PEAKFIND: A FORTRAN PROGRAM
FOR LOCATING AND FITTING PEAKS IN
SEMICONDUCTOR DETECTOR SPECTRA

W. C. Schick, Jr.

AMES LABORATORY, USERDA
IOWA STATE UNIVERSITY
AMES, IOWA

Date Transmitted: June 1975

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
UNDER CONTRACT W-7405-eng-82

DISTRIBUTION OF THIS DOCUMENT UNLIMITED
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
PEAKFIND: A FORTRAN PROGRAM FOR LOCALLING AND FITTING PEAKS IN SEMICONDUCTOR DETECTOR SPECTRA

W. C. Schick, Jr.

Ames Laboratory, ERDA
Iowa State University
Ames, Iowa 50010

Date Transmitted: June 1975

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION DIVISION OF RESEARCH UNDER CONTRACT NO. W-7405-eng-82

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

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

Available from: National Technical Information Service
U. S. Department of Commerce
P. O. Box 1553
Springfield, VA 22161

Price: Microfiche $2.25
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>vi</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. PEAK FINDING PROCEDURE</td>
<td>4</td>
</tr>
<tr>
<td>III. TO FIT OR NOT TO FIT</td>
<td>13</td>
</tr>
<tr>
<td>IV. PEAK FITTING PROCEDURE</td>
<td>17</td>
</tr>
<tr>
<td>V. PROGRAM OUTPUT</td>
<td>23</td>
</tr>
<tr>
<td>A. Printed output</td>
<td>23</td>
</tr>
<tr>
<td>B. Punched-card output</td>
<td>25</td>
</tr>
<tr>
<td>C. Graphical output</td>
<td>27</td>
</tr>
<tr>
<td>VI. PROGRAM INPUT</td>
<td>30</td>
</tr>
<tr>
<td>A. MCA data</td>
<td>30</td>
</tr>
<tr>
<td>B. Data control cards</td>
<td>31</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>38</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>39</td>
</tr>
<tr>
<td>APPENDIX A. PROGRAM DESCRIPTION</td>
<td>40</td>
</tr>
<tr>
<td>APPENDIX B. PROGRAM REQUIREMENTS AND LIMITATIONS</td>
<td>44</td>
</tr>
<tr>
<td>APPENDIX C. PROGRAM LISTING</td>
<td>46</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The shape of the smoothed second derivative in the vicinity of a Gaussian peak. The dashed line indicates the statistical uncertainty in the second derivative.</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Typical gamma-ray spectrum showing peaks identified by PEAKFIND when default options were used. Vertical scale is proportional to the square root of the number of counts.</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Preliminary output provided by PEAKFIND for the spectrum of Fig. 2, indicating the peaks found and the grouping of these peaks into multiplets.</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>Function used for peak fitting.</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Typical output for one fit, which includes peaks 21 and 22 of Figs. 2 and 3.</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>Summary output for fits performed as indicated in Fig. 3.</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>Graphical output for fit 3 of Fig. 6 (peak 5 of Figs. 2 and 3).</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>Graphical output for fit 9 of Fig. 6 (peaks 21 and 22 of Figs. 2 and 3). Output from this fit is also shown in Fig. 5.</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>Typical data deck, showing the control cards which produced the output shown in Figs. 3, 5, 6, 7, and 8.</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>Overall flow chart for PEAKFIND. The subprograms in which the various operations are performed are indicated in brackets on the right.</td>
<td>42</td>
</tr>
<tr>
<td>11</td>
<td>Flow chart for subroutine PKFIND.</td>
<td>43</td>
</tr>
</tbody>
</table>
LIST OF TABLES

I. Format of data control card A. 34
II. Format of data control card C. 35
III. Format of data control card D. 37
ABSTRACT

A FORTRAN program is described which locates peaks in semiconductor detector spectra and then fits a skewed-Gaussian line shape to certain of these peaks to determine centroids, areas, and peak shape parameters. The user may adjust the strictness of criteria used in the peak location procedure to permit the identification of virtually anything that might possibly be a peak, or he may specify that only well-defined peaks are to be found. An estimate is made of the centroid, energy, and intensity (corrected for the energy dependence of the detector efficiency) for each peak found. Fitting is then performed only where the estimated peak intensity exceeds a lower limit specified by the user. Alternatively, the user may specify which specific peaks he wishes to have fitted. The nonlinear least-squares fitting method is a modification of the procedure used in the companion program SKEWGAUS. PEAKFIND normally requires the user to supply only four input data cards for the analysis of a whole multichannel analyzer spectrum, and if default options are chosen for most parameters these cards will be largely blank.
# DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>Distribution Category</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>W. C. Schick</td>
<td>50</td>
</tr>
<tr>
<td>W. L. Talbert</td>
<td>50</td>
</tr>
<tr>
<td>Ames Laboratory Library</td>
<td>10</td>
</tr>
<tr>
<td>ERDA-COO</td>
<td>1</td>
</tr>
<tr>
<td>ERDA-TIC</td>
<td>27</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>138</strong></td>
</tr>
</tbody>
</table>
I. INTRODUCTION

PEAKFIND is a FORTRAN program used for finding and fitting peaks in multichannel analyzer (MCA) spectra obtained with semiconductor detectors. It first locates the peaks that are present in a spectrum and estimates their centroids and areas. It then fits an analytic function to the channels in the vicinity of certain of these peaks to determine (among other things) accurate centroids and areas. These centroids and areas may then be used by another program to determine transition energies and intensities.

The peak finding procedure is based on the method of Mariscotti. It employs an examination of the smoothed second difference between adjacent channels in the MCA spectrum to locate the peaks. The criteria for identifying a peak have deliberately been made very loose, so that the program can be instructed to find virtually everything that might possibly be a peak (except where adjacent peaks are not resolved from each other). The user may adjust both the amount of smoothing and the tightness of the main peak identification criterion if he wishes to change the balance between the probability of missing real peaks on one hand, and of finding statistical fluctuations that are not real peaks on the other hand. The design of PEAKFIND is based on the assumption that the user will consider the missing of real peaks to be a far more serious problem than the finding of spurious peaks. Since any spurious peaks found by the program will usually be very weak, the user may discriminate against them by specifying that only the more intense peaks are to be fitted. This allows the computer to avoid the wasting of time in attempting to fit statistical fluctuations or other spurious features, while at the same time the user is still alerted to the existence of features which might perhaps be real peaks. On a subsequent
execution of the program with the same MCA spectrum the user may specify which specific peaks he wishes to have fitted. This makes it possible to fit the weaker real peaks in the spectrum without wasting time refitting the stronger peaks.

The fitting procedure is very similar to that used in program SKEWGAUS. The fitting function is a Gaussian to which an exponential tail has been smoothly joined on the lower side. There are up to four variable parameters per peak: the height, the centroid, the full width at half maximum (FWHM), and a parameter (τ) which is the distance from the centroid to the junction point between the Gaussian and exponential functions. The latter two parameters may or may not be variable, depending on the intensity of the peak. If the separation between adjacent peaks is sufficiently small, they will be fit together as (part of) a multiplet. Multiplets with as many as nine component peaks can be handled. The background under the peaks is assumed to be linear.

Of course, no program is likely to recognize the separate existence of peaks which are totally unresolved from each other. For such cases it is necessary to use a different program (such as SKEWGAUS) which permits the user to provide input data for each fit to be performed -- in contrast to PEAKFIND, for which the user supplies input data only for the whole spectrum. SKEWGAUS may also prove useful where options are needed which cannot readily be provided in PEAKFIND. These options include the availability of three additional shape parameters per peak, the ability to constrain parameters of different peaks to maintain a fixed ratio during the fitting procedure, the use of a quadratic background, the opportunity to specify which channels are to be fit and to include gaps in the fitting
region, and other features. The greater flexibility of SKEWGAUS is achieved, however, at some cost in convenience: The user must determine for himself approximately where the peaks are located, and must supply some input data for each peak to be fit; PEAKFIND, on the other hand, requires the user to supply only four simple input data cards for the whole spectrum (which will be largely blank if the user has selected default input options), and the computer does the rest.

The author has found from experience that, for relatively simple spectra where no unresolved multiplets are present, PEAKFIND is usually adequate to perform all the fitting that is necessary. Where there are unresolved multiplets, or where more precise fitting is required, it is still very helpful to perform preliminary analysis with PEAKFIND and to leave only the "problem" fits to SKEWGAUS.

PEAKFIND can handle MCA spectra with as many as 8192 channels. The maximum number of peaks that can be handled at one time is 250. Any number of MCA spectra may be analyzed in one execution of the program. In addition, it is possible to analyze one MCA spectrum several times, using different parameters or examining different parts of the spectrum each time.

The load module for PEAKFIND, when compiled with the H level FORTRAN compiler, is about 156 kilobytes long (39 K words) when no overlays are used. A simple overlay structure may be used to permit execution in a 128 kilobyte region of core storage.

The execution speed of PEAKFIND may be illustrated by the following example: A typical recent execution on the IBM 360/65 system at Iowa State University found 139 peaks in an 8192 channel MCA spectrum; performed 13 fits with an average of more than 50 channels in each fit, involving a
total of 29 peaks with 84 variable parameters, and requiring a total of
97 iterations; all in 24.3 seconds of CPU time (exclusive of the plot step).

II. PEAK FINDING PROCEDURE

The method used in PEAKFIND for locating peaks is a modification
of the method of Mariscotti. (Ref. 1 will henceforth be referred to
as MAM.)

Let us assume temporarily that the MCA spectrum is a continuous
curve, rather than a series of discrete points. The peaks that we wish to
locate rise above a background that we shall assume is approximately linear
in the vicinity of a peak. This background may be eliminated by taking the
second derivative of the curve. The shape of the second derivative in the
vicinity of the peak will depend, of course, on the shape of the peak itself;
but for any reasonable peak shape, we may assume that the second derivative
will be rather strongly negative near the highest point of the peak, becoming
positive on either side of the peak and going to zero a short distance away
from the peak. A fluctuation of this sort in the second derivative could be
taken as the indication of the existence of a peak.

In reality, of course, the spectrum consists of discrete points
rather than a continuous curve. This suggests the examination of the second
difference of the MCA data channels. If there were no statistical fluctuations
in the data, the statements of the previous paragraph would continue to be
valid, with the phrase "second difference" replacing "second derivative."
In fact, however, the statistical fluctuations in the second difference are
quite significant. To illustrate, let us call the number of counts in the
ith channel $N_i$, and let us assume that the statistical uncertainty in this
number is $\sqrt{N_i}$. The second difference is given by

$$S_i = N_{i+1} - 2N_i + N_{i-1} \quad (1)$$

The statistical uncertainty in $S_i$ is given by

$$F_i = \sqrt{N_{i+1} + 4N_i + N_{i-1}} \quad (2)$$

Now suppose that we have a peak whose shape is a perfect Gaussian function with a height of 100 counts and a FWHM of 5 channels, rising above a flat background with an average height of 100 counts per channel. Such a peak would be easily recognizable, standing well above the statistical fluctuations in the background. But the value of $S_i$ at the centroid of the peak (where the magnitude of $S_i$ is greatest) would be -21 counts, while the uncertainty in this value would, from (2), be more than 34 counts. Thus, the peak might easily be overlooked in a search of $S_i$ versus channel number. It follows that some procedure for smoothing the statistical fluctuations is essential if the second difference approach is to be used.

The simplest smoothing technique is to perform an unweighted average of the counts in a group of $w$ adjacent channels centered about a given channel.

If this "smoothing window" $w$ is an odd integer, we may set $w = 2m + 1$ and write

$$S_{i+m} = \sum_{j=i-m}^{i+m} S_j \quad (3)$$

where $S_j$ is given by (1). (A weighted average would involve dividing the right side of (3) by $w$, but it is not necessary to perform this calculation.) Further smoothing is accomplished by repeating the application of (3), so that the relationship between successive smoothings of the data is

$$S_{i+m}^{n+1} = \sum_{j=i-m}^{i+m} S_j^n \quad (4)$$
where the superscript of $S$ refers to the number of times the smoothing operation has been performed. MAM gives an extensive treatment of the determination of optimum values for the smoothing window, $w$, and the number of smoothings, $z$. The conclusion is drawn that 5 smoothings with $w$ equal to the odd integer closest to $0.6 \times$ FWHM provides the best compromise between the objectives of being able to recognize small isolated peaks on a large background, and being able to recognize both members of closely-spaced doublets.

The experience of the author has indicated that the amount of smoothing is highly important to the success of the program when one is dealing with complicated spectra. Too much smoothing can cause both members of closely-spaced doublets to be missed, while too little smoothing can cause fairly prominent single peaks to be missed if the statistical fluctuations are great enough. Because of this, the restriction of $w$ to odd integers has been found to be undesirable. An even value for $w$ does not readily lend itself for use with equation (3) because of the fact that the values of $j$ cannot be symmetrically placed about $i$. However, this problem can be overcome if the number of smoothings is even. One can easily show that the recursion formula for the result of two successive smoothings can be written as

$$s_{i+2}^{n+2} = w s_i^n + \sum_{j=1}^{w-1} (w - j) (s_{i-j}^n + s_{i+j}^n)$$

(5)

This expression is equally valid if $w$ is even or odd. PEAKFIND performs its smoothing by applying (5) twice to the second difference, producing the effect of four smoothings. The default value for $w$ is the integer (odd or even) closest to $0.6 \times$ FWHM. The user may, if he wishes, override the default value for $w$ merely by supplying his own value as part of the input data.
It is necessary to calculate the statistical uncertainty in $S_i$ (we henceforth drop the superscript on $S$ since the number of smoothings is assumed to be 4). Where the number of counts in nearby channels is approximately equal to $N_i$, this statistical uncertainty is easily shown to be

$$F_i = \sqrt{\varphi} \frac{N_i}{c_p}$$

(6)

where $\varphi$ is a constant whose value depends on $w$ and the number of smoothings. MAM has a tabulation of $\varphi$ for odd values of $w$ with $z$ from 1 through 5. For $z=4$ as used in PEAKFIND, the values of $\varphi$ are 6, 28, 82, 184, 350, 596...... for $w = 1, 2, 3, 4, 5, 6......$, respectively.

