WRITE (V,118) MP,NARR
0119        WRITE (W,118) MP,NARR
0120  118   FORMAT(/,7X,"SEGMENT FROM CHANNEL",I5," TO CHANNEL",I5,/)
0121        CALL EXTR(AN,LSS,SMIN,SMAX,HH,HI)
0122        CUT = SQRT(SMAX)
0123        IF (SMAX-CUT.LE.SMIN) GO TO 21
0124        CUT = AMAX1(CUT,T)
0125        CALL FONDO(W,HH,L,S,IP,NP)
0126        IF (NP.LT.2) GO TO 5
0127        N1 = 1
0128        N2 = LSS
0129        IF (LU.GT.2) GO TO 24
0130        N2 = 1
0131        N1 = LSS
0132  24    WRITE (V,104)
0133  104   FORMAT (/,7X,"PEAK",4X,"CENTRE",4X,"HEIGHT",6X,"FWHM",4X,"ENERGY",
0134       P5X,"ERROR",6X,"AREA",5X,"ERROR",8X,"FIT CH.",2X,"BASE CH.",/)
0135        MP = 0
0136        NP = 0
0137        LI = 1
0138        LS = LSS
0139  20    CALL PREP(LTN,IP,NP,LI,LS,S,CUT)
0140        IF (NP.EQ.0) GO TO 2
0141        CALL ORDO(IP,NP,S,GTN,0)
0142        CALL ELIM(IP,NP)
0143  22    DO 15 K=1,LSS
0144        ROW1(K) = 1.
0145  15    ROW2(K) = K
0146        DO 23 K=2,LSS-1
0147  23    Q(K) = ALOG(S(K-1)/S(K+1))
```

```
0148          CALL FIND(IBP,K,N,IP,NP,S,Q,A,B,MP)
0149          IF (IBP.EQ.0) GO TO 2
0150          IF (LU.GE.2) CALL PRISP(LU,MP,L,S,Q,W)
0151          SIG = SQRT(2./A)
0152          CTR = -B/A
0153          IBP = CTR
0154          DO 11 I=1,LSS
0155    11    ROW1(I) = GAUSS(I,SIG,1.,CTR)
0156          CALL FIT1R(K,N,A,S)
0157          IF (A.LT.CUT) GO TO 22
0158          MP = MP+1
0159          CK(MP) = CTR
0160          AK(MP) = A
0161          SI(MP) = SIG
0162          ER(MP) = SIG*CONV
0163          IKK(MP) = K+NPAR
0164          INN(MP) = N+NPAR
0165          CC(MP) = CTR+NPAR
0166          EN(MP) = CC(MP)*CONV+ZERO
0167          FW(MP) = 2.355*ER(MP)
0168          CALL STRIP(A,LI,LS,IBP,S)
0169          A = 2.507*SIG*A
0170          AS(MP) = A
0171          DO 16 I=LI,LS
0172    16    A = A+L(I)
0173          SA(MP) = SQRT(A)
0174          KK(MP) = LI+NPAR
0175          NN(MP) = LS+NPAR
0176          WRITE (V,105) MP,IC(MP),CC(MP),AK(MP),FW(MP),EN(MP),
0177         PER(MP),AS(MP),SA(MP),IKK(MP),INN(MP),KK(MP),NN(MP)
0178    105   FORMAT (7X,I2,1X,A1,7F10.2,5X,4I5,/)
0179          IF (N1.GT.LI) N1 = LI
0180          IF (N2.LT.LS) N2 = LS
0181          IF (MP.LT.KP) GO TO 20
0182    2     IF (MP.NE.0) GO TO 4
0183    21    WRITE (W,121)
0184          WRITE (V,121)
0185    121   FORMAT (/,7X,"NO PEAKS",/)
0186          GO TO 5
0187    4     CALL SOMGS(MP,LSS,L,CK,SI,AK,S)
0188          CALL EXTR(S,LSS,SMI,SMA,SMIN,SMAX)
0189          WRITE (V,122)
0190    122   FORMAT ("CHOOSE YOUR PEAKS",/)
0191          CALL FARME(V,1,LSS,MP,CK,SI,AK,SMI,SMA,AN,S,L,IC)
0192          WRITE (V,123)
0193    123   FORMAT (/,"INPUT AB...YZ")
0194          DO 9 I=1,26
0195    9     LP(I) = 1H
0196          READ (V,124) (LP(I),I=1,26)
0197    124   FORMAT (26A1)
0198          NP = MP
0199          DO 17 I=1,NP
0200    17    IP(I) = IFIX(CC(I))
0201          DO 13 I=1,NP
0202          DO 14 J=1,26
0203          IF (LP(J).EQ.IC(I)) GO TO 13
0204    14    CONTINUE
0205          MP = MP-1
0206          IP(I) = 10000
0207          AK(I) = 0.
0208    13    CONTINUE
0209          IF (MP.EQ.0) GO TO 21
0210          WRITE (W,104)
0211          CALL ORDO(IP,NP,S,ILT,1)
0212          DO 10 K=1,MP
0213          DO 18 J=1,NP
0214          IF (IP(K).NE.IFIX(CC(J))) GO TO 18
0215          PK = PK+1.
0216          IF (K.EQ.1) N1 = KK(J)-NPAR
0217          IF (K.EQ.MP) N2 = NN(J)-NPAR
0218          GO TO 19
0219    18    CONTINUE
0220    19    IF (KTAP.NE.0) WRITE (P,103) EN(J),ER(J),AS(J),SA(J)
0221    103   FORMAT (4F10.2)
0222    10    WRITE (W,105) K,IC(J),CC(J),AK(J),FW(J),EN(J),ER(J),
0223         PAS(J),SA(J),IKK(J),INN(J),KK(J),NN(J)
```

