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

\[
D(J-1) > D(J) \quad \text{and} \quad D(J) < D(J+1)
\]

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

\[
S(J-1) < S(J) \quad \text{and} \quad S(J) > S(J+1)
\]

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).

Comparing with 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)

\[ D(I-1) > D(I) \text{ and } D(I) < D(I+1) \]
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)+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

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

where \( I_o \) is the centre of the peak, \( \sigma \) its standard deviation and \( P \) its area. Then the function

\[ f(I) = \ln \left[ \frac{S(I-1)}{S(I+1)} \right] \]

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

\[ \ln \left[ \frac{G(I-1)}{G(I+1)} \right] = \frac{2(I-I_o)}{\sigma^2} \]  

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

\[ Y(I) = A . I + B \]  

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 \( \sigma \) are then obtained from the coefficients \( A \) and \( B \) according to the formulae

\[ I_o = -\frac{B}{A} \quad \quad \sigma = \sqrt{\frac{2}{A}} \]  

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 $I_0$, 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

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

is calculated (subroutine CHSQ) with $M = J-1$, $N = J+1$. The value $C_0$ 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 $C_0$ is then multiplied by a constant factor $F$ to give the quantity $E = C_0F$. 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 SOWGS) 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 \( \gamma \)-peak (the 662 keV \( \gamma \)-ray from a \(^{137}\text{Cs} \) 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'γ)$^{93}$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 µg/cm\(^2\) carbon backing. The FPs 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 FPs 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

(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.
(5) M.A. Mariscotti, Nucl. Instr. and Meth. 50 (1967) 309.
(9) D.C. Robinson, Nucl. Instr. and Meth. 78 (1970) 120.
(10) W. Teoh, Nucl. Instr. and Meth. 109 (1973) 509.
(17) Supplied by Canberra, model 721 K.
(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,γ)$^{127}$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,γ)$^{93}$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.
\( ^{92} \text{Nb (n, n')} ^{92} \text{Nb} \)

\( E_n = 2.7 \text{MeV} \)

\( \theta = 90^\circ \)
\[^{68}\text{Zn} \,(pp)^{68}\text{Zn}\]
\[E_p = 2.8 \text{ MeV}\]
\[\theta = 145^\circ\]
$^{68}\text{Zn} \left(^{pp}\right)^{68}\text{Zn}$