In the vicinity of a peak, it is to be expected that $S$ will have a form similar to that shown in Fig. 1. Here, $i_1$ and $i_2$ are the first and last channels in which $S_i$ is greater than its statistical uncertainty $F_i$; $i_3$ and $i_5$ are the first and last channels in which $S_i$ is negative; and $i_4$ is the channel in which $S_i$ has its maximum negative value. MAM contains a detailed theoretical investigation of the relationships to be expected among these channels under the assumption of a Gaussian peak shape. This investigation leads to the assignment of four conditions for the identification of a peak, based on (a) the magnitude of $S_i$ at $i_4$, (b) the width of the region between $i_3$ and $i_5$, (c) the separation of channels $i_2$ and $i_3$, and (d) the width of the region between $i_1$ and $i_2$. These rather stringent conditions were established in MAM to prevent the mistaken identification of Compton edges, statistical fluctuations, or other spurious features as peaks.

The author's experience with the conditions of MAM have led him to the conclusion that they are far too restrictive for the purposes for which PEAKFIND is intended, in that they cause too many real peaks to be
Fig. 1. The shape of the smoothed second derivative in the vicinity of a Gaussian peak. The dashed line indicates the statistical uncertainty in the second derivative.
missed (especially in complicated spectra). Condition (d) frequently causes the upper component of a partially resolved doublet to be missed, and the lower component may be missed as well if it is smaller than the upper component, due to condition (b). Condition (d) can also prevent the identification of moderately weak peaks that happen to have large statistical fluctuations.

In an attempt to correct these problems, an entirely different set of conditions for recognizing peaks was adopted in PEAKFIND. These conditions were determined from experience rather than from theoretical considerations. It was found that any test on the region from $i_1$ to $i_3$ in Fig. 1 tends to cause real peaks to be missed, especially in partially resolved multiplets. Therefore, it was decided to test only on the region from $i_3$ to $i_5$. The following conditions were established:

1. The region from $i_3$ to $i_5$ must be at least $w+1$ channels wide and not greater than $3 \times$ FWHM channels wide.

2. The sum of the values of $S_i$ between $i_3$ and $i_5$ inclusive must be great enough to satisfy inequality (7) where $C$ (the peak limit parameter) is a constant whose default value is 2.0.

$$\sum_{i=i_3}^{i_5} S_i > C \times w \times F_{i_3}$$  \hspace{1cm} (7)

The first condition tends to prevent the mistaken identification of either narrow statistical fluctuations or broad features such as Compton edges and backscatter peaks. The second condition effectively determines the smallest peak that can be found by the program. It seems more reasonable, from statistical considerations, to base this condition on the total area of the negative region of $S_i$ rather than on the height, as is done in MAM.

These conditions are sufficiently loose that PEAKFIND will occasionally find spurious features that are not real peaks. This generally presents no
problem since the spurious peaks may easily be rejected before fitting is performed, as described in the following section. However, the user may, if he wishes, reduce the tendency to find such spurious peaks by increasing the value of C in condition 2, but at some cost in the likelihood of missing real peaks that he is interested in.

Fig. 2 shows a portion of a typical gamma-ray spectrum of the sort with which PEAKFIND is commonly used, indicating the peaks that were found when the default options for w and C were chosen. The peak width is about 4.6 channels here, so the default value for w is 3 channels. The default value for C is always 2.0. The vertical scale in this figure is proportional to the square root of the number of counts. On such a plot the statistical error bars are approximately constant in length, and it is these error bars that are plotted in Fig. 2.

It will be noted that peak 7 in the figure is actually a Compton edge, and the existence of peaks 10 and 12 is open to question. All the other peaks, however, are almost certainly real. If the peak limit parameter C is raised to 3.0, the program does not find peaks 7, 10, and 12, but it does find all the other nineteen peaks. On the other hand, it may be observed that the program has failed to recognize that peak 6 is an unresolved doublet, that there is a weak component on the upper side of peak 9, and that there is an additional component below peak 14 and one above peak 15. Nothing can be done in PEAKFIND with peaks 6 and 9, but the additional components next to peaks 14 and 15 are found if the window width w is lowered to 2 while C is kept at its default value of 2.0. This combination of parameters also finds peak 12 to be a doublet, and finds two additional peaks (at channels 1157 and 1216) which are probably not real. However, the Compton edge near channel 970 is not mistaken for a peak when the combination w = 3, C = 2.0 is used.
Fig. 2. Typical gamma-ray spectrum showing peaks identified by PEAKFIND when default options were used. Vertical scale is proportional to the square root of the number of counts.
This example should serve to illustrate the effectiveness of PEAKFIND in locating peaks, and also the ways that the program parameters may be adjusted to suit the user's requirements. This is an example taken from actual practice -- the spectrum of which Fig. 2 is a part was analyzed during a study of the decay properties of 64-second $^{140}$Cs.

After a peak has been found, an estimate is made of its centroid and area from the smoothed second difference. The centroid of $S_i$ in the region between $i_3$ and $i_5$ is used as the estimate of the peak centroid. This estimate usually proves to be quite accurate, rarely being in error by more than a few tenths of a channel for reasonably well resolved peaks. The centroid is converted into a peak energy using a linear calibration of channel number versus energy, supplied by the user.

The peak area is assumed to be proportional to the area of the negative region of $S_i$. The latter area is divided by an approximate value for the efficiency of the detector, and then normalized using a value of 100 for the intensity of the strongest peak found. The detector efficiency is assumed to be represented by a straight line on a full logarithmic plot of efficiency versus energy, except at low energies where the straight line smoothly joins a concave-downward parabola. The efficiency curve is determined by only two parameters that may be supplied by the user -- the slope of the line giving $\log_{10}$ (efficiency) versus $\log_{10}$ (energy), and the energy at which the efficiency curve is a maximum. The default values supplied by the program for these parameters are 1.0 for the slope and 0.1 MeV for the energy of maximum efficiency. These are reasonable values for a moderate-sized coaxial Ge(Li) detector.
The intensity estimates obtained in this fashion are not, of course, expected to be extremely accurate. They are merely intended to aid the user in identifying the individual peaks and to provide a basis for deciding whether a peak is intense enough to be fitted. Experience has shown that the estimates are generally well within a factor of two of the correct values for reasonably well-defined peaks.

In the foregoing section, the FWHM has been treated as if it were constant throughout the whole spectrum. In fact, however, the peak widths usually increase with increasing energy. It is not feasible to permit FWHM to vary in the peak finding procedure. On the other hand, there is no problem in taking into account the variation of FWHM with peak energy when the fitting is performed. This is accomplished by permitting the user to supply, as part of his input data, the intercept and slope of a straight line giving FWHM as a function of energy. While the peak search is being performed, a unique value of FWHM is used by the program -- the value appropriate for a peak precisely at the center of the region being searched. After the peaks are found, each is assigned the FWHM value appropriate for its energy.

III. TO FIT OR NOT TO FIT

Before performing a fit to a peak, it is necessary to establish whether the peak is sufficiently well separated from nearby peaks to allow it to be fitted singly, or whether it must be included with other peaks as part of a multiplet. This decision is based on the FWHM of the peak, as obtained from the straight line supplied by the user giving FWHM as a function of energy. If the separation between two adjacent peaks is less than some constant times FWHM (the default value of this constant being 4.0), they
will be fitted together as (part of) a multiplet. Up to nine peaks may be included in one multiplet. The fitting region will extend below the centroid of the lowest peak a number of channels equal to a constant (default value, 3.0) times the FWHM of the lowest peak, and above the centroid of the highest peak a distance equal to another constant (default value, 2.5) times the FWHM of the highest peak.

It is to be expected that the user will not normally wish to fit all the peaks that have been found. Since the existence of many of the weaker peaks is often questionable, one may wish to specify that only the more intense peaks are to be fitted. Since each peak has been given an intensity estimate, it is possible to specify that only those peaks whose intensity exceeds some cutoff value shall be fitted. The default value for this cutoff is 2.0 (relative to 100 for the most intense peak), but this default value is often overridden in practice. In a multiplet fit, fitting will be performed if any component peak has an intensity exceeding the cutoff value.

A second intensity cutoff value is provided to make it possible to determine those peaks for which the FWHM is to be considered a variable fitting parameter. Since the shapes of the weak peaks are often poorly defined, it is desirable to fix FWHM for these peaks to the straight line values determined from the input supplied by the user. For the stronger peaks whose intensity exceeds the FWHM cutoff, the straight line values of FWHM will be treated only as initial estimates; the "best" values will be determined by the fit. The default value for the FWHM intensity cutoff is 5.0. If the user overrides this default by specifying a value greater than 100, then (since no peak has an intensity greater than 100) all peaks will have fixed FWHM's.
The peak shape parameter $\tau$ is treated in a manner similar to FWHM. The user supplies the slope and intercept of a line giving $\tau$ as a function of energy (or accepts default values, as described in the following section). A value of $\tau$ is assigned to each peak from the straight line. This value is considered to be fixed for any peak whose intensity is less than the $\tau$ cutoff, and is treated as an initial fitting estimate for a more intense peak. The default value for the $\tau$ cutoff is 10. By overriding with a value greater than 100, the user may fix the $\tau$'s of all peaks.

Before any fitting is attempted, the program provides printed output giving a list of the peaks found with their estimated energies and intensities, as illustrated in the upper part of Fig. 3. Starred peaks are those whose intensities exceed the fit cutoff value. Next, output is provided to indicate how these peaks are grouped into multiplet fits, as illustrated in the lower part of Fig. 3. This list includes the number of peaks in each multiplet, the energy of the first peak of the multiplet, the first and last channels in the fitting region, the initial estimates of FWHM and $\tau$ for the first peak, and the centroids of all peaks in the multiplet. The output illustrated in Fig. 3 was obtained from an analysis of the spectrum of Fig. 2, using default values for all parameters except the energy calibration and FWHM.

At this point, the user has three options on how to proceed. The options are determined by the value of the index IFIT.

IFIT = 0: No fitting will be attempted. The user wishes to examine the output already provided before proceeding further.

IFIT = 1: Fitting will be attempted for all multiplets in which at least one component peak has an intensity greater than the fit cutoff value.
Fig. 3. Preliminary output provided by PEAKFIND for the spectrum of Fig. 2, indicating the peaks found and the grouping of these peaks into multiplets.
IFIT = 2: Fitting will be attempted for those specific multiplets designated by the user. Multiplets are identified by the fit number as given in the first column of the lower part of Fig. 3.

The last option is useful primarily for fitting selected weaker peaks in a spectrum whose stronger peaks have already been fitted on previous executions of PEAKFIND with IFIT = 1. Obviously, the user cannot know the numbers of the multiplets he wishes to fit unless he has previously obtained a listing similar to that shown in Fig. 3. Furthermore, it is necessary that most of the input parameters remain unchanged between executions in order to guarantee that exactly the same peaks will be found and that they will be grouped into multiplets in exactly the same way. However, those parameters that do not affect the finding or grouping of peaks (notably the cutoff and T parameters) may safely be changed without affecting the multiplet identification.

IV. PEAK FITTING PROCEDURE

The peak fitting procedure is very similar to that used in program SKEWGAUS, and the reader is directed to Ref. 2 for complete details. Only a brief summary is given here.

The fitting function is a Gaussian to which a simple exponential function has been smoothly joined on its lower side. The four variable parameters are the peak height $h$, the centroid $x_0$, the FWHM of the Gaussian, and the distance $\tau$ (in channels) from the centroid to the junction point. These parameters are illustrated in Fig. 4. The height and centroid are always treated as variable fitting parameters, but FWHM and $\tau$ are variable only for those peaks whose intensity estimates exceed the corresponding cutoff values, as explained in the previous section.
Fig. 4. Function used for peak fitting.
The fitting function used in SKEWGAUS includes three additional variable parameters which permit a wider variety of peak shapes to be fit. These additional parameters are the tail parameter \( t \) in Fig. 4, which permits the fitting function to approach a constant value other than zero on its lower side; the lower skewness parameter, which permits some additional skewing of the lower side of the peak; and the upper skewness parameter, which provides for skewing on the upper side of the peak. These three are not available as variable parameters in PEAKFIND; however, provision has been made for the user to supply fixed values for these parameters for each peak. This is accomplished, as with FWHM and \( T \), by supplying the intercept and slope of a straight line giving the value of each parameter as a function of peak energy.

The background in PEAKFIND fits is a straight line given by the expression

\[
f_B = A + B(x - x_1)
\]  

(8)

where \( x \) is channel number and \( x_1 \) is the number of the first channel in the fitting region. The use of a more complicated function allowing for curvature in the background has been found from experience to be undesirable, causing more problems than it solves. If a curved background is clearly called for, SKEWGAUS should be used.

Fitting is accomplished by the standard nonlinear least-squares (NLLS) method. The quantity

\[
Q = \sum_{i} w_i (N_i - f_i)^2
\]  

(9)

is minimized by an iterative process. Here, \( N_i \) is the number of counts in the \( i \)th channel, \( f_i \) is the fitting function evaluated at the \( i \)th channel, and \( w_i \) is a weight associated with the error in \( N_i \). The weight \( w_i \) is given by
the following expression:

\[ w_i = \frac{1}{N_i (1 - c + d^2 N_i) + c N_m} \]  

(10)

Here, \( d \) is the r.m.s. deviation in channel width of the analogue-to-digital convertor of the MCA (i.e., the differential nonlinearity). \( N_m \) is the maximum number of counts among the channels included in the fit. The constant \( c \) is included for the purpose of reducing somewhat the very large weight that would otherwise be given to channels with a small number of counts, where \( N_i \) is very small compared with \( N_m \). The values of \( c \) and \( d \) built into the program are 0.02 and 0.005, respectively, but they may easily be changed if desired. Setting \( c \) and \( d \) both equal to zero will give standard statistical weighting.

