UCRL-52917

GRPANL: a program for fitting complex peak groupings for gamma and x-ray energies and intensities

Raymond Gunnink
Wayne D. Ruhter

January, 1980

Lawrence Livermore Laboratory

CIRCULATION COPY
SUBJECT TO RECALL
IN TWO WEEKS

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately 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 the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

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

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

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

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

Best Available Quality

for original report

call
Reports Library

X37097

UCRL-52917
Distribution Category UC-10

GRPANL: a program for fitting complex peak groupings for gamma and x-ray energies and intensities

Raymond Gunnink
Wayne D. Ruhter

Manuscript date: January, 1980

LAWRENCE LIVERMORE LABORATORY
University of California • Livermore, California • 94550

Available from: National Technical Information Service • U.S. Department of Commerce
5285 Port Royal Road • Springfield, VA 22161 • $7.00 per copy • (Microfiche $3.50)

CONTENTS

Abstract 1
Introduction 1
Fitting Technique and Model 2
    Peak Shape Algorithm 2
    Delineation of Background 4
    Fitting Process 6
Gamma Intensity Calculations 8
GRPANL Operation: An Overview 10
Execution 13
References 17
Appendix: Source Listings 18

GRPANL: a program for fitting complex peak groupings for gamma and x-ray energies and intensities

ABSTRACT

GRPANL is a general-purpose peak-fitting program that calculates gamma-ray and x-ray energies and intensities from a given spectral region. The program requires that the user supply input information such as the first and last channels of the region, the channels to be used as pre- and post-region background, the system gain and zero-intercept, and a list of approximate energy values at which peaks occur in the region. Because the peak position and peak-shape parameters enter nonlinearly into the peak-fitting algorithm, an iterative least-square procedure is used in the fitting process. The program iterates until either all convergence criteria are met or ten iterations have elapsed.

The code described here allows for twenty free parameters and a region as large as 240 data channels. This code runs on an LSI-11 computer with 32K memory and disk-storage capability.

INTRODUCTION

The program called GRPANL (GRouP ANaLysis) first developed in 1974, has evolved as an interactive program for use on minicomputers to analyze peak groupings of gamma and x rays in spectra taken with germanium detectors. Its purpose is to calculate peak positions, or energies, and peak intensities while allowing the operator control with respect to the initial parameters for analysis of the peak grouping. Recently, the program has been adapted by others to perform automatic, though prescribed, analyses of spectra.[1,2]

We describe here the algorithm used by GRPANL and provide a Fortran listing of the source code. Although the program was originally written for use on a PDP-8 computer, the version given in the Appendix of this report operates on a PDP-11 under Digital Equipment Corporation's RT-11-V03 operating system. GRPANL, with its subprograms ULOAD2, ULFIT2, and ULOUT2, is capable of analyzing a peak grouping spanning a region of up to 240 channels and allows up to 20 parameters to be freed, i.e. degrees of freedom allowed with respect to any combination of peak heights, positions, and peak shape parameters. The operator may tie any peak position or intensity to that of another designated peak within a region. X-rays, whose shapes differ from gamma rays, can also be fitted. The peak energies are determined either by specifying the exact gain and zero-intercept, or by allowing calculation of the energy scaling, using specified internal reference peaks. For exact energy determinations, the program accounts for system nonlinearity. Finally, GRPANL can calculate gamma emission rates from peak areas by means of efficiency algorithms, which are based on the same model used by the generalized computer program GAMANAL,[3,4] which is a comprehensive program that not only reduces but interprets data.

FITTING TECHNIQUE AND MODEL

Accurate determination of peak positions and areas is one of the most important aspects of analyzing complex x- and gamma-ray spectra obtained with solid-state detectors. These measurements usually require a fitting procedure, particularly when two or more peaks overlap severely. Several methods of fitting have been developed.[4–7] Each fitting technique is based on a supporting model or methodology. The methodology frequently grows out of such factors as experience, intended use of the analysis program, size and speed of the computer used, and accuracy that is required. In this section, we will discuss the methodology used in the GRPANL program.

PEAK SHAPE ALGORITHM

The most easily detected and intense portion of a gamma-ray peak may be described by a Gaussian function, as illustrated in Fig. 1. However, deviations from the Gaussian distribution known as tailing occur on the low-energy and occasionally on the high-energy side of a peak. The low-side tailing is a result of incomplete charge collection in the detector, though low-angle scattering in the sample source or in surrounding materials can also contribute to low-side tailing. Tailing on the high-energy side arises from signal distortion by the pulse electronics modules. This latter effect should be eliminated or reduced as much as possible.