$E_p = 3.9 \text{ MeV}$

$\theta = 90^\circ$

fig. 7
APPENDIX

0001 FTH4+L
0002 C
0003 C PROGRAM LIZA
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 \[ L+\text{KFORM}+\text{KTAP}+\text{FILE}+\text{TITLE} \]
0010 C \[ \text{E FORMAT (31S,}+3A2,+4X,+1A2) \]
0011 C THE SECOND CARD CONTAINS:
0012 C \[ \text{LENF}+\text{NPAR}+\text{LSG}+\text{IOVF}+\text{KP}+\text{ IP}+\text{ZERO}+\text{CDR}+\text{F}+\text{T} \]
0013 C \[ \text{E FORMAT (71S,}+5X,+AE10.0) \]
0014 C
0015 C DESCRIPTION OF THE PARAMETERS
0016 C
0017 C LU CONTROLS THE OUTPUT IN STEPS OF INCREASING COMPLETENESS
0018 C
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 BY PRESSING 1 ON THE KEYBOARD OF THE VIDEODISPLAY
0023 C \[ LU = 1 \] THE ZONE OF THE SPECTRUM WHICH CONTAINS THE CHOSEN PEAKS
0024 C ARE PRINTED ON THE LINE PRINTERS.
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 PRINTED
0028 C \[ LU = 4 \] THE CONTENTS OF THE SPECTRUM AFTER SUBTRACTION OF THE BACK-
0029 C GROUND AND THE VALUES OF THE FUNCTION \[ \log(4(S-I)/S(I+1)) \]
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(F4.0,+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 RININ
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 2 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
0063 C THE MAXIMUM VALUE OF KP IS 26
0064 C
0065 C IOPT CONTROLS THE OPTIMIZATION OPTIONS:
0066 C \[ IOPT = 0 \] THE OPTIMIZATION ROUTINE IS NOT ACTIVATED
0067 C \[ IOPT = N \] THE OPTIMIZATION ROUTINE IS ACTIVATED WITH A MAXIMUM
0068 C NUMBER OF POINTS EQUAL TO \[ N \]
0069 C
0070 C ZERO (KEV) IS THE VALUE OF THE ENERGY SCALE AT THE CHANNEL
0071 C
0072 C ZERO OF THE SPECTRUM
0022 C CONV (NEW/CHANNEL) IS THE CONVERSION FACTOR OF THE ENERGY SCALE
0023 C
0024 C F IS A FACTOR WHICH AFFECTS THE ACCEPTABILITY OF PEAKS RELATING
0025 C TO THE FIRST FIT IN EACH SEGMENT OF THE SPECTRUM
0026 C F (>1) SHOULD BE DETERMINED EXPERIMENTALLY (SEE SECTION 3.4)
0027 C
0028 C T IS THE HEIGHT BELOW WHICH PEAKS ARE REJECTED
0029 C
0030 C
0031 C PROGRAM IFS
0032 C EXTERNAL GAUS S, IGT, ILT, OTH, LN T
0033 C INTEGER P, K, V, W, T APE, RO U2
0034 C DIMENSION ST(26), AK(26), CK(26), L(176), S(176), Q(176)
0035 C DIMENSION CC(26), EN(26), FW(26), SA(26), ER(26), AS(26), LP(26)
0036 C DIMENSION IN(26), HH(26), KK(26), RN(26), IP(60), IC(30)
0037 C COMMON AN(176), ROMI(176), ROU2(176)
0038 C COMMON LSPG, IOPT, ITP, ZEROG, CONVF, T, FILE(3), KT AP
0039 C DATA IC/IH(I+1), IH(I+2), HH(I+1), HH(I+2), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1)
0040 CIH(I+1), IH(I+2), IH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1), HH(I+2), HH(I+1)
0042 C DATA V/I/K/R/S/U/W/X/TAPE/7/P/A/
0043 C CALL DATA(R)
0044 3 IF (IPF, ET, 0) GO TO 1
0045 C IF (NPAR.EQ.0) GO TO 1
0046 C CALL R-D(IN, .TRUE., NP A, TK, TAP)
0047 1 IF (NK.EQ.0) GO TO 3
0048 C KP = MIN(KP, 26)
0049 FK = 0.
0050 D0 5 LAU = L + NSG
0051 IF (LAU.EQ.NSG) LSS = LS U
0052 IF (LAU.GT.1) GO TO 4
0053 N A R R = N P A R + 4 L S S
0054 C KANAL = L S S
0055 IK = 1.
0056 DO 7 IK = 1, K
0057 C NPAIR = NPAIR-10VF
0058 C NPAIR = NPAIR+4LS S
0059 C KANAL = LSS-10VF
0060 IK = IOVF+1
0061 IF (IK.EQ.1) GO TO 7
0062 DO 8 I = 1, IOVF
0063 AN(I) = AN(I+LSG)
0064 C CALL R-D(IN, .FALSE., KAN AL, IK, TAP)
0065 MP = NPAIR+1
0066 CALL SALTIO(V)
0067 WRITE (V, 118) MP, NPAIR
0068 WRITE (W, 119) MP, NPAIR
0069 FORMAT(/"I5**, SEGMENT FROM CHANNEL"*I5**, TO CHANNEL"*I5**, /
0070 CALL EXTR(NK, LS S, MIN, S, MAX, HH+II)
0071 CALL SALTIO(W)
0072 C CUT = SQRT(SMAX)
0073 C IF (SMAX-CUT.LE.SMIN) GO TO 21
0074 C CUT = SQRT(SMAX)
0075 C CALL FONDO(W, HH+II, LS S, MP, IP, NP)
0076 IF (NP.EQ.2) GO TO 5
0077 N1 = 1.
0078 NZ = LS S
0079 IF (LGT.GT.2) GO TO 24
0080 NZ = 1.
0081 N1 = LS S
0082 24 WRITE (V, 104)
0083 104 FORMAT(/"FEAK","4X","CENTRE","4X","HEIGHT","4X","FWHM","4X","ENERGY","+
0084 "PS","ERROR","6X","AREA","5X","ERROR","8X","FIT CH.","2X","BASE CH.",")
0085 MP = 0.
0086 NP = 0.
0087 L = 1.
0088 LS = LS S
0089 20 CALL PREP(LTH, MP, NP, LS, LS, LS, LS, C U T)
0090 IF (NP.EQ.0) GO TO 2
0091 CALL ORDER(IP, NP, S, O R G, 0)
0092 CALL ELIM(IP, NP)
0093 22 IF (K.EQ.1) LS S = 1.
0094 15 R O W I (K) = K.
0095 ND 2 K = K-S, LS S= 1
0096 23 ND (K) = ALOG(K(K-1)/K(K+1)))