\( Q \) is minimized by solving the set of nonlinear equations

\[ \frac{\partial Q}{\partial \alpha_j} = 0 \]  

(11)

where \( \alpha_j \) represents the \( j \)th free parameter in the fitting function. Since equations (11) are not solvable analytically, an iterative procedure is used to get an approximate solution. Initial estimates are supplied for the values of the free parameters, and equations (11) are expanded in powers of small quantities. The nonlinear terms in the expansion are dropped, and the resulting linear equations are solved by the method of matrix inversion. The solution is a set of new (and hopefully better) values for the \( \alpha_j \), which are then used as estimates in the next iteration.

The iterative procedure continues until one of the following conditions occurs:

(1) All parameters no longer change significantly on successive iterations (normal convergence). A "significant" change is defined to be a fractional change of \( 10^{-3} \) in \( h \), \( 10^{-5} \) in \( x_o \), \( 10^{-3} \) in FWHM, \( 5 \times 10^{-3} \) in \( \tau \), \( 10^{-4} \) in \( A \), and \( 10^{-4} \) in the quantity \((B + 1000)\).
(2) Q increases on successive iterations, or fails to decrease fractionally by more than $10^{-4}$. The test for this condition is not applied until the twentieth iteration since Q frequently increases on early iterations but subsequently decreases.

(3) The coefficient matrix is singular, so the equations resulting from the linearization of (11) cannot be solved. This usually means that some variable parameter is totally undefined.

(4) Thirty iterations have taken place and none of the above conditions has occurred.

After convergence has been achieved, or the iterative procedure has been stopped without convergence because of conditions (3) or (4), the final value of Q is divided by the number of degrees of freedom (the number of channels in the fit minus the number of variable parameters) to determine the variance of the fit. If only statistical fluctuations affected the quality of the fit, one could distinguish a good fit from a poor one by the requirement that the variance be close to 1.0 for a good fit. In fact, however, there are many non-statistical effects that must be considered, the most important being the fact that the fitting function is an arbitrarily chosen representation of the peak shape. Where the number of counts in a peak is large (say, on the order of $10^6$), the statistical uncertainties in the data points are small enough to expose the small differences between the fitting function and the actual line shape, with the result that the variance may be significantly greater than 1.0 for a fit that is completely acceptable. On the other hand, a fit to a small multiplet rising above a large background may yield a variance close to 1.0 even if some major error has occurred, such
as the failure to find all of the components of the multiplet. It follows, then, that the variance alone cannot be relied upon blindly in determining whether a fit is acceptable. Examination of a plot of the fit is strongly recommended.

The uncertainty in the final value of each variable parameter is determined by taking the square root of the product of the appropriate diagonal element of the inverted coefficient matrix times the variance. These uncertainties must be interpreted with some care since they are based on statistical considerations only and do not take into account such non-statistical factors as the imperfect representation of the actual peak shape by the fitting function. Further discussion of the parameter uncertainties (as well as all the other topics covered in this section) may be found in Ref. 2.

The NLLS method of fitting requires that initial estimates be provided for all variable parameters. The background parameters A and B are initialized as, respectively, the number of counts in the first channel of the fitting region and the slope of a line drawn between the first and last channels of the region. The centroid estimates are provided when the peaks are found, as discussed above. The height of a peak is estimated to be the contents of the centroid channel minus the background contribution to that channel. FWHM and τ, as has been mentioned, are given estimates based on the assumption that these parameters vary linearly with energy. In practice, one simply estimates visually the FWHM for a few of the stronger peaks in the spectrum and fits these values to a straight line. The slope and intercept of this line must be supplied as part of the input data. It is not so easy to estimate τ visually, so the user may let the program
itself determine the initial estimates for $\tau$. Since $\tau$ usually varies quite slowly with energy, the default initial estimate is the same for peaks of all energy, being equal to one-half the FWHM for a 1 MeV peak. After the stronger peaks in a spectrum have been analyzed, the user may wish to use the fitted values of FWHM and $\tau$ for these peaks to determine the best straight lines giving FWHM and $\tau$ as functions of energy. These straight lines may be used in fitting the weaker peaks. Experience has shown that the assumption of a linear variation of FWHM and $\tau$ with energy is generally quite satisfactory.

V. PROGRAM OUTPUT

A. Printed output

The preliminary output listing provided after the peaks have been found was discussed above and illustrated in Fig. 3.

A listing of the values of all variable parameters after each iteration of each fit may optionally be provided. A typical listing for one fit is shown in Fig. 5. This fit consists of peaks 21 and 22 of Fig. 2. The parameter values listed under iteration 0 are the initial estimates. The parameters are numbered as follows:

1, height of first peak;
2, centroid of first peak;
3, FWHM of first peak;
4, $\tau$ of first peak;
5, height of second peak;
6, centroid of second peak.

In this example, the intensity estimate for the second peak was sufficiently small that its FWHM and $\tau$ (parameters 7 and 8) were not variable and hence
<table>
<thead>
<tr>
<th>PARAMETERS FOR PEAK (1)</th>
<th>ENERGY</th>
<th>50.445 keV</th>
<th>FWHM</th>
<th>6.668</th>
<th>TAU</th>
<th>2.660</th>
<th>TAIL</th>
<th>0.0</th>
<th>SKLO</th>
<th>0.0</th>
<th>SKUNL</th>
<th>0.0</th>
</tr>
</thead>
</table>

**VALUES OF FREE PARAMETERS AFTER EACH ITERATION**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.427032E+00</td>
<td>1.320500E+00</td>
<td>1.320500E+00</td>
</tr>
<tr>
<td>2</td>
<td>1.241500E+00</td>
<td>1.241500E+00</td>
<td>1.241500E+00</td>
</tr>
<tr>
<td>3</td>
<td>2.440000E+00</td>
<td>2.440000E+00</td>
<td>2.440000E+00</td>
</tr>
<tr>
<td>4</td>
<td>2.440000E+00</td>
<td>2.440000E+00</td>
<td>2.440000E+00</td>
</tr>
<tr>
<td>5</td>
<td>3.001500E+00</td>
<td>3.001500E+00</td>
<td>3.001500E+00</td>
</tr>
<tr>
<td>6</td>
<td>3.001500E+00</td>
<td>3.001500E+00</td>
<td>3.001500E+00</td>
</tr>
<tr>
<td>7</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
</tr>
<tr>
<td>8</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
</tr>
<tr>
<td>9</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
</tr>
<tr>
<td>10</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
<td>3.121000E+00</td>
</tr>
<tr>
<td>11</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>12</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>13</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>14</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>15</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>16</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>17</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>18</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>19</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>20</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>21</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>22</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>23</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>24</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>25</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>26</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>27</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>28</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>29</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
<tr>
<td>30</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
<td>1.030150E+00</td>
</tr>
</tbody>
</table>

**COMPARISON OF ENERGY REGIONS**

| VARIANCE | = 7.86 | var = 8.6913E-03 | VARIANCE/SUM OF WEIGHTS = 6.783E-05 |

Fig. 5. Typical output for one fit, which includes peaks 21 and 22 of Figs. 2 and 3.
are not listed. If there were a third peak, its parameters would be numbered 9 through 12, and so on. The next-to-last parameter (number 9 in this example) is always the background height A. The last free parameter is the background slope B plus 1000.

In the example of Fig. 5, convergence is achieved very quickly because the initial estimates are reasonably close to the correct values and the peaks are very well defined. In other cases convergence is not always so rapid.

After all fits have been performed on one MCA spectrum, a summary output of the essential results of the fits is provided. A typical summary output is shown in Fig. 6. The first few columns are the fit number, the number of peaks in the fit, the number of free parameters, the number of channels in the fit, the variance of the fit, and the reason for convergence or non-convergence. The final values of all parameters for the individual peaks follow. An error of 0.0 indicates that the parameter was not variable. The areas are obtained by numerically integrating the fitting function over all the channels included in the fit. The error in the area is a combination of the error in the height and the error in FWHM. Further details concerning areas may be found in Ref. 2.

This summary output is provided in the same job step as the rest of the program, rather than in a separate job step or a separate job, as is the case for the SKEWGAUS summary output.

8. Punched-card output

Two types of punched-card output may be provided by PEAKFIND. The first type is suitable for input to program SKEWGAUS, and is intended for situations where the fitting done by PEAKFIND is not adequate. One card is
Fig. 6. Summary output for fits performed as indicated in Fig. 3.
provided for every fit in the "Summary of Parameters for Fits" (Fig. 3), appropriate for use as data control card D of SKEWGAUS. These cards have the fit number punched in columns 78-80 for identification purposes. In addition, one card is punched for every peak found, in a manner appropriate for use as data control card E of SKEWGAUS.

The second type of punched-card output is intended for use where the results of the PEAKFIND fits are satisfactory, and are to be used to determine accurate energy and intensity values, including corrections for system nonlinearities, detector efficiency variation, and the like. This processing is done by a program called DRUDGE. PEAKFIND will optionally provide punched cards that may be used as input for DRUDGE. There will be one card for each peak that has been fitted, containing the following information: the centroid, the error in the centroid, the area, the height, the error in the height, FWHM, the error in FWHM, \( \tau \), and the error in \( \tau \). The first five quantities are punched in 10-column fields, the last four quantities in 5-column fields.

Note that these cards for DRUDGE are punched only for those peaks that were actually fitted, while the cards for SKEWGAUS are provided for all peaks found, whether or not any fitting was performed.

C. Graphical output

A graph of the results of a fit is an indispensable aid in determining the quality of the fit. Typical plots are shown in Figs. 7 and 8. It may be observed that the fit in Fig. 8 appears to be quite good, in spite of the fact that the variance of this fit is 7.85, which would normally be taken to indicate a very poor fit. This is a common situation where the height of the peak is in excess of \( 10^5 \) counts, as is the case here.
Fig. 7. Graphical output for fit 3 of Fig. 6 (peak 5 of Figs. 2 and 3).
Fig. 8. Graphical output for fit 9 of Fig. 6 (peaks 21 and 22 of Figs. 2 and 3). Output from this fit is also shown in Fig. 5.
The plots are drawn with a constant number of channels per inch (default value, 10). The channels plotted include five additional channels on either side of the fitting region that were not included in the fit. The fitting function is drawn as a smooth curve through only those channels that were actually included in the fit. The vertical scale is the square root of the number of counts. The plotting symbol is normally a vertical line whose length indicates the statistical uncertainty in the number of counts -- an error bar, in effect. If the number of counts is sufficiently large, the size of the error bar becomes too short for convenience. In this event, a triangle (whose size has no significance) is used as the plotting symbol.

The graphs are plotted by a Cal-Comp plotter. Most of the work of plotting is done by a set of routines designed locally by D. G. Scranton and known as SIMPLOTTER. It must be admitted that not all PEAKFIND fits are as successful as those graphed in Figs. 7 and 8. Because of the unresolved doublets in the spectrum of Fig. 2, it was necessary, when these data were analyzed during the study of the $^{140}$Cs decay scheme, to resort to SKEWGAUS in order to get acceptable fits for a number of the multiplets.

VI. PROGRAM INPUT

A. MCA data

The MCA data may be in several forms -- on cards, magnetic tape, or magnetic disk -- that are appropriate for the MCA's with which this program is generally used. One subprogram, named INPUT, performs the sole function of reading these data. This subprogram is nearly identical with the subprogram of the same name in SKEWGAUS, which is described in detail in Ref. 2. The
only essential difference is that in PEAKFIND an additional option is
provided which permits the reading of the MCA data to be bypassed, so that
a given spectrum may be analyzed more than once.

It is assumed that the initial channel of the MCA spectrum being
read is channel 0, not channel 1 (except for data on cards, for which the
user may specify the initial channel number).

B. Data control cards

The input data deck consists basically of four cards for each MCA
spectrum. These cards are designated by the letters A through D. If the
option of fitting specified multiplets (IFIT=2) has been chosen, there will
be one or more additional cards, of type E.

Card A specifies the form of the MCA data. The format of this card
is given in Table I. It is identical to card A of SKEWGAUS, except that
an additional option has been provided for variable IREAD.

Card B is a title card on which may appear any 80 characters. These
will be printed as the heading of the printed output. In addition, the contents
of columns 1-20 of this card will be the first line of the title on each graph.

Card C contains the options for finding and fitting peaks. The format
for this card is given in Table II.

Card D contains the energy-dependent quantities. The format for
this card is given in Table III.

Card(s) E appears only if IFIT is equal to 2 on card C. A letter
T (for "true") will be punched in each column for which the correspondingly
numbered multiplet is to be fitted. The numbering of the multiplet fits
is given in the first column of the "Summary of Parameters for Fits" (as
illustrated in the bottom of Fig. 3). If the number of multiplets is 80
or less, there will be only one card E. If the number of multiplet fits is between 81 and 160, there will be two cards E, and so on. The user would specify the fitting of multiplet 81 by punching a T in column 1 of the second card E.

After the program has finished with an MCA spectrum it returns to the beginning and reads card A for the next MCA spectrum. Any number of spectra can be handled in a single execution of PEAKFIND. A blank card at the end of the input deck will terminate execution properly. (This blank card is interpreted by the program as card A with IREAD=0.)