The tailing portion of a peak can generally be described by means of one or more terms of exponential form. Some computer programs divide each peak into regions and apply a separate fitting function to each region.[5] The boundary conditions are chosen so that the calculated response and, in most cases its first derivative, form a smooth curve. The analytical expression for the shape function used by GRPANL is adapted from the work of Sanders and Holmes[8] and treats the entire peak region with a single expression:

$$y_i = y_0\left[\exp \alpha(x_i - x_0)^2 + T(x)\right] \qquad (1)$$

where

y_i = net counts in channel x_i for a single peak
y_0 = peak height
α = $-1/2\ \sigma^2 = -2.7726/(\mathrm{FWHM})^2$, where FWHM is the variable containing the peak-width
x_0 = peak position
$T(x)$ = tailing function.

The tailing function is given by

$$T(x) = \left[A \cdot \exp B(x_i - x_0) + C \cdot \exp D(x_i - x_0)\right] \times \left[1 - \exp 0.4\alpha(x_i - x_0)^2\right]\delta \qquad (2)$$

where

A and C = the short- and long-term tailing amplitudes, as illustrated in Fig. 1.
B and D = the tailing slopes
and $\delta = 1$ for $x_i < x_0$; $\delta = 0$ for $x_i > x_0$.

The expression in the right-hand bracket of the tailing function smoothly reduces the tailing intensity to zero as the peak centroid is approached. The long-term tailing component is disregarded (i.e., C = 0) for many applications. However, inclusion of this component is desirable in order to obtain a more accurate fit of a large-peak multiplet having good counting statistics.

The peak-shape parameters in the above expression are y_0, x_0, α, A, B, and occasionally C and D. Peak positions and heights are generally the measurements of interest, and therefore x_0 and y_0 are the principal parameters in the fitting process. The other parameters may be treated as fixed or as free parameters, depending on whether they have been predetermined by calibration. We have found that the values of α, A, and B can be expressed by simple equations involving the gamma-ray energy.

For example, the peak width σ_T can be expressed as a function of the gamma-ray energy E by

$$\sigma_T^2 = \sigma_n^2 + \sigma_s^2 = k_1 + k_2 E \qquad (3)$$

where σ_n is related to the system "noise," and σ_s is a term related to the statistical nature of the charge collection process. Hence, the measured width of