318
CALL FIND(TBP,K,N,IP,HF,S,0,A,H,MP)

IF (IP.EQ.0) GO TO 2

IF (LU.GE.2) CALL PRISPL(U,MP,L,S,0,W)

SIG = SQRT(2./A)

CTR = -A/4

IP = CTR

DO 11 I=1,LSS

KOU(I) = GAUSS(I*SIG,1.,CTR)

CALL FTIR(F,K+H,5)

IF (G,L,T,CUT) GO TO 22

HF = HF+1

CK(NP) = CTR

AK(NP) = A

S1(NP) = SIG

ER(NP) = SIG*CONV

IKK(NP) = K+H*HFAR

INN(NP) = K+H*HFAR

CC(NP) = CTR*HFAR

FK(NP) = CC(HF)*CONV+ZERO

FU(NP) = 2.355*ER(HF)

CALL STRIP(A,L,L,L,LS,1,IP,S)

A = 2.507*SIG*A

AS(NP) = A

DO 16 I=1,LSS

A = 60.*I/126

SA(NP) = SORT(A)

KK(NP) = LI*HFAR

HN(NP) = LS*HFAR

WRITE (V,105) HF,TC(HF),CC(HF),4K(HF),FU(HF),EN(HF)

PER(NP) = AS(NP)+SA(NP)+INN(HF)+INN(HF)+LK(NP)+F1(MF)

FORMAT (7X,12,1X,N1,7F10.2,5X,415/)

IF (N1.GT.LI) N1 = 1

IF (N2.GT.LI) N2 = 1

IF (MP.LT.KF) GO TO 20

IF (MP.NE.0) GO TO 4

WRITE (4,121)

WRITE (4,121)