The data control cards used to analyze the spectrum of Fig. 2 and to produce the output illustrated in Figs. 3, 5, 6, 7 and 8 are shown in Fig. 9. Default values were accepted for most parameters, as may usually be done in practice.
Fig. 9. Typical data deck, showing the control cards which produced the output shown in Figs. 3, 5, 6, 7, and 8.
Table I. Format of data control card A. In this and the following tables, variable names in parentheses have standard default values and hence do not normally need to be supplied by the user.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Name</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>I1</td>
<td>IREAD</td>
<td>Designates source and format of MCA data.</td>
</tr>
<tr>
<td>3-10</td>
<td>A8</td>
<td>IDEN</td>
<td>Identification word for data on disk. Not required for data on tape or cards.</td>
</tr>
<tr>
<td>11-15</td>
<td>4I5</td>
<td>(LOCATN)</td>
<td>Data on cards: LOCATN (1) and LOCATN (2) are the first and last channel numbers of the MCA channels to be read in. These data cards should follow immediately after card A.</td>
</tr>
<tr>
<td>16-20</td>
<td></td>
<td>array</td>
<td>Data on disk: Not required unless data set is to be read by position rather than by identification word. If LOCATN (1) ≠ 0, array gives location(s) on disk of the data set to be read. Must also have NTHOUS≠0 in this case.</td>
</tr>
<tr>
<td>21-25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26-30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31-35</td>
<td>I5</td>
<td>(NTHOUS)</td>
<td>Required only if data set on disk is to be located by position rather than by identification word. Gives length of data set in K (units of 1024 words).</td>
</tr>
<tr>
<td>36-80</td>
<td>--</td>
<td>--</td>
<td>Not referenced by program. May be used to identify data control deck if desired.</td>
</tr>
</tbody>
</table>
Table II. Format of data control card C.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Name</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>I2</td>
<td>IFIT</td>
<td>0 - No fitting will be performed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Multiplets having at least one peak with intensity greater than FITCTF will be fitted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Multiplets designated on card(s) E will be fitted.</td>
</tr>
<tr>
<td>3-4</td>
<td>I2</td>
<td>IPRINT</td>
<td>0 - Preliminary and summary printed output only are provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Complete printed output is provided, including listing similar to Fig. 5 for each multiplet.</td>
</tr>
<tr>
<td>5-6</td>
<td>I2</td>
<td>IPUNCH</td>
<td>0 - No punched card output.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Punched cards for input to SKEWGAUS are provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - Punched cards for input to DRUDGE are provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - Punched cards for both SKEWGAUS and DRUDGE are provided.</td>
</tr>
<tr>
<td>7-8</td>
<td>I2</td>
<td>IGRAPH</td>
<td>0 - No graphical output.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - One graph is plotted for each fit.</td>
</tr>
<tr>
<td>9-10</td>
<td>I2</td>
<td>(NWINDO)</td>
<td>Smoothing window ( w ) in equation (5). Default, ( 0.6 \times \text{FWHM} ) for a peak midway between channels NFIRST and NLAST.</td>
</tr>
<tr>
<td>11-15</td>
<td>I5</td>
<td>(NFIRST)</td>
<td>First and last channel numbers of region to be searched for peaks. Default for NFIRST is 0. Default for NLAST is last channel of MCA spectrum.</td>
</tr>
<tr>
<td>16-20</td>
<td>I5</td>
<td>(NLAST)</td>
<td></td>
</tr>
<tr>
<td>21-25</td>
<td>F5.0</td>
<td>(FITCTF)</td>
<td>If IFIT=1, fitting will be attempted for peaks with intensity greater than FITCTF. Default, 2.0.</td>
</tr>
<tr>
<td>26-30</td>
<td>F5.0</td>
<td>(FWCTF)</td>
<td>When fitting, FWHM will be treated as a variable parameter for peaks with intensity greater than FWCTF. Default, 5.0.</td>
</tr>
<tr>
<td>31-35</td>
<td>F5.0</td>
<td>(TAUCTF)</td>
<td>When fitting, ( \tau ) will be treated as a variable parameter for peaks with intensity greater than TAUCTF. Default, 10.0.</td>
</tr>
<tr>
<td>36-40</td>
<td>F5.0</td>
<td>(PKLMT)</td>
<td>Peak limit parameter ( C ) in equation (7). Establishes smallest fluctuation recognizable as a peak. Default, 2.0.</td>
</tr>
<tr>
<td>Column</td>
<td>Format</td>
<td>Name</td>
<td>Interpretation</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>41-45</td>
<td>F5.0</td>
<td>(FITBEL)</td>
<td>Fitting region extends from FITBEL × FWHM channels below centroid of first peak to FITABV × FWHM channels above centroid of last peak in multiplet. Defaults, 3.0 and 2.5, respectively.</td>
</tr>
<tr>
<td>46-50</td>
<td>F5.0</td>
<td>(FITABV)</td>
<td></td>
</tr>
<tr>
<td>51-55</td>
<td>F5.0</td>
<td>(SEP)</td>
<td>Peaks will be fitted together as (part of) a multiplet if the separation between their centroids is less than SEP × FWHM. Default, 4.0.</td>
</tr>
<tr>
<td>56-60</td>
<td>F5.0</td>
<td>(XSF)</td>
<td>Number of channels per inch on plots. Default, 10.0. Note F format!</td>
</tr>
</tbody>
</table>
Table III. Format of data control card D.

<table>
<thead>
<tr>
<th>Column</th>
<th>Format</th>
<th>Name</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>F10.0</td>
<td>E0</td>
<td>Energy calibration is given by $E(\text{keV}) = E0 (\text{keV}) + ECH (\text{keV/channel}) \times (\text{channel number})$</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.0</td>
<td>ECH</td>
<td></td>
</tr>
<tr>
<td>21-25</td>
<td>F5.3</td>
<td>FWHMO</td>
<td>FWHM is given by $\text{FWHM (channels)} = \text{FWHMO (channels)} + \text{FWMEV (channels/MeV)} \times E (\text{MeV})$</td>
</tr>
<tr>
<td>26-30</td>
<td>F5.3</td>
<td>(FWMEV)</td>
<td></td>
</tr>
<tr>
<td>31-35</td>
<td>F5.3</td>
<td>(TAU0)</td>
<td>$\tau$ is given by $\tau (\text{channels}) = \text{TAU0 (channels)} + \text{TAUMEV (channels/MeV)} \times E (\text{MeV})$</td>
</tr>
<tr>
<td>36-40</td>
<td>F5.3</td>
<td>(TAUMEV)</td>
<td>Defaults: \text{TAU0}, 0.5 $\times$ (FWHMO + FWMEV); \text{TAUMEV}, 0.</td>
</tr>
<tr>
<td>41-45</td>
<td>6F5.3</td>
<td>(TAIL1)</td>
<td>Intercept and slope of straight lines giving tail, lower skewness, and upper skewness as functions of energy. These parameters are not variable parameters in PEAKFIND as they are in SKEWGAUS, but they may be assigned fixed values.</td>
</tr>
<tr>
<td>46-50</td>
<td></td>
<td>(TAILS)</td>
<td></td>
</tr>
<tr>
<td>51-55</td>
<td></td>
<td>(SKULO1)</td>
<td></td>
</tr>
<tr>
<td>56-60</td>
<td></td>
<td>(SKULOS)</td>
<td></td>
</tr>
<tr>
<td>61-65</td>
<td></td>
<td>(SKUHI1)</td>
<td></td>
</tr>
<tr>
<td>66-70</td>
<td></td>
<td>(SKUHIS)</td>
<td></td>
</tr>
<tr>
<td>71-75</td>
<td>F5.3</td>
<td>(EFFS)</td>
<td>Slope of straight line giving $\log_{10}$ (detector efficiency) versus $\log_{10}$ (peak energy). Default, 1.0.</td>
</tr>
<tr>
<td>76-80</td>
<td>F5.3</td>
<td>(EFMAX)</td>
<td>Energy (in MeV) of maximum efficiency of detector. Default, 0.1.</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

PEAKFIND began as a revision of program PALMUD II, written by Taff and Champion\textsuperscript{5} as a revision of the original program of Mariscotti.\textsuperscript{6} It gradually became clear that some rather drastic changes were desirable, and so the program was rewritten almost from scratch, with only a few vestiges of the original programs remaining. Many aspects of SKEWGAUS were incorporated into PEAKFIND, and various additional features have been added in the five years that the program has been in use. Many of the past and present members of the TRISTAN group have contributed helpful suggestions during this time. The peak fitting function, which is also used in SKEWGAUS, was devised by Fred Wohn. The matrix inversion subroutine DUMNV was supplied by the I.S.U. Computation Center staff.
REFERENCES

APPENDIX A. PROGRAM DESCRIPTION

PEAKFIND consists of a MAIN program; eight subroutines named INPUT, PKFIND, OUT1, PKFIT, NLLS, FUNCT, DUMNV, and OUT2; and seven labeled COMMON areas named COM1, COM2, COM3, COM4, COM6, COM7, and COM8. Additional subprograms required are all in the standard FORTRAN library, except for the subroutines GRAPH and ORIGIN, which are part of the SIMPLOTTER package and are used only for producing the graphical output.

MAIN establishes the standard values for various parameters; reads cards B, C, and D; assigns default values; and calls the various subprograms.

INPUT reads card A and the MCA data. MAIN calls INPUT once for each spectrum to be analyzed.

PKFIND finds the peaks and estimates their centroids, areas, and intensities. MAIN calls PKFIND once for each spectrum to be analyzed.

OUT1 writes the preliminary output. It groups the peaks into multiplets, determines the channels to be included in each multiplet fit, and finds the initial estimates of FWHM and $\tau$ for each peak. MAIN calls OUT1 once for each spectrum to be analyzed.

PKFIT decides whether to perform each fit. It initializes the various arrays that are needed for fitting. After a fit has been performed, PKFIT saves the results for the summary output. MAIN calls PKFIT once for each spectrum to be analyzed.

NLLS does the nonlinear least-squares fitting. It also prints the intermediate output, if required (i.e., the values of the variable parameters after each iteration). This subroutine is very similar to a subroutine of the same name in SKEWGAUS. PKFIT calls NLLS once for each fit to be performed.
FUNCT evaluates the fitting function and its partial derivatives with respect to the variable parameters. NLLS calls FUNCT once for each channel in each iteration of each fit. This subprogram is a somewhat simplified version of the subroutine of the same name in SKEWGAUS, because of the smaller number of variable parameters (four per peak instead of seven). FUNCT is also called by OUT2 when the summary output is provided.

DUMNV is a standard double-precision matrix-inversion routine. NLLS calls DUMNV once for each iteration of each fit.

OUT2 provides the summary output and the graphs. MAIN calls OUT2 once for each spectrum to be analyzed.

At the end of analysis of a given spectrum the program cycles back to the beginning to read the first data card for the next spectrum. Any number of spectra may be analyzed in a single execution of the program.

An overall flow chart for PEAKFIND is given in Fig. 10. A flow chart for subroutine PKFIND is given in Fig. 11.
Fig. 10. Overall flow chart for PEAKFIND. The subprograms in which the various operations are performed are indicated in brackets on the right.
Fig. 11. Flow chart for subroutine PKFIND.
APPENDIX B. PROGRAM REQUIREMENTS AND LIMITATIONS

The maximum number of channels that may be included in an MCA spectrum is 8192. This number may be changed simply by changing the dimension of array DATA in common area COM1, and of array A in subroutine PKFIND. The value assigned to variable LENGTH at the beginning of MAIN should also be changed accordingly.

The maximum number of peaks that can be handled is 250. This number may be changed by changing the dimension of all arrays dimensioned to 250 in common areas COM2 and COM8. The value assigned to variable MAXPKS at the beginning of MAIN should be changed accordingly.

The maximum number of channels in a given fit is 200. This number may be changed by changing the dimension of all arrays dimensioned to 200 in COM6, and the dimension of array YCAL in OUT2. In addition, the dimension of array XC in OUT2 should be one greater than the maximum number of channels, and the dimension of arrays X and Y should be 10 greater (to allow for the plotting of five additional channels on each side of the fitting region). The value assigned to variable MXFTSZ at the beginning of MAIN should be changed accordingly.

The maximum number of peaks in a multiplet fit is 9. To change this number one must change the dimension of all arrays dimensioned to 9 in common area COM7 (and of array TERM, which must be one greater) and also of all arrays dimensioned to 38. This number is the maximum number of variable parameters, equal to four times the maximum number of peaks plus two. The dimension of arrays in NLLS dimensioned to 38 must similarly be changed. Array A (the coefficient matrix) must have a first dimension equal to twice the number of variable parameters, and array AA (equivalenced to A) must
be dimensioned to the square of the number of variable parameters. The third argument in the CALL DUMNV statement (card NLLS0770) must also be equal to the maximum number of variable parameters. The value assigned to MLTPLT at the beginning of MAIN is the maximum number of peaks per fit.

The maximum number of iterations permitted per fit is 30. This number may be changed by changing the value assigned to MAXIT at the beginning of MAIN. The assignment of most other standard and default values is also done in MAIN so that they may easily be changed if desired.