```
0224          IF (LU.NE.0) GO TO 12
0225          WRITE (V,126)
0226    126   FORMAT (/,"TO GET A PRINTED PLOT PRESS 1")
0227          READ (V,*) MP
0228          IF (MP.NE.1) GO TO 5
0229    12    CONTINUE
0230          CALL SOMGS(NP,LSS,L,CK,SI,AK,S)
0231          CALL EXTR(S,LSS,SMI,SMA,SMIN,SMAX)
0232          CALL FARBE(W,N1,N2,NP,CK,SI,AK,SMI,SMA,AN,S,L,IC)
0233          IF (LU.LT.2) GO TO 5
0234          WRITE (W,112)
0235    112   FORMAT (/,2X,"CONSTR. SPECTRUM",/)
0236          WRITE (W,108) (S(I),I=1,LSS)
0237    108   FORMAT (10(2X,F10.0))
0238    5     CONTINUE
0239          IF (KTAP.NE.0) WRITE (P,103) PK,PK,PK,PK
0240          KP = PK
0241          WRITE (W,106) KP
0242    106   FORMAT (2X,I3," PEAKS HAVE BEEN LISTED",/)
0243          IF (KFORM.EQ.2) GO TO 3
0244          IF (NARR.EQ.LENF) GO TO 3
0245          KANAL = LENF-NARR
0246          CALL RIDIN(.TRUE.,KANAL,IK,TAPE)
0247          GO TO 3
0248          END