105 FORMAT (/,'NO PEAKS',/)

121 FORMAT (/,'NO PEAKS',/)

122 FORMAT ('CHOOSE YOUR PEAKS',/)

123 FORMAT (/,"INPUT AB...YZ")

124 FORMAT (26A1)

125 IF (HF.NE.0) GO TO 21

126 WRITE (4,104)

127 CALL ORD0(IP,HF,S,1,LY,1)

128 DO 10 K=1,HP

129 DO 10 J=1,HP

130 IF (IP(J),EQ.IC(I)) GO TO 13

131 CONTINUE

132 HP = HF-1

133 IF (IP(J))=0 GO TO 13

134 CONTINUE

135 IF (HP.NE.0) GO TO 21

136 WRITE (4,105) K,TC(J),CC(J),AK(J),FU(J),EN(J),ER(J)

137 WRITE (4,105) K,TC(J),CC(J),AK(J),FU(J),EN(J),ER(J)

138 CONTINUE

139 CONTINUE

140 CONTINUE

141 CONTINUE

142 CONTINUE

143 CONTINUE

144 CONTINUE

145 CONTINUE

146 CONTINUE

147 CONTINUE

148 CONTINUE

149 CONTINUE

150 CONTINUE

151 CONTINUE

152 CONTINUE

153 CONTINUE

154 CONTINUE

155 CONTINUE

156 CONTINUE

157 CONTINUE

158 CONTINUE

159 CONTINUE

160 CONTINUE

161 CONTINUE

162 CONTINUE

163 CONTINUE

164 CONTINUE

165 CONTINUE

166 CONTINUE

167 CONTINUE

168 CONTINUE

169 CONTINUE

170 CONTINUE

171 CONTINUE

172 CONTINUE

173 CONTINUE

174 CONTINUE

175 CONTINUE

176 CONTINUE

177 CONTINUE

178 CONTINUE

179 CONTINUE

180 CONTINUE

181 CONTINUE

182 CONTINUE

183 CONTINUE

184 CONTINUE

185 CONTINUE

186 CONTINUE

187 CONTINUE

188 CONTINUE

189 CONTINUE

190 CONTINUE

191 CONTINUE

192 CONTINUE

193 CONTINUE

194 CONTINUE

195 CONTINUE

196 CONTINUE

197 CONTINUE

198 CONTINUE

199 CONTINUE

200 CONTINUE

201 CONTINUE

202 CONTINUE

203 CONTINUE

204 CONTINUE

205 CONTINUE

206 CONTINUE

207 CONTINUE

208 CONTINUE

209 CONTINUE

210 CONTINUE

211 CONTINUE

212 CONTINUE

213 CONTINUE

214 CONTINUE

215 CONTINUE

216 CONTINUE

217 CONTINUE

218 CONTINUE

219 CONTINUE

220 CONTINUE

221 CONTINUE

222 CONTINUE

223 CONTINUE
IF (LU.EQ.0) GO TO 12
0225 WRITE (W,126)
0226 126 FORMAT (/"TO GET A PRINTED PLOT PRESS 1")
0227 READ (U,*) HP
0228 IF (HP.NE.1) GO TO 5
0229 CONTINUE
0230 CALL SMOBS(NF,LSS,LX,SH,SI,AK,S)
0231 CALL EXTR(S,NF,SH,SM,SMAX)
0232 CALL FARRB(NH,HP,LX,SH,SI,SHM,AN,S,L,IC)
0233 IF (LU.LT.2) GO TO 5
0234 WRITE (W,112)
0235 112 FORMAT (/"SECOND APPROX.",/)
0236 WRITE (W,100) (S(I), I=1,LSS)
0237 100 FORMAT (10C6X,S1)
0238 5 CONTINUE
0239 IF (KTP.NE.0) WRITE (F,103) PK,PK,PK
0240 KP = PK
0241 WRITE (W,105) KP
0242 105 FORMAT (7X,"PEAKS HAVE BEEN LISTED",/)
0243 IF (KFORM.EQ.2) GO TO 3
0244 IF (NARR.EQ.LENF) GO TO 3
0245 KAHAL = LENS-NARR
0246 CALL RIDE(.TRUE.,KAHAL,IK,TAPE)
0247 GO TO 3
0248 END