As we observed in the Introduction, the load module for PEAKFIND is about 156 kilobytes long when no overlays are used. An overlay tree that will permit execution in a 128 kilobyte region of core is generated by the following set of instructions to the linkage editor:

```
INSERT MAIN
    OVERLAY ONE
INSERT INPUT
    OVERLAY ONE
INSERT PKFIND
INSERT OUT1
    OVERLAY ONE
INSERT FUNCT
INSERT COM6
INSERT COM7
INSERT COM8
    OVERLAY TWO
INSERT PKFIT
INSERT NLLS
INSERT DUMNV
    OVERLAY TWO
INSERT OUT2
```
C PROGRAM 'PEAKFIND'
C
C MAIN PROGRAM

CCMMCN /COM1/ LENGTH,NFIRST,NLAST,NWIND,DATA(8192)
CCMMCN /COM2/ MAXPKS,NPEAKS,POS(250),ENERGY(250),AREA(250),
1 FWHM(250),TAU(250),NPKS(250),FITLO(250),FITHI(250),NFITS
CCMMCN /COM3/ EO,ECH,FWHMO,FWMEV,TAUO,TAUMEV,SKUHII,SKUHIS,
1 TAILI,TAILS,NSKULO,NSKUHI,NL01,NHI1,MAXIT,ITQ,
1 EPSQ,EFFS,EFMXLG,EFTERM,PKLMT,NDR,NPRTR,NPNCN
CCMMCN /COM4/ IFIT,IPRINT,IPUNCH,IGRAPH,IFIX,MLTPLT,MXFTSZ,EPSHT,
1 EPSX,EPSFW,EPSTAU,EPSA,EPSE,B,DIFLIN,CONST,SEP,FITABV,FITBEL,
1 FITCTF,FWCTF,TAUCTF,XSF

C DIMENSION LABEL(20)
C
C NRDR=5
C NPRTR=6
C NPNCH=7
C LENGTH=8192
C MAXPKS=250
C MXFTSZ=200
C MLTPLT=9
C MAXIT=30
C
C 10 CALL INPUT(DATA,LENGTH,12)

C
READ CARDS B, C, AND D
READ (NRDR,100) LABEL
100 FORMAT (20A4)
READ (NRDR,102) IFIT,IPRINT,IPUNCH,IGRAPH,NWINDO,NFIRST,NLAST,
1 FITCTF,FWCTF,TAUCTF,PKLMT,FITBEL,FITABV,SEP,XSF
102 FORMAT (SI2,215,10F5.0)
READ (NRDR,104) EO,ECH,FWHMC,FWMEV,TAU0,TAUMEV,TAILI,TAILS,
1 SKULOI,SKULOS,SKUHII,SKUHIS,EFFS,EFMAX
104 FORMAT (2F10.0,12F5.3)
ASSIGN DEFAULT VALUES
IF (NLAST .EQ. 0 .OR. NLAST .GE. 12) NLAST=I2-1
IF (FITCTF .EQ. 0.) FITCTF=2.0
MINIMUM INTENSITY FOR PEAK TO BE FIT
IF (FWCTF .EQ. 0.) FWCTF=5.0
MINIMUM INTENSITY FOR FWHM TO BE VARIABLE
IF (TAUCTF .EQ. 0.) TAUCTF=10.0
MINIMUM INTENSITY FOR TAU TO BE VARIABLE
IF (XSF .EQ. 0.) XSF=10.0
NUMBER OF POINTS PER INCH ON PLOTS
IF (FITBEL .EQ. 0.) FITBEL=3.0
IF (FITABV .EQ. 0.) FITABV=2.5
FIT REGION EXTENDS FROM (FITBEL*FWHM) CHANNELS BELOW FIRST PEAK TO (FITABV*FWHM) CHANNELS ABOVE LAST PEAK
IF (SEP .EQ. 0.) SEP=4.0
MAXIMUM SEPARATION BETWEEN PEAKS IN A MULTIPLE
FIT IS (SEP*FWHM)
IF (FKLMT .EQ. 0.) PKLMT=2.0
PKLMT DETERMINES SMALLEST FLUCTUATION IN 2ND DERIVATIVE THAT WILL BE RECOGNIZED AS A PEAK
IF (ECH .EQ. 0.) ECH=0.5
IF (FWHMO .EQ. 0.) FWHMO=6.0
IF (TAUO .EQ. 0.) TAUO=(FWHMO+FWMEV)/2.0
IF (EFFS .EQ. 0.) EFFS=-1.0
EFFS=-ABS(EFFS)

C SLOPE OF EFFICIENCY CURVE ON FULL LOG PLOT

IF (EFMAX .LE. 0.) EFMAX=0.1

C ENERGY WHERE EFFICIENCY IS A MAXIMUM
EFMAXLG=ALOG10(EFMAX)
EFTERM=EFMAXLG+0.1505

C THE FOLLOWING QUANTITIES ARE CONVERGENCE CRITERIA FOR
C VARIABLE PARAMETERS

EPSHT=.001
EPSX0=.0001
EPSF=.001
EPSTAU=.005
EPSA=.0001
EPSB=.0001
EPSQ=.0001

C FOLLOWING PARAMETERS ARE USED FOR MISCELLANEOUS PURPOSES

ITQ=20

C ITERATION ON WHICH Q TEST IS FIRST APPLIED

DIFLIN=.005

C DIFFERENTIAL NONLINEARITY OF MCA

CCNST=.02

C CONSTANT USED TO REDUCE STATISTICAL WEIGHT GIVEN TO
C CHANNELS WITH A SMALL NUMBER OF COUNTS

\$SKULO=2

C EXPONENT IN LOWER SKEWNESS TERM

NL;=\$SKULO-1

C EXPONENT IN UPPER SKEWNESS TERM

NHI:\$SKUHI-1
C
C FIND PEAKS
C
CALL FK FIND
C
WRITE RESULTS OF PK FIND
C
CALL OUT1(LABEL)
IF (IFIT .EQ. 0) GO TO 10
C
FIT PEAKS
C
CALL FK FIT
C
WRITE RESULTS OF PEAK FITTING
C
CALL OUT2(LABEL)
GO TO 1C
END
SUBROUTINE INPUT(DATA, LENGTH, I2)
C
THIS IS A GENERAL SUBROUTINE FOR READING AN MCA SPECTRUM INTO
ARRAY 'DATA'. LENGTH IS THE SIZE OF THE ARRAY. I2 IS THE SIZE
OF THE SPECTRUM.
C
DIMENSION DATA(LENGTH)
REAL*8 IDEN, ID WORD(175)
DIMENSION LNGTH(175),LOCATN(4)
INTEGER*2 TEMP(2)
LOGICAL CHECK
EQUIVALENCE (L, TEMP(1))

C READ CARD A
C
READ (5,100,END=80) IREAD, IDEN, LOCATN, NTHOUS
100 FORMAT (I1,I1X,A8,5I5)
IF (IREAD .LT. 1) STOP
MTHOS=LENGTH/1024
IF (NTHOUS .GT. MTHOS) NTHOUS=MTHOS
IF (IREAD .EQ. 2) GO TO 20

C READ CARDS
C
I1=LOCATN(1)+1
I2=LOCATN(2)+1
IF (I2 .GT. LENGTH) I2=LENGTH
IF (IREAD .EQ. 2) GO TO 12
READ (5,102) (DATA(I), I=1,12)
102 FCRMAT (8F7.0)
GO TO 14
12 READ (5,104) (DATA(I), I=I1,I2)
104 FCRMAT (10F7.0)
14 WRITE (6,106) I2
106 FCRMAT (** CARDS READ *** DATA SET LENGTH**,I6)
RETURN

C READ TAPES
C
20 IF (IREAD .GT. 5) GO TO 60
LREC=256
IF (IREAD .EQ. 5) LREC=1024
I1=1
I2=LREC
IF (IREAD.EQ.3) GO TC 30

C
DO 26 J=1,17
READ (8,122,END=28) ID,(DATA(I), I=I1,I2)
122 FCRMAT (I6,I8(128F6.0))
I1=I1+LREC
26 I2=I2+LREC
28 I2=I2-LREC
WRITE (6,124) ID,I2
124 FCRMAT (*DATA SET*,I7,* HAS BEEN READ FROM TAPE, LENGTH*,I6)
RETURN

C
GEO 4K ANALYZER
30 DO 32 J=1,17
READ (8,126,END=28) ID,ID1,ID2,(DATA(I), I=I1,I2)
126 FCRMAT (3I6,128F6.0,128F6.0)
I1=I1+LREC
32 I2=I2+LREC
GO TO 23

C
READ DISK

C
60 IF (IREAD.GT.6) RETURN
DEFINE FILE 10(6000.525,U,IP)
IF (LOCATN(1).GT.0) GO TO 66

C
LOCATE DATA SET
K=1
DO 64 J=1,5
READ (10*J) (IDWORD(I),LENGTH(I), I=I1,175)
62 I=1,175
IF (IDWORD(I).NE. IDEN) GO TO 62
LOCATN(K)=I+175*(J-1)
IF (LCCATN(K).GE. 750) GO TO 66
LTEMP=LENGTH(I)
NTHOUS=TEMP(2)
NTHIS=TEMP(1)*4  
IF (NTHIS .GE. NTHOUS) GO TO 66  
K=K+1  
62 CONTINUE  
64 CONTINUE  
WRITE (6,130) IDEN  
130 FORMAT('" *** ALL OF DATA SET NO. ",A8," COULD NOT BE FOUND IN TABLE OF CONTENTS")  
IF (LOCATN(1) .LE.0) STOP  
NTHCLS=(K-1)*4  
READ DATA SET  
66 JSTOP=2*NTHOUS  
JJ=1  
I1=1  
I2=512  
DO 74 K=1,4  
IFCINT=8*LOCATN(K)+1  
IF (IPOINT .LE. 8) GO TO 76  
CC 72 J=1,8  
READ (10'IPOINT) IDWORD(J),(DATA(I),I=I1,I2)  
IF (IDWORD(J) .EQ. IDEN) GO TO 67  
WRITE (6,132) IDEN,IDWORD(J),JJ  
76 WRITE (6,134) IDEN,LOCATN  
134 FORMAT ('"DATA SET ",A8," HAS BEEN READ FROM LOCATIONS",I4)
RETURN
80 STOP
END

* SUBROUTINE PKFIND
C
C  PEAK FINDING PROCEDURE REFERENCE: M.A. MARISCOTTI, NUC. INST.
C  AND METH., VOL. 50 (1967), PAGES 309-320
C
C  COMMON /COM1/ LENGTH,NFIRST,NLAST,NWINDO,DATA(8192)
C  COMMON /COM2/ MAXPKS,NPEAKS,POS(250),ENERGY(250),AREA(250),
1  FWHM(250),TAU(250),NPKS(250),FITLO(250),FITHI(250),NFITS
C  COMMON /COM3/ E0,ECH,FWHM,FWMEV,TAU0,TAUMEV,SKULO1,SKULOS,
1  SKUHII,SKUHIS,TAILI,TAILS,NSKU0,NSUH1,NL01,NHI1,MAXIT,ITQ,
1  EPSQ,EFFS,EFXLG,EFTERM,PKLMT,NRDR,NPRTR,NPNCH
DIMENSION A(8192),XM(15)
REAL FCTRd 5)/6, 2.3, .82,.184, 350., 596., 933., 1392., 1974.,
1 2700., 3586., 4648., 5902., 7364., 9050. /,NOPEAK

C  FIND SECOND DIFFERENCE
C
IMIN=NFIRST+1
IMAX=NLAST+1
I1=IMIN+1
I2=IMAX-1

DO 20 I=I1,I2
20 A(I) = DATA(I+1) - 2*C*DATA(I) + DATA(I-1)
A(IMIN)=0.
A(IMAX)=0.

C  FIND NWINDO
C
CHMID=(NFIRST+NLAST)/2
EMID=(E0+ECH*CHMID)/1000.
FWMID=FWHM+FWMEV*EMID
IF (NWINDO .EQ. 0) NWINDO=0.6*FWMID+0.5
IF (NWINDO .LE. 15) GO TO 21
WRITE (NPRTR,100) NWINDO
100 FORMAT ('-  *** ERROR--NWINDO IS',I3,' WHICH IS GREATER THAN 15') STOP

21 N1=NWINDC-1
WINDOW=NWINDO
IF (N1 .LE. 0) GO TO 32
DO 22 I=1,N1
22 XN(I)=NWINDO-I

C SMOOTH SECOND DIFFERENCE 4 TIMES WITH AN NWINDO CHANNEL WINDOW
C
I1=ININ+N1
I2=IMAX-N1
DO 30 II=1,2
DO 26 I=I1,I2
SLM=WINDOW*A(I)
DO 24 L=1,N1
24 SUM=SUM+XN(L)*(A(I-L)+A(I+L))
26 A(I-N1)=SUM
DO 28 I=I1,I2
I3=I1+I2-I
28 A(I3)=A(I3-N1)
C TAKE CARE OF ENDS
DO 30 L=1,N1
A(IMIN-1+L)=A(I1)
A(IMAX+1-L)=A(I2)
30 CONTINUE
C INITIALIZE PARAMETERS
32 N1MINUS=NWINDO+1
N1PLUS=3.0*FWMID
NOPEAK=PKLMT*WINDOW
FACTOR=FCTR(NWINDO)
J = 0

MAIN LOOP

DO 60 I = II, I MAX
F = SQRT(ABS(FACTOR*DATA(I)))/1000.
IF (DATA(I) .LT. 1.) F = SQRT(FACTOR)/1000.
S = A(I)/1000.
IF (S .GT. 0.) GO TO 42
IF (S1 .LE. 0.) GO TO 58

I3 = I
FS = F1
GO TO 58

42 IF (S1 .GE. 0.) GO TO 53
I5 = I - 1

CHECK SIZE OF REGION OF NEGATIVE SECOND DERIVATIVE

NWIDE = I5 - I3 + 1
IF (NWIDE .LT. N1MINUS .OR. NWIDE .GT. NPLUS) GO TO 58

CHECK MAGNITUDE OF NEGATIVE SECOND DERIVATIVE

SUM = 0.
DO 44 K = I3, I5
SUM = SUM + A(K)
SUM = SUM/1000.
RATIO = SUM/FS
IF (RATIO .LE. NO'FAK) GO TO 58

C
C HOORAY, WE HAVE A PEAK. COMPUTE POSITION AND AREA.
C
J=J+1
TOT=0.
DO 50 K=13,15
CH=K-1
50 TOT=TOT+A(K)*CH
PCS(J)=-TOT/(1000.*SUM)
AREA(J)=SUM*NWIDE
IF (J .GE. MAXPKS) GO TO 80
C
C FINISH MAIN LOOPS
C
58 S1=S
F1=F
60 CONTINUE
C
C NORMALIZE AREAS AND CORRECT FOR EFFICIENCY
C
80 NPEAKS=J
AMAX=0.
XOVER=EFTERM+0.1505
DO 86 J=1,NPEAKS
ENERGY(J)=A35+E0+EC-POS(J)
ELOG=ALOG10(ENERGY(J))-3.C
IF (ELOG .LT. XOVER) GC TO 82
EFF=EFFS*ELOG
GC TC 84
82 EFF=EFFS*(1.66096*(ELOG-EFMXLG)**2+EFTERM)
84 AREA(J)=AREA(J)/(10.**EFF)
IF (AREA(J) .LE. AMAX) GO TO 86
AMAX=AREA(J)
86 CONTINUE
CO 88 J=1,NPEAKS
88 AREA(J)=100.*AREA(J)/AVAX
RETURN
END
SUBROUTINE OUT1(LABEL)