**  NO ERRORS**     PROGRAM = 03094       COMMON = 00904



0249          SUBROUTINE DATA(R,W)
0250          INTEGER ROW2,R,W
0251          DIMENSION ITITLE(15)
0252          COMMON AN(176),ROW1(176),ROW2(176)
0253          COMMON LENF,NPAR,LSG,IOVF,NOK,NSG,LSS,LUS,KFORM
0254          COMMON IOPT,LU,KP,ZERO,CONV,F,T,IFILE(3),KTAP
0255          DO 210 I = 1,15
0256    210   ITITLE(I) = 2H
0257          READ (R,201) LU,KFORM,KTAP,(IFILE(I),I=1,3),(ITITLE(I),I=1,15)
0258    201   FORMAT (3I5,3A2,4X,15A2)
0259          IF (LU.EQ.99) GO TO 99
0260          WRITE (W,202) (ITITLE(I),I=1,15)
0261    202   FORMAT (20X,15A2,/)
0262          READ (R,101) LENF,NPAR,NARR,LSG,IOVF,KP,IOPT,ZERO,CONV,F,T
0263    101   FORMAT (7I5,5X,4E10.0)
0264          IF (KFORM.EQ.2) WRITE (W,107) (IFILE(I),I=1,3)
0265    107   FORMAT (2X,"SPECTRUM READ FROM ",3A2,/)
0266          NPAR = (NPAR/8)*8
0267          NARR = (NARR/8)*8
0268          LSG = (LSG/8)*8
0269          IOVF = (IOVF/8)*8
0270          WRITE (W,110) LENF
0271    110   FORMAT (2X,"SPECTRUM OF",I5," CHANNELS",/)
0272          WRITE (W,109) KP
0273    109   FORMAT (2X,"MAXIMUM",I3," PEAKS PER SEGMENT",/)
0274          NOK = NARR-NPAR
0275          NSG = (NOK-IOVF)/LSG
0276          IF (NOK-NSG*LSG.GT.IOVF) NSG = NSG+1
0277          LSS = LSG+IOVF
0278          LUS = NOK-(NSG-1)*LSG
0279          WRITE (W,106) ZERO,CONV,T,F
0280    106   FORMAT (2X,"ZERO ",F8.3," CONV ",F7.3,"  T ",F6.1,"  F ",F6.1,/)
0281          WRITE (W,108) IOPT
0282    108   FORMAT (2X,"IOPT",I3,/)
0283          RETURN
0284    99    WRITE (W,103)
0285    103   FORMAT (5X,"BASTA COSI")
0286          STOP
0287          END

**  NO ERRORS**     PROGRAM = 00431       COMMON = 00904
```

```
0288          SUBROUTINE RIDIN(GARB,KANAL,IK,T)
0289          INTEGER T
0290          COMMON AN(176),D(256),AA(8),LF,NPAR,NX(6),KFORM,NY(11),IFILE(3)
0291          LOGICAL GARB
0292          IF (KFORM.EQ.2) GO TO 6
0293          N = KANAL/8
0294          IF (N.LT.1) RETURN
0295          DO 1 M=1,N
0296          IF (KFORM.EQ.0) GO TO 2
0297          READ (T,9) (AA(K),K=1,8)
0298    9     FORMAT (8(F6.0,1X))
0299          GO TO 3
0300    2     READ (T,8) (AA(K),K=1,8)
0301    8     FORMAT (7X,8(F6.0,1X))
0302    3     IF (GARB) GO TO 1
0303          DO 4 K=1,8
0304          IF (AA(K).LT.1.) AA(K) = 1.
0305          AN(IK) = AA(K)
0306    4     IK = IK+1
0307    1     CONTINUE
0308          RETURN
0309    6     NPR = NPAR+IK
0310          M = NPR*2/128
0311          N = NPR+KANAL
0312          N = (N-M*64)*2
0313          CALL EXEC(14,102B,D,N,IFILE,M)
0314          M = NPR-M*64
0315          N = M+KANAL
0316          DO 7 K=M,N
0317          IF (D(K).LT.1.) D(K) = 1.
0318          AN(IK) = D(K)
0319    7     IK = IK+1
0320          RETURN
0321          END