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

SURROUTINE DATA(R,W)
0249 INTEGER ROWS,6,J
0250 DIMENSION TITLE(15)
0251 COMMON A(N),R041(176),R0M2(176)
0252 COMMON LENF,NPAR,LSG,IOVF,NOK,HS=1,LSS,LUS,KFORM
0253 COMMON IOPT,LU,KP,ZERO,CONV,F,T,IFILE(3),KTAPE
0254 COMMON 0.01 J = 1,15
0255 0.01 FORMAT (33S,3A2,4X,15A2)
0256 210 TITLE(I) = 2M
0257 READ (R,201) LU,KFORM,KTAPE,(IFILE(I),I=1,3),(ITITLE(I),I=1,15)
0258 201 FORMAT (33S,3A2,4X,15A2)
0259 200 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 (7155,5X,4E10.0)
0264 IF (KFORM.EQ.2) WRITE (W,107) (IFILE(I),I=1,3)
0265 102 IF (KFORM.EQ.1) READ FROM "",3(4,2)/
0266 (FILES,3A2)*R
0267 NPAR = (NPAR/3)*R
0268 LSG = (LSG/3)*R
0269 IOVF = (IOVF/3)*R
0270 WRITE (W,110) LENS
0271 110 FORMAT (7X,"SPECTRUM OF",15," CHANNELS",/)
0272 WRITE (W,109) KP
0273 109 FORMAT (7X,"MAXIMUM","I", "PEAKS PER SEGMENT",/)
0274 NOK = NARR-NPAR
0275 NSG = (NOK-IOVF)/LSG
0276 IF (NOK+HS=1,LSS) NSG = NSG+1
0277 LSS = LSG+IOVF
0278 LUS = NOG=1-LSG
0279 WRITE (W,205) ZERO,CONV,F,F,F
0280 205 FORMAT (7X,"ZERO","F",F",F","F","F","F","F")
0281 WRITE (W,108) IOPT
0282 108 FORMAT (7X,"IOPT","I",/)
0283 RETURN
0284 99 WRITE (W,103)
0285 103 FORMAT (5X,"LASTA COSI")
0286 STOP
0287 END

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

320
SUBROUTINE R1651(M,N,KAVAL,IK,T)
INTEGE T
COMMON 4569(170),D,(255),A(6),L,F,FPAR,NX(6),KFORM,NY(11),I(FILE(3)
LOGICAL SAKB
IF (KFORM.EQ.2) GO TO 6
N = KAVAL/G
IF (N.LT.1) RETURN
DO 1 M=1,N
IF (KFORM.EQ.0) GO TO 2
READ (T,5) (AA(K),K=1,8)
FORMAT (B(F6,0,1X))
GOTO 3
DO 4 K=1,8
IF (AA(K).LT.1.) AA(K) = 1.
4
ANIK) = AA(K)
IF (IK) = IK+1
CONTINUE
RETURN
6
NF= NF+1K
N = NF*2/128
N = NF+KAVALL
N = (N-N+64)*2
CALL EXEC (14,102,6,2,16,FILE,M)
M = NF*1664
N = M+KAVAL
DO 7 K=1,N
7
IF (DK,L.T.1.) D(K) = 1.
ANIK) = DK
IF (IK) = IK+1
CONTINUE
RETURN
END

*NO ERRORS** PROGRAM 00268 COMMON 00903

SUBROUTINE FORDIFFA(N,HH,L,S,JIF,NF)
EXTERNAL GH
INTEGER N
DIMENSION IF(1),S(1),L(1)
COMMON HH(170),NX(25),NF,MY(6),LSS
Hf = 0
LI = 12
LS = LSS-11
CALL PREP (GTH,IF,NF,LI,LS,AN,HH)
CALL FLAT (IF,NF,L,LS)
IF (NF.LT.2) RETURN
CALL CHOSX(NF,F)
IF (NF.LT.2) RETURN
CALL CLEAR (NF)
IF (NF.LT.2) RETURN
DO 20 I=1,NF
I = IF(I-1)
LS = IF(I)
A = AN(1S)
A = A+SORT(A)
B = AN(LI)
B = B+SORT(B)
A = (A-B)/FLOAT(LS-1I)
R = B-A
IF (L.EQ.2) LI = 1
IF (L.EQ.NF) LS = LSS
DO 31 K=1,NF
L(K) = A*K+B
31 CONTINUE
DO 30 J=1,LSS
30 CONTINUE
** NO ERRORS**  PROGRAM = 00331   COMMON = 00337