COMMON /COM1/ LENGTH, NFIRST, NLAST, NWINDO, DATA(8192)
COMMON /COM2/ MAXPKS, NPEAKS, POS(250), ENERGY(250), AREA(250),
1 FWHM(250), TAU(250), FITLO(250), FITHI(250), NFITS
COMMON /COM3/ E0,ECH,FWHMO,FWMEO,TAU0,TAUMEV,SKULOI,SKULOS,
1 SKUHII,SKUHIS,TAIL,TAILS,NSKULO,NSKUHI,NL01,NHI1,MAXIT,ITQ,
1 EPS0,EFFS,FMXLG,EFTEFM,PKLM,NRD,NPRT,NPNCH
COMMON /COM4/ IFIT,IPRINT,IPUNCH,GRAPH,IFIX,MLTPLT,MXF=TSZ,EPSHT,
1 EPSX0,EPST,FITBEL,EPS,EPST,PITBEL,PITBEL,CONST,SEP,FITABV,FITBEL,
1 FITCTF,FWCTF,TACCTF,XSF
DIMENSION LABEL(20)
INTEGER FITHI,FITLO

C PRINT OUT RESULTS OF PKFIND
C
WRITE (NPRTR,100) LABEL
100 FORMAT (1H1,25X,20A4/I10)
WRITE (NPRTR,102) NFIRST, NLAST, NWINDO, PKLMNT
102 FORMAT (* PEAKFIND SEARCH BETWEEN CHANNELS ' ,I4, ' AND ' ,I4,
1 10X,*SMOOTHING WINDOW = ',I3,10X,*PEAK LIMIT PARAMETER = ',F5.1)
WRITE (NPRTR,104) FITCTF,FITBEL,E0,ECH,TAIL,TAILS
104 FORMAT (*OCUTOFFS: FIT=' ,F7.1,7X,*FITBEL = ' ,F4.1,7X,*ENERGY = ',
1 F7.2,* KEV + ',F9.6,* KEV PER CHANNEL ' ,7X,*TAIL = ' ,F6.3,' + ,
1 F6.3,* PER MEV)
WRITE (NPRTR,106) FWCTF,FITABV,FWHMO,FWMEO,SKULOI,SKULOS
106 FORMAT (*FWHM'S: ' ,F6.1,7X,*FITABV = ' ,F4.1,9X,*FWHM = ' ,F6.3,
1 ' + ',F7.3,* PER MEV',19X,*SKULO = ',F6.3,' + ',F6.3,* PER MEV')
WRITE (NPRTR,108) TAUCF,SEP,TAUCF,TAUMEV,SKUHII,SKUHIS
108 FORMAT (*FWHM'S: ' ,F7.3,* PER MEV',19X,*SKUHI = ',',F6.3,' + ',F6.3,* PER MEV'/1H-)
WRITE (NPRTR,110)
110 FORMAT ('0 PEAK NUMBER CHANNEL ENERGY',9X)
   1 'INTENSITY TO BE FIT'/10)
   12 J=1,NPEAKS
   DO IF (AREA(J) .GT. FITCTF) GO TO 10
   WRITE (NPRTR,112) J,POS(J),ENERGY(J),AREA(J)
   112 FCRMAT (12X.10X.F7.2,10X.F7.3)
   GO TO 12
   10 WRITE (NPRTR,114) J,POS(J),ENERGY(J),AREA(J)
   114 FORMAT (12X.10X.F7.2,10X.F7.3,8X,1H*)
12 CONTINUE
   IF (NPEAKS .LT. MAXPKS) GO TO 14
   WRITE (NPRTR,116) MAXPKS
   116 FORMAT ('0NUMBER OF PEAKS FOUND EXCEEDS '*,I3)
   C
   C WRITE OUT PARAMETERS FOR FITS
   C
14 WRITE (NPRTR,120)
120 FORMAT ('0SUMMARY OF PARAMETERS FOR FITS*/'FIT NPKS ENERGY FIRST LAST FWHM TAU CENTROIDS*/' NO.*,17X)
   1 'CHAN CHAN*/)
   DO 16 J=1,NPEAKS
   FWHM(J)=FWHMO+FWMEV*ENERGY(J)/1000.*
   16 TAU(J)=TAUO+TAUMEV*ENERGY(J)/1000.*
   C
   C MAIN LOOP--J LABELS PEAKS, K COUNTS FITS
   C
   C DETERMINE NUMBER OF PEAKS IN EACH FIT, ALSO 1ST AND LAST CHANNELS
   C
   J=1
   K=1
20 J=J+MLT,T-1
   FITLO(K)=POS(J)-FITBEL*FWHM(J)
   IF (FITLO(K) .LT. NFIRST) FITLC(K)=NFIRST
   DO 22 JJ=J,J1
   C
   C
IF (JJ .EQ. NPEAKS) GO TO 24
IF ((POS(JJ+1)=POS(JJ)) .GT. SEP*FWHM(JJ)) GO TO 24
IF (POS(JJ+1)+FITABV*FWHM(JJ+1) .GE. FITLO(K)+MXFTSZ) GO TO 24

22 CONTINUE
JJ=J+MLTPLT-1
24 NPKS(K)=JJ-J+1
FITHI(K)=POS(JJ)+FITABV*FWHM(JJ)
IF (FITHI(K) .GT. NLAST) FITHI(K)=NLAST
IF (FITHI(K)=FITLO(K) .GE. 4XFTSZ) FITHI(K)=FITLO(K)+4XFTSZ
WRITE (NPRT*,122) K,NPKS(K),ENERGY(J),FITLO(K),FITHI(K),FWHM(J),TAU(J),POS(I),I=J,JJ
122 FORMAT (1X,13,15,F9.1,217*F7.2,9F9.2)
C
PUNCH CARDS IF IPUNCH=1 OR 3
C
IF ( IPUNCH .EQ. 0 0R 3PUNCH .EQ. 2 ) GO TO 28
WRITE (NPARCH,124) NPKS(K),ENERGY(J),FITLO(K),FITHI(K),K
124 FORMAT (122*111 F6.1,' KEV',10X,2(5,37X,I3)

DO 26 I=J,JJ
WRITE (NPARCH,126) POS(I)
26 CONTINUE
126 FORMAT ('11',18X,F10.2)

28 IF (JJ .GE. NPEAKS) GO TO 30
J=JJ+1
K=K+1
GC TO 20
C
30 NFITS=K
WRITE (NPARCH,130) NFITS
130 FORMAT (1H0/1H0,20X,'NUMBER OF FITS IS ',I3)
RETURN
END
SUBROUTINE K=IT
C
COMMON /COM1/ LENGTH,NFIRST,NLAST,NWINDO,DATA(8192)
PKFT0010
PKFT0020
PKFT0030
CCMMCN /COM2/ MAXPKS,NPEAKS,POS(250),ENERGY(250),AREA(250),
1 FWHM(250),TAU(250),NPKS(250),FITLO(250),FITHI(250),NFITS
CCMMCN /COM3/ E0,ECH,FWHMO,FWMEV,TAU0,TAUMEV,SKULOI,SKULOS,
1 SKUHI,SKUHIS,TAILI,TAILS,NSKULO,NSKUHI,NLOI,NHI,MAXIT,ITQ,
1 EPSG,EFFS,EFMLXLG,EFTERM,PKLMT,NPDR,NPRT,NPNCH
CCMMCN /COM4/ IFIT,IPRINT,IPUNCH,IGRAPH,IFIX,MLTPLT,MXFTSZ,EPSSH,
1 EPSX0,EPFSW,EPSTAU,EPSA,EPSB,DIFLIN,CONST,SP,FITABV,FITHEL,
1 FITCTF,FWCTF,TAUCFT,IFS
COMMON /COM6/ W(200),X(200),Y(200),NPTS,EPAS(38),STDE(38),
1 VAR,ICONV,ITRATN,SCAPEA
COMMON /CM7/ P(38),IHOLO(38),DERIV(38),TERM(10),ALPHA1(9),
1 ALPHA2(9),BETA1(9),BETA2(9),BETA3(9),TAIL(9),SKULO(9),SKUHI(9),
1 NPRMS,NP,NFREE,NJ,I OPT,XFIRST,BPRIME
COMMON /CM8/ NFRSV(250),ICNVSV(250),ITRNSV(250),VARSV(250),
1 HEIGHT(250),SDHT(250),SDXO(250),SDFW(250),SDTAU(250),ASV(250),
1 BSV(250),LFIT(252)
REAL*EV
INTEGER FITHI,FITLO
INTEGER*2 NFRSV,ICNVSV,ITRNSV
LOGICAL*1 LFIT
C READ CARD(S) E
C IF (IFIT .NE. 2) GO TO 6
C READ (NRDR,110) (LFIT(L),L=1,NFITS)
110 FORMAT (80L1)
C MAIN LOOP--L LABELS FITS, J LABELS PEAKS, K LABELS PAPAMETERS
C
6 JMAX=0
DO 50 L=1,NFITS
  JWIN=JMAX+1
  JMAX=JMAX+NPKS(L)
C
C DECIDE WHETHER TO PERFORM FIT
IF (IFIT .EQ. 2) GO TO 10
DO 8 J=JMIN, JMAX
IF (AREA(J) .LT. FITCTF) GO TO 8
LFIT(L)=.TRUE.
GC TC 12
8 CONTINUE
LFIT(L)=.FALSE.
GO TO 50
10 IF (.NOT. LFIT(L)) GO TO 50

C  INITIALIZE X(CHANNEL NUMBER) AND Y(COUNTS PER CHANNEL) ARRAYS

12 IMIN=FITLO(L)
IMAX=FITHI(L)
NFTS=IMAX-IMIN+1
X(1)=IMIN
XFIRST=X(1)
Y(1)=DATA(IMIN+1)
IF (Y(1) .LT. 1.0) Y(1)=0.5
YMAX=Y(1)
DO 14 I=2,NPTS
X(I)=X(I-1)+1.
Y(I)=DATA(IMIN+I)
IF (Y(I) .LT. 1.0) Y(I)=0.5
IF (Y(I) .LE. YMAX) GO TO 14
YMAX=Y(I)
14 CONTINUE

C  INITIALIZE WEIGHT ARRAY

C1=1.0-CONST
C2=CCNST*YMAX
D2=DIFLIN*DIFLIN
DO 16 I=1,NPTS
16 \ W(I)=1.0/(Y(I)*(C1+0.2*Y(I))+C2)

C
C INITIALIZE PARAMETER, IHOLD, AND EPS ARRAYS
C
C DO BACKGROUND PARAMETERS FIRST
C
NJ=NPKS(L)
NP=4*NJ
NPRMS=NP+2
P(NP+1)=Y(1)
BPRIME=(Y(NPTS)-Y(1))/(X(NPTS)-X(1))
P(NP+2)=BPRIME+1000.
IHOLD(NP+1)=1
IHOLD(NP+2)=1
EPS(NP+1)=EPSA
EPS(NP+2)=EPSB

C INITIALIZE PEAK PARAMETERS
K=1
DO 20 J=JMIN,JMAX
C
P(K+1) = CENTROID
P(K+1)=POS(J)
IHOLD(K+1)=1
EPS(K+1)=EPSXC
IPOS=POS(J)+1.5
C
P(K) = HEIGHT
P(K)=DATA(IPOS)-P(NP+1)-BPRIME*(P(K+1)-XFIRST)
IF (P(K) .LT. 1.0) P(K)=1.0
IHOLD(K)=1
EPS(K)=EPSHT
C
P(K+2) = FWHM
P(K+2)=FWHM(J)
IHOLD(K+2)=1
IF (AREA(J) .LT. FWCTF) IHOLD(K+2)=0
EPS(K+2)=EPSFW
C
P(K+3) = TAU
P(K+3)=TAU(J)
IHCLD(K+3) = 1
IF (AREA(J) .LT. TAUCTF) IHOLD(K+3) = 0
EPS(K+3) = EPSSTAU
20 K = K + 4
C
C INITIALIZE QUANTITIES NEEDED FOR FUNCTION EVALUATION
C
K = 1
DO 22 J = 1, NJ
MEV = ENERGY(J+JMIN-1)/1000.
ALPHA1(J) = 1.66511/P(K+2)
ALPHA2(J) = P(K+3) * ALPHA1(J)
TAIL(J) = TAIL + TAIL * MEV
SKULO(J) = SKULO + SKULCS * MEV
SKUHI(J) = SKUHI + SKUHIS * MEV
BETA1 (J) = P(K) * TAIL(J)
BETA2(J) = NSKULO * P(K) * SKULO(J)
BETA3(J) = NSKUHI * P(K) * SKUHI(J)
22 K = K + 4
C
C DETERMINE NUMBER OF FREE PARAMETERS
C
NFREE = 0
DO 24 K = 1, NPRMS
IF (IHOLD(K) .EQ. 1) NFREE = NFREE + 1
24 CONTINUE
C
C PRINT HEADING FOR NLLS OUTPUT
C
IF (IPPINT .EQ. 0) GO TO 26
WRITE (NPRTR, 100) LIMIN, IMAX, JMIN, JMAX
WRITE (NPRTR, 102) ENERGY(JMIN), P(3), P(4), TAIL(I), SKULO(I), SKUHI(I)
102 FORMAT ('1-PARAMETERS FOR PEAK(1): ENERGY =', F7.1, ' KEV = WHM')
1 = 'F7.3,4X, TAU = 'F7.3,4X,'TAU = 'F7.3,3X,'SKUHI = 'F7.3/1H0)' 

PERFORM FIT

26 CALL NLLS(IPRINT)

SAVE RESULTS OF FIT

NFRSV(L) = NFREE
VARSV(L) = VAR
ICNVSv(L) = ICONV
ITRNSV(L) = ITRATN
ASV(L) = P(NP+1)
BSV(L) = BPRIME
K = 1
DO 30 J = JMIN, JMAX
HEIGHT(J) = P(K)
PCS(J) = P(K+1)
FWHM(J) = P(K+2)
TAU(J) = P(K+3)
SCHT(J) = STDE(K)
SDX0(J) = STDE(K+1)
SDFW(J) = STDE(K+2)
SDTAL(J) = STDE(K+3)
ENERGY(J) = E0 + ECH * POS(J)
30 K = K + 4