**  NO ERRORS**     PROGRAM = 00268       COMMON = 00903


0322          SUBROUTINE FONDO(W,HH,L,S,IF,NF)
0323          EXTERNAL GTN
0324          INTEGER W
0325          DIMENSION IF(1),S(1),L(1)
0326          COMMON AN(176),NX(529),NPAR,NY(4),LSS
0327          NF = 0
0328          LI = 12
0329          LS = LSS-11
0330          CALL PREP(GTN,IF,NF,LI,LS,AN,HH)
0331          CALL FLAT(IF,NF,LI,LS)
0332          IF (NF.LT.2) RETURN
0333          CALL CHOIX(NF,IF)
0334          IF (NF.LT.2) RETURN
0335          CALL CLEAR(IF,NF)
0336          IF (NF.LT.2) RETURN
0337          DO 30 I=2,NF
0338          LI = IF(I-1)
0339          LS = IF(I)
0340          A = AN(LS)
0341          A = A+SQRT(A)
0342          B = AN(LI)
0343          B = B+SQRT(B)
0344          A = (A-B)/FLOAT(LS-LI)
0345          B = B-A*LI
0346          IF (I.EQ.2) LI = 1
0347          IF (I.EQ.NF) LS = LSS
0348          DO 31 K=LI,LS
0349    31    L(K) = A*K+B
0350    30    CONTINUE
0351          DO 15 I=1,LSS
0352          IF (L(I).LT.0) L(I) = 0
0353          S(I) = AN(I)-L(I)
0354          IF (S(I).LT.1.) S(I) = 1.
0355    15    CONTINUE
```

```
0356          DO 24 I=1,NF
0357    24    IF(I) = LF(I)+NPAR
0358          WRITE (W,117) (IF(I),I=1,NF)
0359    117   FORMAT (2X,"BACKGRND CH.",10I5)
0360          RETURN
0361          END

**  NO ERRORS**     PROGRAM = 00331       COMMON = 00887


0362  .       SUBROUTINE FLAT(F,NF,LI,LF)
0363          INTEGER F(1)
0364          COMMON AN(176)
0365          IF (NF.GE.60) RETURN
0366          NFI = NF
0367          NI = 1
0368    1     LS = LI+4
0369          IF (LS.GT.LF) RETURN
0370          IF (NI.GT.NFI) GO TO 5
0371          I = NI
0372    2     IF (F(I).GE.LI) GO TO 3
0373          I = I+1
0374          IF (I.LE.NFI) GO TO 2
0375          GO TO 4
0376    3     NI = I+1
0377          IF (F(I).LE.LS) GO TO 8
0378    4     NI = I
0379    5     SUM = 0.
0380          DO 6 I=LI,LS
0381    6     SUM = SUM+AN(I)
0382          SUM = SUM/5
0383          SS = SQRT(SUM)
0384          SI = SUM-SS
0385          SS = SUM+SS
0386          DO 7 I=LI,LS
0387          IF (AN(I).LT.SI) GO TO 8
0388          IF (AN(I).GT.SS) GO TO 8
0389    7     CONTINUE
0390          NF = NF+1
0391          F(NF) = LI+2
0392          IF (NF.EQ.60) RETURN
0393    8     LI = LS+1
0394          GO TO 1
0395          END

**  NO ERRORS**     PROGRAM = 00204       COMMON = 00352


0396          SUBROUTINE CHOIX(NF,IF)
0397          EXTERNAL LTN,ILT
0398          DIMENSION IF(1)
0399          COMMON AN(176),IB(534),LSS
0400          CALL ORDO(IF,NF,AN,ILT,1)
0401          N = 5
0402    7     IN = LSS/N
0403          LS = IN
0404          NI = 1
0405          DO 8 K=1,N
0406          IF (K.EQ.N) LS = LSS
0407          I = 0
0408          DO 9 J=NI,NF
0409          IF (IF(J):GT.LS) GO TO 10
0410          I = I+1
0411    9     CONTINUE
0412    10    IF (I.GE.3) GO TO 6
0413          N = N-1
0414          IF (N.GT.3) GO TO 7
0415          GO TO 5
0416    6     NI = J
0417    8     LS = LS+IN
0418    5     CALL ORDO(IF,NF,AN,LTN,0)
0419          IN = LSS/N
0420          I = 0
```

```
0421          LI = 1
0422          LS = IN
0423          DO 1 K=1,N
0424          IF (K.EQ.N) LS = LSS
0425          DO 2 J=1,NF
0426          IF (IF(J).LT.LI.OR.IF(J).GT.LS) GO TO 2
0427          I = I+1
0428          IB(I) = IF(J)
0429          GO TO 3
0430    2     CONTINUE
0431    3     LI = LI+1N
0432    1     LS = LS+IN
0433          DO 4 J=L,I
0434    4     IF(J) = IB(J)
0435          NF = I
0436          RETURN
0437          END