0356   DO 24 I=1,NF
0357   24 IF(I) = IF(I)+NPAR
0358   WRITE (W,J1) (IF(I),I=1,NF)
0359   117 FORMAT (7X,"NACKGROUND (H.",10IX)
0360   RETURN
0361   END

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

0362   SUBROUTINE F(LT,F,L,LF)
0363   INTEGER F(1)
0364   COMMON AN(176)
0365   IF (NF.FE.60) RETURN
0366   NI = NF
0367   NT = 1
0368   L5 = LI+4
0369   IF (L5.GT.LF) RETURN
0370   IF (K.GT.NF1) GO TO 5
0371   I = NT
0372   IF (F(I).GE.LI) GO TO 3
0373   I = I+1
0374   IF (I.LE.NF1) GO TO 2
0375   GO TO 4
0376   NT = I+1
0377   IF (F(I).LE.LS) GO TO 8
0378   NT = I
0379   5 SUM = 0.
0380   DO 5 I=1,LS
0381   6 SUM = SUM+AN(I)
0382   SUM = SUM/N
0383   SS = SORT(SUM)
0384   SI = SUM+SS
0385   SS = SUM+SS
0386   DO 7 I=1,LS
0387   IF (AN(I).LT.SI) GO TO 8
0388   IF (AN(I).LT.SS) GO TO 8
0389   7 CONTINUE
0390   NF = NF+1
0391   F(NF) = LT+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 FCHC(NF,IF)
0397   EXTERNAL LIN,ILT
0398   DIMENSION IF(1)
0399   COMMON AN(176),TAB(514).LS
0400   CALL ORDO(IF,NF,.AN,ILT,1)
0401   N = 5
0402   7 IN = LS/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   9 CONTINUE
0411   10 IF (I.GE.3) GO TO 6.
0412   N = N-1
0413   IF (N.GT.3) GO TO 7
0414   GO TO 5
0415   NI = J
0416   6 LS = LS+1
0417   8 LS = LS+IN
0418   5 CALL ORDO(IF,NF,.AN,ILT,N)
0419   0 IN = LS/N
0420   I = 0
**NO ERRORS**

PROGRAM = 0013

COMMON = 0028

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

**NO ERRORS**

PROGRAM = 00160

COMMON = 0000

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

**NO ERRORS**

PROGRAM = 00067

COMMON = 0000

0527 SUBROUTINE FIND(IFP,K,N,IP,NP,S,O,A,B,NP)
0528 EXTERNAL INC,DEC,IGT,ILT
0529 COMMON NX(364),LS,NS,NZ,ICTP,NN(6),I
0530 INC = 0
0531 DEC = 0
0532 IF (IP(K).EQ.0) RETURN
0533 K = K+1
0534 N = K+2
0535 CALL FIT2R(K,N,A,B,0)
0536 IF (A.EQ.0.) GO TO 5
0537 I = -B/A
0538 IF (I.LT.K.OR.I.GT.N) GO TO 5
CALL CHSA(K+M,A,B,Q,CK)

SIG = SORT(A)

IF (NP.GT.0) GO TO 3

CH = F*(K+0.001)

SGM = 2.5*SIG

IF (CH.GT.CH) GO TO 5

IF (SIG.GT.SGM) GO TO 5

IBP = 1

NP = NP-1

IF (NP.EQ.0) GO TO 7

DO 4 I=1,NP

1P(I) = IF(I+1)

IF (I+1.EQ.NP) GO TO 4

IF (TOFT.EQ.0) RETURN

M = K-TOFT

IF (M.LT.1) M = 1

CALL OPT3(INCR,CH,A,B,R,K,N,M,0)

M = H+TOFT

IF (M.GT.LSS) M = LSS

CALL OPT3(INCR,GT,CH,A,B,R,K,N,M,0)

RETURN

END

** NO ERRORS**  PROGRAM = 00239  COMMON = 0693B

SUBROUTINE CHSA(N,H,A,B,Q,C)

DIMENSION Q(1)

C = 0.