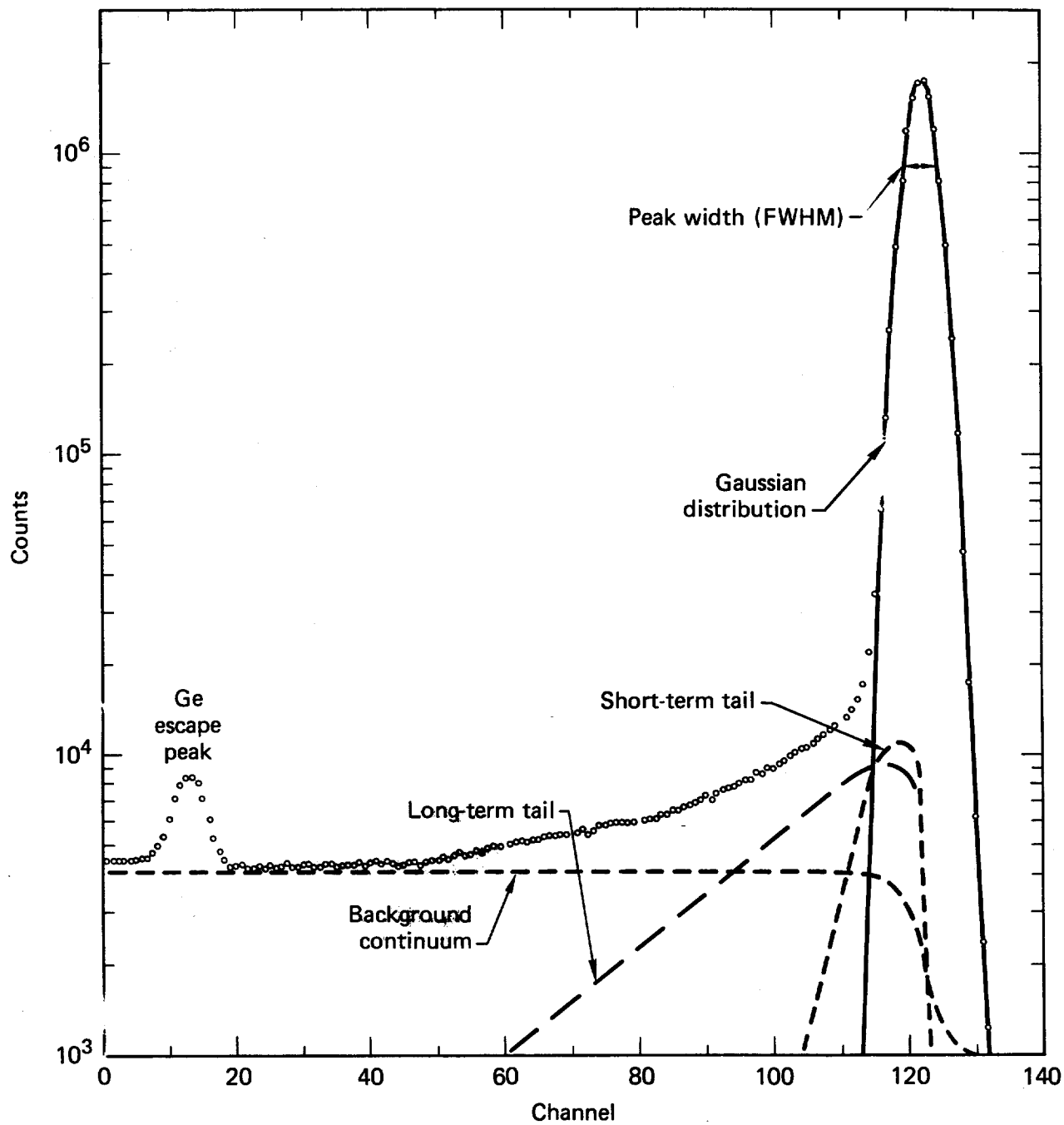

FIG. 1. The principal component of the peak distribution is described by a Gaussian function. The low-side tailing can be described by two additional functions.

two isolated peaks, preferably a low- and a high-energy peak, can be used to determine the values for coefficients k_1 and k_2. The above expression can subsequently be used to calculate the width parameter for a peak of any other energy.

By experience we have found that the tailing slope B can generally be treated as a constant for a given detector system. For an accurate determination of B, a high-energy peak where the tailing is most severe should be used. The peak should be caused by a single gamma ray and have good counting statistics. Once B is determined and held constant, the tailing amplitude A can then be expressed by:

$$\ln A = k_3 + k_4 E \tag{4}$$

The same two peaks that were used to determine the values for k_1 and k_2 determine k_3 and k_4.

There are now five basic peak-shape constants: k_1, k_2, k_3, k_4, and B, plus the two long-term tailing constants, C and D, required only in special cases. GRPANL allows entry of the name of a disk file of peak-shape constants. The operator may still control whether the parameters calculated using these shape constants will be fixed in the fitting process or whether they will only be used as a starting estimate toward improving its value. If a disk file containing shape constants is not constructed the operator must provide initial estimates of the parameters, that is, estimates for the peak width (FWHM), the tailing amplitude A and the tailing slope B.

The peak shapes of x rays differ from those of gamma rays of the same energies because atomic transitions have much shorter lifetimes than nuclear transitions. This difference becomes more pronounced with increasing energy. The intrinsic radiative width of heavy-element x rays at 100 keV is about 100 eV, whereas gamma-ray widths are typically a small fraction of an electron volt. Furthermore, the intrinsic distribution is Lorentzian in shape. When the broadened x-ray distribution is convoluted with the nearly Gaussian instrumental response, the result is a more complex peak shape than that produced by gamma rays.

The modification to the peak shape algorithm necessary to fit an x-ray distribution, illustrated in Fig. 2, is discussed in an earlier report.[9] GRPANL uses this modified algorithm.

DELINEATION OF BACKGROUND

The peaks of any x- or gamma-ray spectrum lie on top of a background distribution resulting from the Compton scattering process. In GRPANL, this distribution must be removed before the peaks can be fitted. This is done by specifying regions on either side of the peak grouping that establish corresponding background levels. The operator may also specify the slopes of the background continuum preceding and following the peak grouping if a peak grouping rests on a sloping background.

An explicit function[4] calculates the interpolated background values under the peak grouping. This function has the following form:

$$B_i = b_n + (b_m - b_n)\left(\frac{\sum_{j=n}^{j=i} y_j}{\sum_{k=n}^{k=m} y_k}\right) \tag{5}$$

where

B_i = computed background at channel i,
y_i = spectrum count of channel i,
b_n = average background level of the low-energy side of the peak,
b_m = average background level of the high-energy side of the peak.

This function produces smoothed steps in the background that occur at positions of greatest intensity. This procedure works equally well for single peaks and for complex peak groupings, an example of which is shown in Fig. 3. An examination of Eq. (5) shows that it is essentially equivalent to the error-function form[6] for describing the background continuum. However, the function in Eq. (5) is used to remove the background before the fitting process commences, whereas the error-function method includes it in the fitting process. The latter method therefore requires more fitting parameters and greater calculational time.