**  NO ERRORS**     PROGRAM = 00216       COMMON = 00887


0438          SUBROUTINE CLEAR(IF,NF)
0439          DIMENSION IF(1)
0440          COMMON AN(176)
0441          I = 1
0442          J = 2
0443    1     II = IF(I)
0444          IJ = IF(J)
0445          IF (IJ.LT.II+16) GO TO 3
0446    2     IF (J.GE.NF) RETURN
0447          I = J
0448          J = J+1
0449          GO TO 1
0450    3     NF = NF-1
0451          K = I
0452          IF (AN(IJ).GE.AN(II)) K = J
0453          IF (K.GT.NF) RETURN
0454          DO 4 I=K,NF
0455    4     IF(I) = IF(I+1)
0456          J = K
0457          GO TO 2
0458          END

**  NO ERRORS**     PROGRAM = 00110       COMMON = 00352


0459          SUBROUTINE EXTR(T,M,SMIN,SMAX,SMI,SMA)
0460          DIMENSION T(1)
0461          SMAX = SMA
0462          SMIN = SMI
0463          DO 1 I=1,M
0464          IF (SMAX.LT.T(I)) SMAX = T(I)
0465          IF (SMIN.GT.T(I)) SMIN = T(I)
0466    1     CONTINUE
0467          RETURN
0468          END

**  NO ERRORS**     PROGRAM = 00074       COMMON = 00000


0469          SUBROUTINE PREP(CFR,N,M,LI,LS,S,H)
0470          LOGICAL CFR
0471          DIMENSION N(1),S(1)
0472          COMMON NX(886),LSS
0473          I = LI-1
0474          IF (I.LT.3) I = 3
0475          L = LS+1
0476          IF (L.GT.LSS-2) L = LSS-2
0477    2     IF (CFR(S(I),S(I-1))) GO TO 4
0478          IF (CFR(S(I),S(I+1))) GO TO 4
0479          IF (CFR(S(I),H)) GO TO 4
```

```
0480          IF (M.GE.60) RETURN
0481          M = M+1
0482          N(M) = I
0483          I = I+1
0484    4     I = I+1
0485          IF (I.LE.L) GO TO 2
0486          RETURN
0487          END
```

```
**  NO ERRORS**     PROGRAM = 00132       COMMON = 00887
```

```
0488          SUBROUTINE ORDO(X,N,S,LOP,IF)
0489          LOGICAL LOP
0490          INTEGER X(1)
0491          DIMENSION S(1)
0492          IF (N.LT.2) RETURN
0493          L = N-1
0494    1     I = 0
0495          K = 1
0496    2     J = K+1
0497          IF (IF.EQ.0) GO TO 5
0498          IK = X(K)
0499          IJ = X(J)
0500          IF (LOP(IK,IJ).OR.IK.EQ.IJ) GO TO 3
0501          GO TO 6
0502    5     A = S(X(K))
0503          B = S(X(J))
0504          IF (LOP(A,B).OR.A.EQ.B) GO TO 3
0505    6     M = X(K)
0506          X(K) = X(J)
0507          X(J) = M
0508          I = K
0509    3     IF (K.GE.L) GO TO 4
0510          K = K+1
0511          GO TO 2
0512    4     IF (I.EQ.0) RETURN
0513          L = I-1
0514          GO TO 1
0515          END
```

```
**  NO ERRORS**     PROGRAM = 00160       COMMON = 00000
```

```
0516          SUBROUTINE ELIM(IP,NP)
0517          DIMENSION IP(1)
0518          I = 0
0519    1     I = I+1
0520          IF (I.GE.NP) RETURN
0521          IF (IP(I).NE.IP(I+1)) GO TO 1
0522          NP = NP-1
0523          DO 2 J=I,NP
0524    2     IP(J) = IP(J+1)
0525          GO TO 1
0526          END
```