END OF MAIN LOOP

50 CONTINUE
RETURN
END
SUBROUTINE NLLS(IPRINT)
REAL*8 GET, AA(1444), A*4 (76, 39)
EQUIVALENCE (AA(1), A(1))
DIMENSION MPRNT(8), QPRNT(8), B(38), IND1(38), IND2(38), DELTAP(38)
EQUIVALENCE (IND1(1), DELTAP(1))
REAL MEV

INITIALIZATION

ICCNV=1
IOPT=1
NNF=2*NFREE
M=1
ITRATN=C
O=1.E20
MPPNT(1)=0
L=0
DO 8 K=1, NPRMS
STDE(K)=0.
IF (IHOLD(K) .NE. 1) GC TO 8
L=L+1
PPRNT(L,1)=P(K)
8 CONTINUE

EVALUATE ERROR MATRIX
10 ITSTN=ITSTN+1
   IF (ITSTN .LE. MAXIT) GO TO 12
   ICCNV=6
   GO TO 50
12 DO 14 K=1,NFREE
   B(K)=0.
   DO 14 L=1,NNF
14 A(L,K)=0.
   OLDQ=Q
   Q=0.
   DO 18 I=1,NPTS
      CALL FUNCT(X(I),YC)
      DIFF=Y( I )-YC
      DO 16 K=1,NFREE
      DW=W(I)*DERIV(K)
      B(K)=B(K)+DW*DIFF
      DO 16 L=K,NFREE
      A(2*L-1,K)=A(2*L-1,K)+DW*DERIV(L)
16 A(2*L-1,K)=A(2*L-1,K)+DW*DERIV(L)
18 Q=Q+W(I)*DIFF*DIFF
   GFFNT(M)=Q
   DO 20 K=2,NFREE
      K1=K-1
      DO 20 L=1,K1
20 A(2*L-1,K)=A(2*K-1,L)

C CHECK FOR Q CONVERGENCE. CHECK FOR FULL PRINT ARRAYS.
   IF (ITSTN .LT. ITQ) GO TO 24
   IF (Q .LE. OLDQ) GO TO 22
   ICCNV=2
   GO TO 50
22 IF ((OLDQ/Q-1.0.) .GT. EPSQ) GO TO 24
   ICONV=3
   GO TO 50
24 CONTINUE
30 IF (M .EQ. 8) GO TO 50
C INVERT MATRIX AND GET NEW PARAMETERS
C
32 CALL DUMNV(AA,NREF,38,DET,IND1,IND2)
   IF (DET .NE. 0 .OR. 0) GO TO 34
   ICONV=4
   GO TO 50
34 DO 36 K=1,NFREE
   DELTAP(K)=0.
   DO 36 L=1,NFREE
36 DELTAP(K)=DELTP(K)+A(2*L-1,K)*B(L)
   ICONV=5
   M=M+1
   L=C
   DO 38 K=1,NPRMS
   IF (I1HOLP(K),NE. 1) GC TO 38
   L=L+1
   IF (I1DD(K,4),NE. 2 ) GO TO 37
   IF (K .GT. NP ) GO TO 37
   IF (ABS(DELTAP(L)) .GT. .5 ) DELTAP(L)=SIGN(.5,DELTAP(L))
   THIS STATEMENT LIMITS CENTROID CHANGE TO .5 CHANNELS
37 DF=DELTP(L)/P(K)
   IF (ABS(DP) .GT. EPS(K)) ICONV=1
   IF (ABS(DP) .GT. 0.5) DP=SIGN(0.5,DP)
   THIS STATEMENT LIMITS ALL PARAMETER CHANGES TO .5 TIMES PREVIOUS VALUE
   P(K)=P(K)*(1.0+DP)
   PFRNT(L+M)=P(K)
   CONTINUE
   MPRNT(M)=ITRATN
   K=1
   DO 42 J=1,NJ
   MEV=(E0+ECH*P(K+1))/1000.0
   ALPHA1(J)=1.66511/P(K+2)
ALPHA2(J) = P(K+3) * ALPHA1(J)  
TAIL(J) = TAIL1 + TAIL_S * MEV  
SKULO(J) = SKULO1 + SKULO_S * MEV  
SKUHI(J) = SKUHI1 + SKUHI_S * MEV  
BETA1(J) = P(K) * TAIL(J)  
BETA2(J) = NSKULO * P(K) * SKULO(J)  
BETA3(J) = NSKUHI * P(K) * SKUHI(J)  

K = K + 4  
BPRIME = P(K+1) - 1000.  

L = 0  
LL = -1  
DO 44 K = 1, NPRMS  
IF (IHOLD(K) .NE. 1) GC TO 44  
L = L + 1  
LL = LL + 2  
STDE(K) = ABS(A(L,L))  
44 CONTINUE  

RETURN FOR NEXT ITERATION UNLESS ICONV = 5  

IF (IPRINT .LE. 0) GO TO 58  
WRITE (NPRINT,100)  
100 FORMAT("VALUES OF FREE PARAMETERS AFTER EACH ITERATION",/1H0.,  
1 26X,"ITERATION")  
WRITE (NPRINT,102) (MPRNT(L), L = 1, M)  
102 FORMAT(" PARAM",16,7I14)  
L = 0  
DO 52 K = 1, NPRMS
IF (IHOLD(K) .NE. 1) GO TO 52
L=L+1
WRITE (NPRTR,104) K,(PFRNT(L,MM),MM=1,M)
104 FORMAT (I3,1P8E14.5)
52 CONTINUE
IF (ICONV .LT. 5) GO TO 56
Q=0.
IOPC=C
DC 54 I=1,NPTS
CALL FUNCT(X(I),YC)
54 Q=Q+W(I)*(YC-Y(I))**2
QFRNT(M)=Q
56 WRITE (NPRTR,106) (QFRNT(L),L=1,M)
106 FORMAT ('Q',1P8E14.5)
58 M=0
IF (ICONV .EQ. 1) GO TO 32
IF (IFRINT .EO. 0) GO TO 70
GO TO (32,60,62,64,66,68),ICONV
60 WRITE (NPRTR,110)
110 FORMAT ('ITERATION STOPPED BECAUSE Q INCREASED')
L=0
DC 61 K=1,NFREE
IF (IHOLD(K) .NE. 1) GO TO 61
L=L+1
P(K)=P(K)-DELTAP(L)
61 CONTINUE
G=CLOG
GO TO 70
62 WRITE (NPRTR,112) FPSQ
112 FORMAT ('Q DECREASED BY LESS THAN',1PE3.1)
GO TO 70
64 WRITE (NPRTR,114) ITRATN
114 FORMAT ('MATRIX A IS SINGULAR ON ITERATION',I3)  
GO TO 70  
66 WRITE (NPRTR,116) IT.RATN  
116 FORMAT ('CONVERGED IN',I3,' ITERATIONS')  
GO TO 70  
68 WRITE (NPRTR,118) MAXIT  
118 FORMAT ('DID NOT CONVERGE IN',I3,' ITERATIONS')
C
C FIND VARIANCE AND ERRORS
C
70 VAR=Q/FLOAT(NPTS-NFREE)
SUMW=0.
DO 72 I=1,NPTS
72 SUMW = SUMW+W(I)
VSW=VAR/SUMW
IF (IFRINT .EQ. 0) GO TO 74
WRITE (NPRTR,120) VAR,0,VSW
120 FORMAT ('VARIANCE = ',F6.2,'= ',1PE11.4,'= ',VARIANCE/SUM OF WIGHTEIS =',E11.4)
74 DO 76 K=1,NPRMS
76 STDE(K)=SQRT(STDE(K)*VAR)
RETURN
ENC
SLBROUTINE = FUNCT(X,Y)
C
COMMON /COM3/ E0,ECH,FWHM,FWMEV,TAU>T,AUMEV,SKU_0I,SKU_J5,SK
1 SKUHII,SKUHIS,TAIL1,TAILS,NSKUL0,NSKUH1,NLDI,NHI1,MAXIT,ITQ,
1 EPSQ,EFSS,EFMRG,EFTERM,PKLMT,NDIR,NNPRTR,NPNCH
COMMON /COM7/ P(38),IHCLD(38),DERIV(38),TERM(10),ALPHA1(9),
1 ALPHA2(9),BETA1(9),BETA2(9),BETA3(9),TAIL(9),SKUL0(9),SKUH1(9),
1 NPRMS,NP,NFREE,NJ,I0PT,XFIST,BPRIMF
INTERGER REGION
C
C FIND BACKGROUND FIRST
X1 = X - XFIRST
TERM(NJ+1) = P(NP+1) + BPRIME*X1
Y = TERM(NJ+1)

C
C MAIN LOGIC
C IF IOPT = 0, DERIVATIVES ARE NOT TO BE CALCULATED
C
K = 1
L = 0
DC 50 J = 1, NJ
Z = ALPHA1(J)*(X - P(K+1))
Z1 = -Z
Z2 = Z*Z
IF (Z1 .GT. 0.0) Z2 = Z2 - Z1*Z1
IF (Z2 .GT. 25.0) GO TO 8
ZETA = EXP(-Z2)
GO TO 10
8 ZETA = 0.0
C
C Z1 > 0 DEFINES REGION I
C
10 IF (Z1 .LE. 0.0) GO TO 12
REGION = 1
Z = Z1
GAMMA = Z**NL01
DELTA = P(K)*ZETA*(1.0 - TAIL(J) + SKULO(J)*GAMMA*Z)
TERM(J) = DELTA + BETA1(J)
Y = Y + TERM(J)
IF (IOPT .EQ. 0) GO TO 50
DELTA2 = 2.0*DELTA
ETA = ZETA*GAMMA*BETA2(J)
THETA = ALPHA2(J)
GO TO 20

C
C Z < 0 AND Z1 < 0 DEFINE REGION II
12 IF (Z < 0.0) GO TO 14
    REGION=2
    GAMMA=0.0
    DELTA=P(K)*ZETA*(1.0-TAIL(J))
    TERM(J)=DELTA+BETA1(J)
    Y=Y+TERM(J)
    IF (IOPT .EQ. 0) GO TO 50
    DELTA2=2.0*DELTA
    ETA=0.0
    THETA=Z
    GO TO 20

C  
Z > 0 DEFINES REGION III
C
14 REGION=3
    GAMMA=Z**NHI1
    DELTA=P(K)*ZETA*(1.0+SKUHI(J)*GAMMA*Z)
    TERM(J)=DELTA
    Y=Y+TERM(J)
    IF (IOPT .EQ. 0) GO TO 50
    DELTA2=2.0*DELTA
    ETA=ZETA*GAMMA*BETA3(J)
    THETA=Z
C
C  CALCULATE DERIVATIVES
C
20 IF (IHOLD(K) .EQ. 0) GO TO 24
    DRV=TERM(J)/P(K)
    L=L+1
    DERIV(L)=DRV
24 IF (IHOLD(K+1) .EQ. 0) GO TO 28
    DRV=(DELTA2*THETA-ETA)*ALPHA1(J)
    IF (REGION .EQ. 1) DRV=-DRV
    L=L+1
DERIV(L)=DRV
28 IF (IHOLD(K+2) .EQ. 0) GO TO 32
   DRV=(DELTA2*Z2-Eta*Z)/F(K+2)
   L=L+1
   DERIV(L)=DRV
32 IF (IHOLD(K+3) .EQ. 0) GO TO 50
   DRV=0.
   IF (REGION .EQ. 1) DRV=-ALPHA1(J)*(Eta+DELTA2*Z)
   L=L+1
   DERIV(L)=DRV
50 K=K+4
   DERIV(L+1)=1.0
   DERIV(L+2)=X1
   RETURN
END
SUBROUTINE DUMNV(A,N,NN,D,L,M)

C
C PURPOSE
C INVERT A MATRIX USING DOUBLE PRECISION ARITHMETIC.
C
C USAGE
C CALL DUMNV(A,N,NN,D,L,M)
C
C DESCRIPTION OF PARAMETERS
C A - DOUBLE PRECISION INPUT MATRIX, DESTROYED IN COMPUTATION
C AND REPLACED BY THE RESULTANT INVERSE.
C N - ORDER OF MATRIX A.
C NN- FIRST DIMENSION OF THE A ARRAY.
C D - RESULTANT DETERMINANT (DOUBLE PRECISION).
C L - INTEGER WORK VECTOR OF LENGTH N.
C M - INTEGER WORK VECTOR OF LENGTH N.
C
C METHOD
C THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT THE MATRIX IS SINGULAR.

**DIMENSION A(1), L(1), M(1)**

**DOUBLE PRECISION A, D, BIGA, HOLD**

**SEARCH FOR LARGEST ELEMENT**

D = 1.0
NK = NN
DO 80 K = 1, N
NK = NK + NN
L(K) = K
M(K) = K
KK = NK + K
BIGA = A(KK)
DO 20 J = K, N
IZ = NN*(J - 1)
DO 20 I = K, N
IJ = IZ + I
10 IF(DABS(BIGA) - DABS(A(IJ))) < 15, 20, 20
15 BIGA = A(IJ)
L(K) = I
M(K) = J
20 CONTINUE

**INTERCHANGE ROWS**

J = L(K)
IF(J - K) < 35, 35, 25
KI = K - NN
DO 30 I = 1, N
KI = KI + NN
HOLD = A(KI)
JI = KI + J
30 CONTINUE
A(KI) = A(JI)

30 A(JI) = HOLD

C INTERCHANGE COLUMNS

35 I=M(K)
   IF(I-K) 45, 45, 38
   JF=NN*(I-I)
   DO 40 J=1,N
       JK=NK+J
       JI=JP+J
       HOLD=-A(JK)
       A(JK) = A(JI)
   40 A(JI) = HOLD

C DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA)

45 IF(BIGA) 48, 46, 48
   D=0.0
   RETURN
   48 DO 55 I=1,N
       IF(I=K) 50, 55, 50
       IK=NK+I
       A(IK) = A(IK)/(-BIGA)
   55 CONTINUE

C REDUCE MATRIX

   DO 65 I=1,N
       IK=KK+I
       HOLD=A(IK)
       IJ=I-NN
       DO 65 J=1,N
           IJ=IJ+NN
   65 CONTINUE
IF(I-K) 60,65,60
60 IF(J-K) 62,65,62
62 KJ=IJ-I+K
   A(IJ)=HOLD*A(KJ)+A(IJ)
65 CONTINUE.