DO 1 I=H,N

1 C = C+(I(I)-A*I-B)**2

C = C/<M-H+1>

RETURN

END

** NO ERRORS**  PROGRAM = 00035  COMMON = 00000

SUBROUTINE OPT3(IOP,LOP,CH,A,B,1,K,KK,M,N)

LOGICAL LOP

DIMENSION Q(1)

N = K

1 N = IOP(N)

IF (IOP(H,M)) RETURN

IK = HMAX(H+M)

MK = HMAX(N+KK)

CALL FIT2R(IK,MK,A,B,0)

CALL CHSA(IK,MK,A,B,0)

IF (H.GT.C) RETURN

IF (A.EQ.0.) RETURN

I = -B/A

IF ((I-N)*(I-KK),GT,0) RETURN

K = N

AL = A

R1 = B

GO TO 1

END

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

325
0520       SUBROUTINE FITR(I1,I2,AK1,ESP)
0521       DIMENSION ESP(I)
0522       COMMON AN(176),ROW1(176)
0523       CI = 0.
0524       C2 = 0.
0525       DO 1 I=I1,I2
0526       1   CI = CI+ROW1(I)*ESP(I)
0527       C2 = C2+ROW1(I)**2
0528       AK1 = CI/C2
0529       RETURN
0530       END

** NO ERRORS**  PROGRAM   =   00080   COMMON   =   00080

0600       SUBROUTINE FIT2R(I1,I2,AK1,AK2,0)
0601       INTEGER ROW2
0602       DIMENSION 0(I)
0603       COMMON AN(176),ROW1(176),ROW2(176)
0604       CI = 0.
0605       C2 = 0.
0606       C3 = 0.
0607       A = 0.
0608       B = 0.
0609       DO 1 I=I1,I2
0610       CI = CI+ROW1(I)**2
0611       C2 = C2+ROW1(I)*ROW2(I)
0612       C3 = C3+ROW2(I)**2
0613       A = A+ROW1(I)*0(I)
0614       1   R = R+ROW2(I)*0(I)
0615       DELTA = C1*C3-C2**2
0616       DX1 = R*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   =   00080

0622       SUBROUTINE STRIP(A,LI,LS,IMF,P)
0623       DIMENSION S(I)
0624       COMMON AN(176),ROW1(176),HI(182),LS
0625       DO 23 I=1,LS
0626       ROW1(I) = ROW1(I)*A
0627       SK = SQRT(S(I)**P)
0628       LI = C
0629       LS = 0
0630       DO 26 I=IMF,LS
0631       IF (ROW1(I).LT.5K) GO TO 26
0632       LS = I
0633       GO TO 27
0634       26 CONTINUE
0635       27   DO 28 I=IMF,1,-1
0636       IF (ROW1(I).LT.5K) 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 = LS
0642       DO 16 I=1,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   =   00087
** SUBROUTINE SOMGS(NH; LSS, L, CK; SI, A, S) **
** DIMENSION SI(I), CK(I), S(I), L(I) **
** DO 1 K=1, LSS **
** H = 0. **
** DO 2 J=1, NH **
** IF (A(J), LEO, O.) GO TO 2 **
** H = H + GAUSS(K; SI(J), A(J), CK(J)) **
** 2 CONTINUE **
** S(K) = H + L **
** RETURN **
** END **

** SUBROUTINE FARG(EU; MJ1, MJ2, NH; CK; SI, AK, SMIN, SHAX, S, SS, L, IC) **
** DIMENSION SI(I), HX(Max), CK(I), SI(I), AK(I, SS), L(I), IC(I) **
** COMMON AN(176), R0HU(176), I0HU(176), LEHF, HPAR **
** DO 1 J=1, 30 **
** 1 H(J) = -1. **
** DO 2 K=1, MJ2 **
** X = SMAX - SMIN **
** H(20) = (L(K) - SMIN) / X **
** H(21) = (SS(K) - SMIN) / X **
** H(30) = (S(K) - SMIN) / X **
** DO 3 J=1, NH **
** Y = GAUSS(K, SI(J), AK(J), CK(J)) **
** 3 CONTINUE **
** Y = H(J) **
** Y = Y*SMAX / S - 0.0032 **
** IF ((H(J), L. H(20)) H(J) = -1. **
** RETURN **
** END **