If a background continuum slope has been specified, the steps in the background are superimposed on the specified slope, and the size of the steps is increased or decreased depending on the sign and steepness of the slope. If the two input

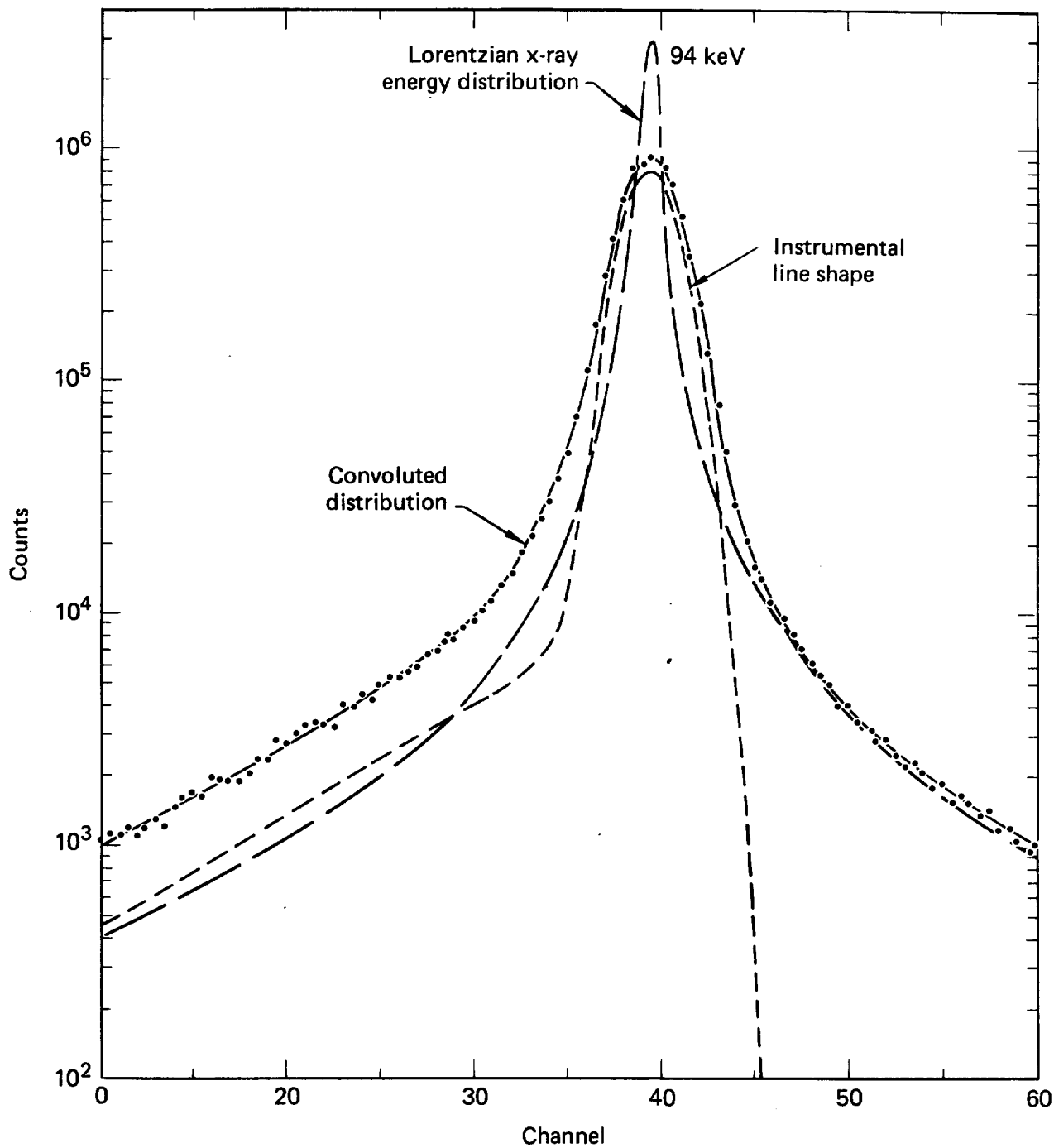

FIG. 2. The intrinsic x-ray energy distribution is Lorentzian in shape. This distribution convolutes with the instrumental line shape resulting in a modified peak shape.

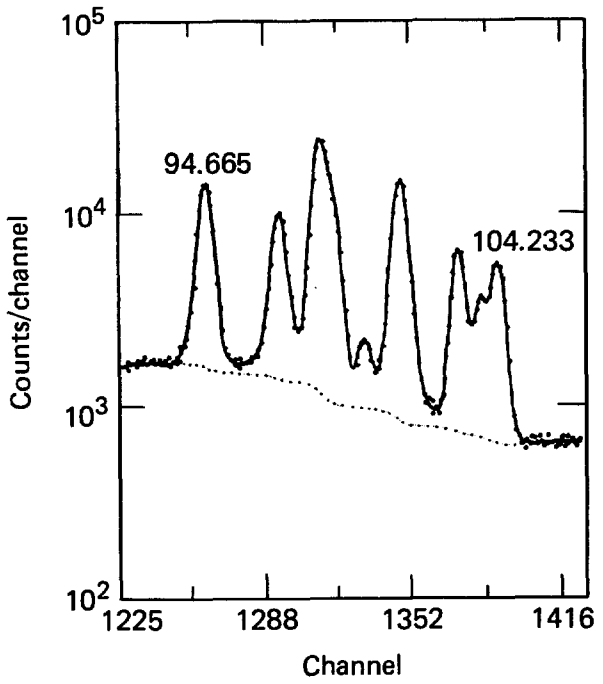

FIG. 3. The background continuum for a complex multiplet as shown here may be accurately calculated by the explicit function given in Eq. (5).

slopes are different, the background continuum makes a gradual change from the one to the other.