C    DIVIDE ROW BY PIVOT
C
KJ=K-NN
DO 75 J=I,N
   KJ=KJ+NN
   IF(J-K) 70,75,70
70 A(KJ)=A(KJ)/BIGA
75 CONTINUE

C    PRODUCT OF PIVOTS
C
   D=D*BIGA
C    REPLACE PIVOT BY RECIPROCAL
C
   A(KK)=1.DO/BIGA
80 CCNTINUE

C    FINAL ROW AND COLUMN INTERCHANGE
C
K=N
100 K=(K-1)
   IF(K) 150,150,105
105 I=L(K)
   IF(I-K) 120,120,108
108 JQ=NN*(K-1)
   JR=NN*(I-1)
   DO 110 J=1,N
   JK=JQ+J
110 CONTINUE
HOLD = A(JK)
JI = JR + J
A(JK) = -A(JI)
110 A(JI) = HOLD
120 J = M(K)
   IF J = K THEN 100, 100, 125
125 KI = K - NN
   DO 130 I = 1, N
   KI = KI + NN
   HCLD = A(KI)
   JI = KI - K + J
   A(KI) = -A(JI)
130 A(JI) = HCLD
   GO TO 100
150 RETURN
END

SUBROUTINE OUT2 (LABEL)

COMMON /COM1/ LENGTH, NFIRST, NLAST, NWIND3, DATA(8192)
COMMON /COM2/ MAXPKS, NPEAKS, POS(250), ENERGY(250), AREA(250)
1 FWHM(250), TAU(250), NPKS(250), FITLO(250), FITHI(250), NFITS
COMMON /COM3/ E0, ECH, FWHMO, FWMEV, TAUJ, TAUMEV, SKU_0, SKU_J,
1 SKUHI, SKUHIS, TAIL, TAILS, NSKULO, NSKUHI, NLO1, NH1, MAXIT, IT0,
1 EPS3, EFFS, EFMLG, EFTERM, PKLMT, NRDR, NPRTR, NPNCH
COMMON /COM4/ IFIT, IPRINT, IPUNCH, IGRAPH, IFIX, MLTPLT, MXFTSZ, EPSHT,
1 EPSX0, EPSFW, EPSDAU, EPSA, EPSB, DIFLIN, CONST, SEPI, FITABV, FITBEL,
1 FITCTF, FWCTF, TAUCTF, XSF
COMMON /COM7/ P(38), IHOLO(38), DEPIV(38), TERM(10), ALPHA1(9),
1 ALPHAI(9), BETAI(9), BETAI(9), TAIL(9), SKUO(9), SKUHI(9),
1 NPRMS, NE, NFREE, NJ, IOPT, XFIRST, BPRIME
COMMON /COM8/ NFRSV(250), ICNVSV(250), ITNSV(250), VARSV(250),
1 HEIGHT(250), SDHT(250), SDOX(250), SDFW(250), SDTAU(250), ASV(250),
1 ESV(250), LFIT(252)
C
DIMENSION X(210), Y(210), YCAL(200), XC(201), LABEL(25)
INTEGER FITHI,FITLO
INTEGER NFRSV,ICNVSV,ITRNSV
LOGICAL LFIT
LOGICAL LABEL2(20)
REAL XLAB(5)
REAL MEV

PRINT HEADINGS
WRITE (NPRTR,100) LABEL
100 FORMAT (1H1,25X,20A4/1H0)
WRITE (NPRTR,102) NFIRST,NLAST,NWINDX,PKLM
102 FORMAT (' PEAK FIND SEARCH BETWEEN CHANNELS ' ,I4, ' AND ' ,I4, 
1 10X,' SMOOTHING WINDOW =',I3,1DX,' PEAK _LIMIT _PARAMETER =',F5.1)
WRITE (NPRTR,104) FITCTF,FITBEL,E0,ECH,TAILI,TAILS
104 FORMAT (' 0 CUTOFFS: FIT,','F7.1,7X,'FITBEL =','F4.1,7X,'ENERGY =', 
1 F7.2,' KEV +=','F9.6,' KEV PER CHANNEL',7X,'TAIL =','F6.3,' ++, 
1 F6.3,' PER MEV')
WRITE (NPRTR,106) FWCTF,FITABV,FWHMO,FMMEV,SKU01,SKU05
106 FORMAT (11X,'FWHM','=6.1,7X,'FITBEL =','F4.1,9X,'FWHM =','F8.3, 
1 '+'F7.3,' PER MEV',19X,'SKU01 =','F6.3,' ++',F6.3,' PER MEV')
WRITE (NPRTR,108) TAUCTF,SEP,TAMU,TAMEV,SKUH1,SKUH5
108 FORMAT (11X,'TAU','F7.1,10X,'SEP','=4.1,10X,'TAU','=8.3,' ++, 
1 F7.3,' PER MEV',19X,'SKUH1 =','F6.3,' ++',F6.3,' PER MEV/1H-')
WRITE (NPRTR,110)
110 FORMAT (40X,'SUMMARY OF FITTING RESULTS')
WRITE (NPRTR,112)
112 FORMAT ('FIT NPKS NFREE NPTS VAR ICNV NIT'/25X,'ENERGY', 
1 'HEIGHT & ERROR CENTROID & ERROR FWHM & ERROR', 
1 'TAU & ERROR AREA & ERROR & % ERROR')

MAIN LOOP--L LABELS FITS, J LABELS PEAKS
IOPT=0
JMAX=0
DO 70 L=1,NFITS
JMIN=JMAX+1
JMAX=JMAX+NPKS(L)
IF (.NOT. LFIT(L)) GO TO 70

C INITIALIZE X AND Y ARRAYS

C
IMIN=FITLO(L)-5
IF (IMIN .LT. NFIRST) IMIN=NFIRST
IMAX=FITHI(L)+5
IF (IMAX .GT. NLAST) IMAX=NLAST
NEXPTS=IMAX-IMIN+1
NFTS=FITHI(L)-FITLO(L)+1
X(1)=IMIN
Y(1)=DATA(IMIN+1)
IF (Y(1) .LT. 1.0) Y(1)=0.5
DO 14 I=2,NEXPTS
X(I)=X(I-1)+1.
Y(I)=DATA(IMIN+I)
IF (Y(I) .LT. 1.0) Y(I)=0.5
14 CONTINUE

C FIND PARAMETERS FROM FIT

C
NJ=NPKS(L)
NP=4*NJ
NPRMS=NP+2
P(NP+1)=ASV(L)
BFRIME=ESV(L)
K=1
DO 20 J=JMIN,JMAX
P(K)=HEIGHT(J)
20 CONTINUE

OUT20550
OUT20560
OUT20570
OUT20580
OUT20590
OUT20600
OUT20610
OUT20620
OUT20630
OUT20640
OUT20650
OUT20660
OUT20670
OUT20680
OUT20690
OUT20700
OUT20710
OUT20720
OUT20730
OUT20740
OUT20750
OUT20760
OUT20770
OUT20780
OUT20790
OUT20800
OUT20810
OUT20820
OUT20830
OUT20840
OUT20850
OUT20860
OUT20870
OUT20880
P(K+1) = PCS(J)
P(K+2) = FWHM(J)
P(K+3) = TAU(J)

20 K = K + 4

C FIND ALPHAS, BETAS, ETC.

K = 1
DO 22 J = 1, NJ
MEV = ENERGY(J + JMIN - 1) / 1000.
ALPHA1(J) = 1.66511 / P(K+2)
ALPHA2(J) = P(K+3) * ALPHA1(J)
TAIL(J) = 0.0
SKULC(J) = SKULOI * SKU LCS * MEV
SKUHI(J) = SKUHI1 * SKUHI S * MEV
BETA1(J) = 0.0
BETA2(J) = NSKULO * P(K) * SKULO(J)
BETA3(J) = NSKUHI * P(K) * SKUHI(J)

22 K = K + 4

C FIND AREAS

DO 24 J = JMIN, JMAX
AREA(J) = 0.
DO 30 I = 1, NE XPTS
CALL FUNCT(X(I), YC)
DO 26 J = JMIN, JMAX
AREA(J) = AREA(J) + TER * (J - JMIN + 1)

30 CONTINUE

DO 32 J = 1, NJ
TAIL(J) = TAIL1 + TAIL S * MEV
BETA1(J) = TAIL(J) * P(4*J - 3)

32 CONTINUE

C WRITE RESULTS
C OUT21240
WRITE (NPRT3.120) L, NPKS(L), NFPSVL(L), NPTS, VARSV(L), ICNVSV(L).
1 ITRNSV(L)
120 FCPMAT (1H0, I3, I5, I7, I6, I1X, F6, 2, I5, I5)
DO 36 J = JMIN, JMAX
ERAREA = SQRT((SDHT(J)/HEIGHT(J))**2 + (SDFW(J)/FWHM(J))**2)
SCAREA = ERAREA*AREA(J)
WRITE (NPRT3.122) ENERGY(J), HEIGHT(J), SDHT(J), POS(J), SDX0(J),
1 FWHM(J), SDFW(J), TAU(J), SDTAU(J), AREA(J), SDAREA, ERAREA
1 1PE13, 3, E11, 1, 2 PF6.1)
IF (IPUNCH .LT. 2) GO TO 36
WRITE (NPCH.124) POS(J), SDXC(J), AREA(J), HEIGHT(J),
1 FWHM(J), SD^WC(J), TAU(J), SDTAU(J)
124 FCRMAT (2F10, 2, 3F10, 1, 4F5, 2)
36 CONTINUE
C DC GRAPHING
C
IF (IGRAPH .EQ. 0) GO TO 70
38 XMIN = X(1) - AMOD(X(1), 10.)
IF (XSF .LT. 4.0) XSF = 4.0
IF (XSF .GT. 20.) XSF = 20.
XSIZE = (X(NEXP) - XMIN) / XSF
XSIZE = XSIZE + 0.5 - AMOD(XSIZE, 0.5)
IF (XSIZE .LT. 5.0) XSIZE = 5.0
YSIZE = 8.0
C PUT FIT NUMBER INTO POSITIONS 5, 6, AND 7 OF LABL2
C
IDUM = L/100 + 240
LABL2(5) = LDUM(4)
IDUM = MOD(L/10, 10) + 240
LABL2(6) = LDUM(4)
IDUM = MOD(L, 10) + 240
LABL2(7) = LDUM(4)

C SET LP GRAPH ARRAYS

C

YMN = 1.0E8
YMx = 1.
DO 40 I = 1, NEXPTS
Y(I) = SQRT(Y(I))
IF (Y(I) LT YMN) YMN = Y(I)
IF (Y(I) GT YMX) YMX = Y(I)
40 CONTINUE

XC(1) = FITLO(L)
DO 42 I = 1, NPTS
CALL FUNCT(XC(I), YC)
YC = SQRT(ABS(YC))
IF (YC LT YMN) YMN = YC
IF (YC GT YMX) YMX = YC
YCAL(I) = YC
XC(I+1) = XC(I) + 1.0
42 CONTINUE

C

C FIND Y SCALE FACTOR AND SYMBOL SIZE

C

YMN = YMN - 0.5
YSF = (YMX - YMN) / (FSIZE - 1.0)
IYSF = ALOG(YSF) + 1.0
IF (IYSF GT 6) GO TO 70
GO TO (50, 52, 54, 56, 58, 60), IYSF
YSF = 1.0
GC TC 62
50 YSF = YSF + C.5 - AMOD(YSF, 0.5)
GC TO 62
52 YSF = YSF + 1.0 - AMOD(YSF, 1.0)
GC TO 62
54 YSF = YSF + 2.0 - AMOD(YSF, 2.0)
GO TC 62
56 YSF=YSF+5.0-AMOD(YSF,5.0)
GO TO 66
58 YSF=YSF+20.0-AMOD(YSF,20.0)
GO TO 66
60 YSF=YSF+50.0-AMOD(YSF,50.0)
GO TO 66
C
62 YFIR=YMN-AMOD(YMN,YSF)
SIZE=1.0/YSF
CALL ORIGIN(SIZE,0.5)
CALL GRAPH(NEXPTS,X,Y,13,7,XSIZE,YSIZE,XSF,XMIN,YSF,YFIR,XLAB,
1 YLAB,LABL,LABL2)
GO TC 68
66 CALL ORIGIN(0.08,0.5)
YFIR=YMN-AMOD(YMN,YSF)
CALL GRAPH(NEXPTS,X,Y,2,7,XSIZE,YSIZE,XSF,XMIN,YSF,YFIR,XLAB,
1 YLAB,LABL,LABL2)
68 CALL GRAPH(NPTS,XC,YCAL,2,2.0,5,YSIZE,XSF,XMIN,YSF,YFIR,J,0,0)
70 CONTINUE
C
END OF MAIN LOOP
C
WRITE (NPRTR,126)
126 FORMAT ('EXPLANATION OF ICONV: 2 - Q INCREASED ON LAST ITERATION',
1'/24X,'3 - Q DECREASED BY LESS THAN EPSQ',
1'/24X,'4 - ERR OR MATRIX WAS SINGULAR',
1'/24X,'5 - CONVERGED NORMALLY',
1'/24X,'6 - DID NOT CONVERGE',
1 IN MAXIT ITERATIONS',
1'/24X,'100')
RETURN
END
INSERT MAIN
OVERLAY ONE
INSERT INPUT
OVERLAY ONE
INSERT PKFIND
INSERT OUT1
    OVERLAY ONE
INSERT FUNCT
INSERT COM6
INSERT COM7
INSERT CCM8
    OVERLAY TWO
INSERT FKFIT
INSERT NLL5
INSERT CUMMV
    OVERLAY TWO
INSERT OUT2