```
**  NO ERRORS**     PROGRAM = 00067       COMMON = 00000
```

```
0527          SUBROUTINE FIND(IBP,K,N,IP,NP,S,Q,A,B,MP)
0528          EXTERNAL INCR,IDEC,IGT,ILT
0529          DIMENSION IP(1),S(1),Q(1)
0530          COMMON NX(986),LSS,NY,NZ,IOPT,NW(6),F
0531          IBP = 0
0532    4     IF (NP.EQ.0) RETURN
0533          K = IP(1)-1
0534          N = K+2
0535          CALL FIT2R(K,N,A,B,Q)
0536          IF (A.LE.0.) GO TO 5
0537          I = -B/A
0538          IF (I.LT.K.OR.I.GT.N) GO TO 5
```

```
0539          CALL CHSQ(K,N,A,B,Q,CK)
0540          SIG = SQRT(2./A)
0541          IF (MP.GT.0) GO TO 3
0542          CH = F*(CK+0.001)
0543          SGM = 2.5*SIG
0544    3     IF (CK.GT.CH) GO TO 5
0545          IF (SIG.GT.SGM) GO TO 5
0546          IBP = 1
0547    5     NP = NP-1
0548          IF (NP.EQ.0) GO TO 7
0549          DO 6 I=1,NP
0550    6     IP(I) = IP(I+1)
0551    7     IF (IBP.EQ.0) GO TO 4
0552          IF (IOPT.EQ.0) RETURN
0553          M = K-IOPT
0554          IF (M.LT.1) M = 1
0555          CALL OPT3(IDEC,ILT,CH,A,B,K,N,M,Q)
0556          M = N+IOPT
0557          IF (M.GT.LSS) M = LSS
0558          CALL OPT3(INCR,IGT,CH,A,B,N,K,M,Q)
0559          RETURN
0560          END

**  NO ERRORS**     PROGRAM = 00239       COMMON = 00898


0561          SUBROUTINE CHSQ(N,M,A,B,Q,C)
0562          DIMENSION Q(1)
0563          C = 0.
0564          DO 1 I=N,M
0565    1     C = C+(Q(I)-A*I-B)**2
0566          C = C/(M-N+1)
0567          C = SQRT(C)
0568          RETURN
0569          END

**  NO ERRORS**     PROGRAM = 00085       COMMON = 00000


0570          SUBROUTINE OPT3(IOP,LOP,CH,A1,B1,K,KK,M,Q)
0571          LOGICAL LOP
0572          DIMENSION Q(1)
0573          N = K
0574    1     N = IOP(N)
0575          IF (LOP(N,M)) RETURN
0576          IK = MINO(N,KK)
0577          MK = MAXO(N,KK)
0578          CALL FIT2R(IK,MK,A,B,Q)
0579          CALL CHSQ(IK,MK,A,B,Q,HC)
0580          IF (HC.GT.CH) RETURN
0581          IF (A.LE.0.) RETURN
0582          I = -B/A
0583          IF ((I-K)*(I-KK).GT.0) RETURN
0584          K = N
0585          A1 = A
0586          B1 = B
0587          GO TO 1
0588          END

**  NO ERRORS**     PROGRAM = 00116       COMMON = 00000
```

```
0589          SUBROUTINE FIT1R(I1,I2,AK1,ESP)
0590          DIMENSION ESP(1)
0591          COMMON AN(176),ROW1(176)
0592          C1 = 0.
0593          C2 = 0.
0594          DO 1 I=I1,I2
0595          C1 = C1+ROW1(I)*ESP(I)
0596    1     C2 = C2+ROW1(I)**2
0597          AK1 = C1/C2
0598          RETURN
0599          END
```