The selection of the group boundary position is usually a matter of judgement, and it is frequently difficult to determine or define where a peak should start or end. This is particularly true if one is to include long-term tailing and/or x rays in the fitting process, since these distributions extend to considerable distances from the peak, as previously illustrated in Figs. 1 and 2.

Regardless of where the boundaries are chosen, the regions selected for measuring the average background levels usually contain small contributions from the peaks. GRPANL handles this problem by adjusting the measured levels according to the current value of the contribution of the peaks to the two background regions. The adjustment is refined after each iteration of the least-squares fitting process.

Although long-term tailing may be included in the fitting process, GRPANL does *not* include the area beneath the long-term tailing as part of the reported peak area. However, the area under the short-term tail is included. From experience, we recommend that a short-term tailing slope of B = 1 be used to distinguish the two forms of tailing. The short-term tailing slope should usually have a value between 1.0 and 5.0. The tailing amplitude varies according to the slope value with high amplitude occurring with higher slope values. It also varies according to the magnitude of peak tailing with low energy peaks of good detectors exhibiting rather low values (0.01–0.1) and higher energy peaks exhibiting tailing amplitude values greater than 1.0.

FITTING PROCESS

Following the removal of the background continuum, the net counts in each channel of a peak multiplet are considered as a set of data values Y_i that have been measured with respect to an independent variable x_i and to which several overlapping single peaks may be contributing according to the expression

$$Y_i = \sum_{j=1}^{n} y_{ji} \tag{6}$$

where n is the number of peaks in the grouping.

We now must find a function which, when evaluated point by point, most closely approximates the original data values (Y_i). This evaluation is done by minimizing chi square (X^2). However, because the equations describing the peak shape are not of a linear form, an iterative procedure is used to arrive at a solution to the fit. The method used in GRPANL is a non-linear least-squares technique based on linearizing the equations by performing a first-order, Taylor's series expansion about the trial values of the free parameters. This procedure is also known as the Gauss-Seidel or Newton-Raphson method. From Eqs. (1) and (2), we see that y_i is related to several peak parameters which we here designate simply as p_k.

That is,

$$y_{ij} = f(p_1, p_2 \ldots p_k) \tag{7}$$

By retaining only the first term of the Taylor's expansion of this equation, we derive

$$y_{ij} - f(p_1^\circ, p_2^\circ, \ldots, p_k^\circ) = \sum_{k=1}^{n} \left(\frac{\partial f}{\partial p_k}\right) \delta p_k \tag{8}$$

where

y_{ij} = the contribution of the j-th peak to channel i

p_k° = current best estimate in the k-th peak parameter

Δp_k = the change required in the current best estimate of the k-th parameter to obtain a better fit to the data.

The current "free" parameter values are updated following each iteration, and the process is continued until the current change in each parameter is less than the prescribed amount.

The rate of convergence depends upon a number of factors. Among these are the complexity of the grouping, the number of free parameters, the convergence criteria, and the accuracy of the initial parameter estimates. In addition to slow convergence, poor parameter estimates may also lead to an ill-conditioned matrix or a fitting sequence that does not converge. GRPANL precludes or rectifies these conditions by including boundary constraints, damping or limiting the estimated changes in the free parameters and limiting the allowable number of iterations to 10.

Convergence is determined by comparing the magnitude of the last changes made in each of the parameters being determined with internally specified convergence criteria. Each convergence criterion value is calculated using an equation that contains a fixed and a variable component, the latter being related to the error associated with the parameter being determined. The use of the error component allows early convergence because the specification arising only from the fixed component would be needlessly tight when applied to statistically poor data.

Once convergence has been attained, final calculations are made for peak energies, intensities, and the respective errors. The latter are based on counting statistics and on the goodness-of-fit (the variable QFIT)* and are normally calculated using the formula

$$\mathrm{ERROR}(j) = (\mathrm{DM}(j) * \mathrm{QFIT})^{1/2} \tag{9}$$

where

DM(j) = diagonal error matrix element of parameter j

and

$$\mathrm{QFIT} = \left(\sum_{i=1}^{n} R_i^2\right)/\mathrm{DOF} \tag{10}$$

R_i = the weighted residuals of the fit

and

DOF = degrees of freedom.

With this method, a poor fit in one region affects all the calculated errors, whereas intuitively we know that this is not necessarily true. We have therefore modified Eq. (10) so that a "localized" goodness-of-fit is calculated and applied to each peak in the multiplet. In the new expression, each R_k^2 value is appropriately weighted according to its proximity to the peak in question.