** SUBROUTINE PLOT(LU; NH, JK, SMIN, SHAX, IC) **
** DIMENSION H(I), IC(I) **
** COMMON AN(176), R0FJ(125), I1J **
** DATA RCAR/122/, **
** NC = 1 **
** DO 1 I=1, NCAR **
** 1 IF (I, L, NCAR) GO TO 2 **
** IF (IP(I), 1) IF (IP(I), 1, 1) **
** IF (J, G, J) NC = J **
** JF(J) = IC(I) **
** 2 CONTINUE **
** WRITE (LU, 10) I, JF(I), 1 **
** 10 FORMAT (1X, I4, 125A1) **
** RETURN **
** END **

** SUBROUTINE PRISP(LU, HP, L, S, O, W) **
** INTEGER W **
** DIMENSION L(I), S(I), O(I) **
** COMMON AN(176), RX(354), LSS **
** IF (I, P, GT, O.) GO TO 26 **
** WRITE (W, 102) **
** 102 FORMAT (/X, "SPECTRUM") **
** WRITE (W, 103) (AN(I), 1, I=1, LSS) **
** 103 FORMAT (/X, "BACKGROUND") **
** WRITE (W, 104) (OUT(I), 10, I=1, LSS) **
** 104 FORMAT (/X, "FACE") **
** WRITE (W, 105) **
** 105 FORMAT (/X, "BACK") **
** WRITE (W, 106) **
** 106 FORMAT (/X, "BACKGROUND") **

00448 00449 00450 00451 00452 00453 00454 00455 00456 00457 00458 00459 00460 00461 00462 00463 00464 00465 00466 00467 00468 00469 00470 00471 00472 00473 00474 00475 00476 00477 00478 00479 00480 00481 00482 00483 00484 00485 00486 00487 00488 00489 00490 00491 00492 00493 00494 00495 00496 00497 00498 00499 00500 00501 00502 00503 00504 00505 00506 ** NO ERRORS ** PROGRAM = 00102 COMMON = 00000 ** NO ERRORS ** PROGRAM = 00281 COMMON = 00362 ** NO ERRORS ** PROGRAM = 00123 COMMON = 00490 ** NO ERRORS ** PROGRAM = 00102 COMMON = 00000 ** NO ERRORS ** PROGRAM = 00281 COMMON = 00362 ** NO ERRORS ** PROGRAM = 00123 COMMON = 00490 ** NO ERRORS ** PROGRAM = 00102 COMMON = 00000 ** NO ERRORS ** PROGRAM = 00281 COMMON = 00362 ** NO ERRORS ** PROGRAM = 00123 COMMON = 00490 ** NO ERRORS ** PROGRAM = 00102 COMMON = 00000 ** NO ERRORS ** PROGRAM = 00281 COMMON = 00362 ** NO ERRORS ** PROGRAM = 00123 COMMON = 00490
** NO ERRORS**  PROGRAM = 00235  COMMON = 00887

0720  FUNCTION GAUSS(I,ST,A,CK)
0721  IF (A.EQ.0.) GO TO 1
0722  E = I-CK
0723  E = (EXP(-E)+2*ST**2)/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
0744  LOGICAL FUNCTION LTH(A,B)
0747      LTH = A.LT.B
0748      RETURN
0749      END

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

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

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

0754  SUBROUTINE SALT3(V)
0755      INTEGER V
0756      INF = 01541B
0757      WRITE (V+1) INF
0758      FORMAT (A2)
0759      RETURN
0760      END

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