```
**  NO ERRORS**     PROGRAM = 00080       COMMON = 00704
```

```
0600          SUBROUTINE FIT2R(I1,I2,AK1,AK2,Q)
0601          INTEGER ROW2
0602          DIMENSION Q(1)
0603          COMMON AN(176),ROW1(176),ROW2(176)
0604          C1 = 0.
0605          C2 = 0.
0606          C3 = 0.
0607          A = 0.
0608          B = 0.
0609          DO 1 I=I1,I2
0610          C1 = C1+ROW1(I)**2
0611          C2 = C2+ROW1(I)*ROW2(I)
0612          C3 = C3+ROW2(I)**2
0613          A = A+ROW1(I)*Q(I)
0614    1     B = B+ROW2(I)*Q(I)
0615          DELTA = C1*C3-C2**2
0616          DX1 = B*C1-A*C2
0617          DX2 = A*C3-B*C2
0618          AK1 = DX1/DELTA
0619          AK2 = DX2/DELTA
0620          RETURN
0621          END
```

```
**  NO ERRORS**     PROGRAM = 00217       COMMON = 00880
```

```
0622          SUBROUTINE STRIP(A,LI,LS,IBP,S)
0623          DIMENSION S(1)
0624          COMMON AN(176),ROW1(176),M(182),LSS
0625          DO 23 I=1,LSS
0626    23    ROW1(I) = ROW1(I)*A
0627          SK = SQRT(S(IBP))
0628          LI = 0
0629          LS = 0
0630          DO 26 I=IBP,LSS
0631          IF (ROW1(I).GT.SK) GO TO 26
0632          LS = I
0633          GO TO 27
0634    26    CONTINUE
0635    27    DO 28 I=IBP,1,-1
0636          IF (ROW1(I).GT.SK) GO TO 28
0637          LI = I
0638          GO TO 29
0639    28    CONTINUE
0640    29    IF (LI.EQ.0) LI = 1
0641          IF (LS.EQ.0) LS = LSS
0642          DO 16 I=LI,LS
0643          S(I) = S(I)-ROW1(I)
0644          IF (S(I).LT.1.) S(I) = 1.
0645    16    CONTINUE
0646          RETURN
0647          END
```

```
**  NO ERRORS**     PROGRAM = 00183       COMMON = 00887
```

```
0648          SUBROUTINE SOMGS(NG,LSS,L,CK,SI,A,S)
0649          DIMENSION SI(1),A(1),CK(1),S(1),L(1)
0650          DO 1 K=1,LSS
0651          H = 0.
0652          DO 2 J=1,NG
0653          IF (A(J).EQ.0.) GO TO 2
0654          H = H+GAUSS(K,SI(J),A(J),CK(J))
0655    2     CONTINUE
0656    1     S(K) = H+L(K)
0657          RETURN
0658          END
```

```
**  NO ERRORS**     PROGRAM = 00102       COMMON = 00000
```

```
0659          SUBROUTINE FARBE(LU,N1,N2,NG,CK,SI,AK,SMIN,SMAX,S,SS,L,IC)
0660          DIMENSION S(1),H(30),CK(1),SI(1),AK(1),SS(1),L(1),IC(1)
0661          COMMON AN(176),ROW1(176),IOW2(176),LENF,NPAR
0662          DO 1 J=1,30
0663    1     H(J) = -1.
0664          DO 2 K=N1,N2
0665          X = SMAX-SMIN
0666          H(28) = (L(K)-SMIN)/X
0667          H(29) = (SS(K)-SMIN)/X
0668          H(30) = (S(K)-SMIN)/X
0669          DO 3 J=1,NG
0670          Y = GAUSS(K,SI(J),AK(J),CK(J))
0671          Y = Y+L(K)
0672          H(J) = (Y-SMIN)/X-0.0082
0673          IF (H(J).LE.H(28)) H(J) = -1.
0674    3     CONTINUE
0675          J = K+NPAR
0676    2     CALL PLOT(LU,H,J,SMIN,SMAX,IC)
0677          RETURN
0678          END
```