Thus

$$\mathrm{QFIT}(j) = \frac{\sum_{i=1}^{n} R_i^2 \left[\exp(\alpha \cdot \Delta x_i^2)\right]^{1/2}}{\sum_{i=1}^{n} \left[\exp(\alpha \cdot \Delta x_i^2)\right]^{1/2}} \tag{11}$$

where

α = peak width parameter

Δx_i = distance from peak position.

*In this report, computer variable names of one or more letters are used to signify single-function variables.

GAMMA INTENSITY CALCULATIONS

GRPANL also calculates gamma emission rates from the determined peak areas. The instructions are based on the same efficiency model that is used in the GAMANAL program.[3,10] A basic premise of the model is that the overall efficiency can be treated as the product of two components: the intrinsic efficiency of the detector, defined in Eq. (12), and the source–detector geometry. The latter is calculated after determining two hypothetical points in space, one at which the detector is said to reside and another at which all of the sample is said to be concentrated. To determine the efficiency for a particular source–detector geometry, the rule of physics is invoked; that the detection efficiency is inversely proportional to the square of the source-to-detector distance. The fundamental equation used by the model is

$$\epsilon_E \equiv \frac{C_E}{\gamma_E} \cdot d^2 \qquad (12)$$

where

ϵ_E = counting efficiency for gamma ray of energy E, normalized to 1cm
C_E = net counts per unit time in peak
γ_E = source emission rate of gamma energy E
d = source-to-detector distance.

Once a calibration curve (ϵ as a function of E) has been determined, the equation can be transposed to yield photon emission rates, γ_E, for sources of unknown intensity.

A typical configuration for a cylindrical detector and a point source is shown in Fig. 4. To use Eq. (12), the parameter d must be assessed by using measured or calculated values shown in Fig. 4 for:

x = distance from source to detector window
w = distance from window to active detector surface
p = effective penetration of gamma ray into detector
r = effective interaction radius in the detector
r_o = limiting value of r at low energy
d = distance from source to effective zone of interaction
T = detector thickness
R = detector radius.

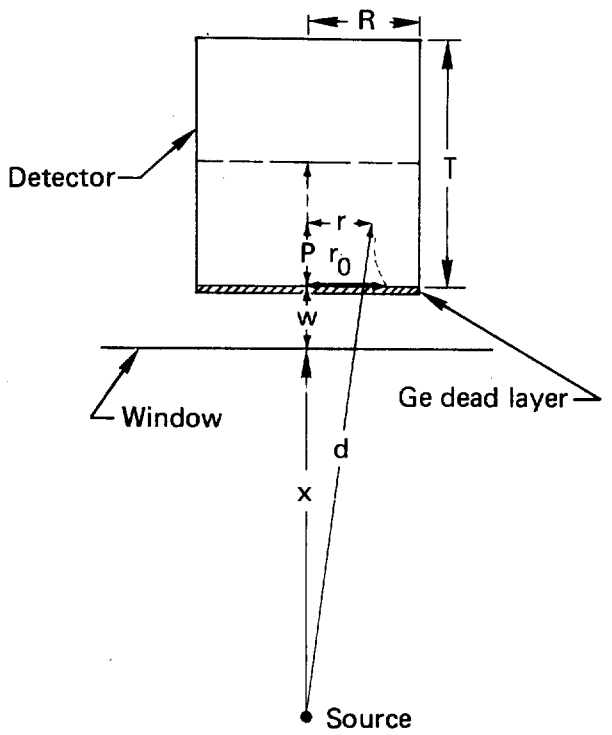

FIG. 4. A point source and a cylindrical detector used to model the effects of geometry on photopeak counting efficiency. Distances shown are defined in the section, *Gamma Intensity Calculations.*

From simple geometry we have

$$d^2 = (x + w + p)^2 + r^2 \ . \qquad (13)$$

Of these, x is the only easily measured parameter. The others must be empirically determined, as discussed in Ref. 10. Our experience indicates that the geometry contribution to the counting efficiency for point sources can be calculated to within ± 1–2% of the correct values for all energies and all usable source-to-detector distances.

The intrinsic detection efficiencies as defined in Eq. (12) are determined by counting absolute standards at relatively large distances and normalizing these efficiencies to a 1-cm distance. These values may be described by a fifth- or sixth-order polynomial equation of the form

$$\ln \epsilon_i = \sum_{j=1}^{n} a_j (\ln E_i)^{(j-1)} \qquad (14)$$

where

ϵ_i = efficiency at the *i*th energy
E_i = gamma energy
a_j = coefficients of the polynomial.

We have found that a single function cannot adequately describe the full range of energies of interest, 0.05–4.0 MeV. Thus, we use two polynomial equations that overlap in the 100–200 keV energy region. After the data are fitted by a least-squares analysis, the resulting curves are examined in the region of overlap, and a crossover energy is chosen that gives the same efficiency by either equation. In this way, the efficiency polynomial can be fitted to the calibration to within ± 1–2%.

Many sources are not point sources. The model also contains formulae to account for the perturbations such as by source extension in two or three dimensions, gamma ray attenuation by the matrix of a volume sample, and by absorbers between the source and the detector. These corrections are also discussed in Ref. 10.

For many prospective users of GRPANL, the inclusion and use of the efficiency model may perhaps be ancillary to the principal purpose of the program, which is to obtain peak positions (energies) and areas. We will therefore not further describe in this report details on the use of the model.

GRPANL OPERATION: AN OVERVIEW

GRPANL is written in FORTRAN IV for use on PDP-11 systems operating with Digital Equipment Corporation's RT-11 V03 or RSTS/E V06B. Because of its size, the program is broken up into GRPANL and three subprograms: ULOAD2, ULFIT2, and ULOUT2. GRPANL begins the execution, and control to the other programs is transferred directly by the subroutine CHAIN, found in the above-named system's subroutine library. The size of common blocks GNL and PASS prevents them from being transferred automatically by the CHAIN subroutine. The transfer is accomplished by each subprogram which writes the contents of the common blocks into the files GNL.DAT and PASS.DAT on the system device before calling CHAIN. After transfer to the next subprogram is completed, the subprogram reads back into core the respective common blocks. The user must make available at least 24 blocks of contiguous disk space on the system device before starting GRPANL for this operation to succeed.

The input data for the program is contained in two disk files: the peak-shape parameter file and the spectral data file. The peak-shape parameter file contains the five basic peak-shape constants, k_1, k_2, k_3, k_4 and B, and the two long-term tailing constants, C and D, which are necessary only in special cases. These constants may be calculated for a detector system by the program PKSHAP, listed in the Appendix. Input to PKSHAP requires the measured widths and tail amplitudes of two isolated peaks in a spectrum taken with the detector system. GRPANL may be used to initially determine these parameters. If a peak-shape parameter file does not exist, GRPANL requests the operator to provide initial estimates of the peak width (stored in variable FWHM), tailing amplitude (EXP1), and tailing slope (EXP2). The new values obtained by GRPANL are then used as input to PKSHAP, provided the value of QFIT, as defined in Eq. (10) is reasonable.

The user enters the shape constant results printed out by PKSHAP into a peak-shape parameter file, a file created and edited by the program FILES, which is also listed in the Appendix. FILES is a general-purpose program for preparing or editing unformatted sequential files. With FILES, the user enters the five shape constants sequentially into the first five locations of the peak-shape parameter file. If the user wishes to use long-term tailing in the fitting process, then the two long-term tailing constants should be entered into locations 7 and 8 of the file. If a correction for system nonlinearity is to be made, the appropriate coefficients are entered into locations 11 through 15. These coefficients are determined by least-squares fitting a power-series polynomial to data where the dependent variable is energy, and the independent variable is the corresponding channel location. The user enters in location 11-15 the coefficients beyond first-order divided by the first-order coefficient, i.e., gain.

The spectral data file contains the pulse-height data which is stored in standard PDP-11 floating-point notation, two words per channel count. The ULOUT2 program, which uses the peak areas to calculate gamma emission rates, requires the livetime (seconds) for the count in the first location of the data file. The second location in this file stores the time of the count, in decimal days. This is for informational purposes only, and it is not necessary for the operation of the code. The spectral data for the region indicated to the GRPANL code is read into core by the subroutine RDSK, which is also listed in the Appendix.

GRPANL reads in the input parameters from the terminal and from the detector parameter disk file, if any. From this information the subprogram calculates the initial values for the peak-shape parameters and sets all the appropriate flags that indicate fixed or free parameters. Program control is transferred next to ULOAD2, which loads the spectral data for the region being fit. The subprogram ULOAD2 determines the net channel data by removing the background calculated by the subroutine BKGRD. From the net data, ULOAD2 determines the initial values for the peak positions and heights. After ULOAD2, ULFIT2 performs an iterative fit to the data. After each iteration, the program prints on the terminal the current peak positions and heights. The current values for the peak shape parameters are also printed if a complete output was initially requested by the user. The program continues its iteration until the current indicated change in each free parameter is less than a prescribed amount, or until 10 iterations have occurred.

When the fitting process is completed, program control is transferred to ULOUT2. The function of the subprogram ULOUT2 is to calculate the peak energies, peak areas, errors on the peak areas, and the value of the reduced chi-square. The program prints this information along with the input information given by the user and the values determined for the peak-shape parameters with their associated errors. If the user chose the option of a complete printout, ULOUT2 lists, channel by channel, the net counts to which the fit was made, the residuals (difference between net count and calculated count), and the standard deviation, as illustrated in Fig. 5, an abbreviated listing of the output.

The ULOUT2 subprogram can also calculate gamma emission rates from the determined peak areas. In the ULOUT2 listing in the Appendix, the FORTRAN statements necessary for this calculation are shown as comment statements. If ULOUT2 is used, a third disk file of detector-parameter data is required. This file is also created using the formatting program FILES. Sixteen entries are needed for the detector-parameter file:

(1-6) The high-energy portion of the detector's intrinsic efficiency curve.