```
**  NO ERRORS**     PROGRAM = 00281       COMMON = 00882
```

```
0679          SUBROUTINE PLOT(LU,H,K,SMIN,SMAX,IC)
0680          DIMENSION H(1),IC(1)
0681          COMMON AN(176),JP(125),NC,I,J
0682          DATA NCAR/122/
0683          NC = 1
0684          DO 1 I=1,NCAR
0685    1     JP(I) = 1H
0686          DO 2 I=1,30
0687          J = H(I)*NCAR+1
0688          IF (J.LT.1) GO TO 2
0689          IF (J.GT.NC) NC = J
0690          JP(J) = IC(I)
0691    2     CONTINUE
0692          WRITE (LU,10) K,(JP(I),I=1,NC)
0693   10     FORMAT (1X,I4,125A1)
0694          RETURN
0695          END
```

```
**  NO ERRORS**     PROGRAM = 00123       COMMON = 00480
```

```
0696          SUBROUTINE PRISP(LU,MP,L,S,Q,W)
0697          INTEGER W
0698          DIMENSION L(1),S(1),Q(1)
0699          COMMON AN(176),NX(534),LSS
0700          IF (MP.GT.0) GO TO 26
0701          WRITE (W,102)
0702   102    FORMAT (/,7X,"SPECTRUM",/)
0703          WRITE (W,108) (AN(I),I=1,LSS)
0704   108    FORMAT (10(2X,F10.0))
0705          WRITE (W,106)
0706   106    FORMAT (/,7X,"BACKGROUND",/)
```

```
0707          WRITE (W,111) (L(I),I=1,LSS)
0708   111    FORMAT (10(2X,I10))
0709   26     IF (LU.LT.4) RETURN
0710          WRITE (W,107)
0711   107    FORMAT (/,7X,"SUBTR. SPECTRUM",/)
0712          WRITE (W,108) (S(I),I=1,LSS)
0713          WRITE (W,109)
0714   109    FORMAT (/,7X,"LOGARITHM",/)
0715          Z = 0.
0716          WRITE (W,110) Z,(Q(I),I=2,LSS-1),Z
0717   110    FORMAT (10(2X,F10.3))
0718          RETURN
0719          END

**  NO ERRORS**     PROGRAM = 00236       COMMON = 00887


0720          FUNCTION GAUSS(I,SI,A,CK)
0721          IF (A.EQ.0.) GO TO 1
0722          E = I-CK
0723          E = (E**2)/(2*SI**2)
0724          IF (E.GT.100.) GO TO 1
0725          GAUSS = A*EXP(-E)
0726          RETURN
0727   1      GAUSS = 0.
0728          RETURN
0729          END

**  NO ERRORS**     PROGRAM = 00089       COMMON = 00000


0730          FUNCTION INCR(N)
0731          INCR = N+1
0732          RETURN
0733          END

**  NO ERRORS**     PROGRAM = 00013       COMMON = 00000


0734          FUNCTION IDEC(N)
0735          IDEC = N-1
0736          RETURN
0737          END

**  NO ERRORS**     PROGRAM = 00014       COMMON = 00000


0738          LOGICAL FUNCTION ILT(N,M)
0739          ILT = N.LT.M
0740          RETURN
0741          END

**  NO ERRORS**     PROGRAM = 00015       COMMON = 00000


0742          LOGICAL FUNCTION IGT(N,M)
0743          IGT = N.GT.M
0744          RETURN
0745          END

**  NO ERRORS**     PROGRAM = 00015       COMMON = 00000
```

```
0746          LOGICAL FUNCTION LTN(A,B)
0747          LTN = A.LT.B
0748          RETURN
0749          END

**  NO ERRORS**     PROGRAM = 00016       COMMON = 00000


0750          LOGICAL FUNCTION GTN(A,B)
0751          GTN = A.GT.B
0752          RETURN
0753          END

**  NO ERRORS**     PROGRAM = 00016       COMMON = 00000


0754          SUBROUTINE SALTO(V)
0755          INTEGER V
0756          INP = 015414B
0757          WRITE (V,1) INP
0758    1     FORMAT (A2)
0759          RETURN
0760          END

**  NO ERRORS**     PROGRAM = 00025       COMMON = 00000
```