(7-11) The low-energy portion of the detector's intrinsic efficiency curve.

(12) The crossover energy for the above two polynomial equations describing the efficiency vs energy curve for the detector.

(13) The detector depth (cm).

(14) The detector radius (cm).

(15) Surface distance, window to detector (cm).

(16) Germanium dead layer on the front surface of the detectors (g/cm^2).

These sixteen values are read into the array DCNST.

```
 PROGRAM GRPANL LLL VERSION 1          21-APR-80  21:49

FILE NAME  IAEAP1                            PARAMETER FILE

LIVE TIME  4300.0 SECONDS
START DAY     0.000
KEV/CHANNEL  0.110457


GROUP STARTS AT CHANNEL     463
GROUP ENDS AT CHANNEL       530
BACKGROUND LOW ENERGY         5
BACKGROUND HIGH ENERGY        5
ZERO                     107.6716

REFERENCE PK #1 ENERGY = 129.294
REFERENCE PK #2 ENERGY = 208.000

PARAMETER TABLE          FREE
ALPHA                     YES
SHORT TAIL AMPLITUDE      YES
SHORT TAIL SLOPE          NO

LONG TERM TAIL AMPLITUDE  NO
LONG TERM TAIL SLOPE      NO

LOW ENERGY REGION BACKGROUND SLOPE    0.0000
HIGH ENERGY REGION BACKGROUND SLOPE   0.0000


PEAK    ENERGY  WIDTH   UNCERTAIN  FIXED  UNCERTAIN   RATIOED     RATIO
NUMBER   KEV     EV     ENERGY     TO     INTENSITY   TO

    1  159.955   0.00     NO        5        YES        0         0.00000
    2  160.190   0.00     NO        4        NO         4        -0.05170
    3  160.280   0.00     YES       0        YES        0         0.00000
    4  161.450   0.00     YES       0        YES        0         0.00000
    5  164.580   0.00     YES       0        YES        0         0.00000
    6  165.930   0.00     YES       0        YES        0         0.00000


 ANALYSIS RESULTS

QFIT   1.0 FWHM 0.95927+-0.0011 KEV/CHANNEL  0.110457
                                EXP1  0.18617+-0.0701  EXP2  1.99897+-0.0000
                                EXP3  0.00000+-0.0000  EXP4  0.00000+-0.0000


   CHANNEL    ENERGY  COUNTS  GAMMAS/MIN  PCTERR
 1  473.341   159.955    4537. 0.0000E+00   6.6
 2  475.753   160.222      42. 0.0000E+00   0.0
 3  477.598   160.426    2553. 0.0000E+00   4.3
 4  487.160   161.482     826. 0.0000E+00  12.7
 5  515.212   164.580   28900. 0.0000E+00   1.0
 6  524.431   165.599     289. 0.0000E+00  47.8
CHANNEL  ENERGY  INDEX   YNET    RESIDUALS  STD DEV
 463     158.81    1     10.29     -8.28     -0.27
 464     158.92    2     89.26     59.31      1.85
 465     159.03    3    121.23     72.03      2.21
 466     159.14    4    130.21     50.30      1.54
 467     159.25    5    151.18     25.66      0.78
 468     159.37    6    162.15    -25.90     -0.78
 469     159.48    7    234.10    -32.44     -0.95
```

FIG. 5. Full output from GRPANL run.

EXECUTION

To begin execution of GRPANL, the user must have a file of spectral data on disk. An optional file of peak-shape parameters, and/or a detector parameter file may also be constructed prior to execution. A carriage return must follow all input messages. File names have a maximum of six characters, which may be preceded by the device name and a colon if they do not reside on the default device, e.g. *device:file*.

User: R GRPANL

Routine: COMPLETE PRINTOUT?

User: Y selects for inclusion in the output the position, net area, and error for each peak in the fit, and lists the net counts, calculated fit, residual (the difference between the net count and the calculated count), and standard deviation for each channel in the region of interest. An abbreviated sample of the output is shown in Fig. 5.

N deletes channel by channel results from the report.

Routine: SPECTRUM ID IS

User: *device:file*

specifies the file name of the spectral data file.

Routine: GROUP STARTS AT

User: *nnnn* specifies the first channel in the region of interest, for use by subroutine RDSK in filling array Y from spectral data.

Routine: GROUP ENDS AT

User: *nnnn* specifies the last channel in the region of interest. The number of channels between the starting and ending points may not exceed 240.

Routine: # OF BKG. CHANNELS IN FRONT OF GROUP

User: *nnnn* specifies the number of channels preceding the starting channel to be used to calculate the background level on the low-energy side of the region.

Routine: # OF BKG. CHANNELS IN BACK OF GROUP

User: *nnnn* specifies the number of channels following the ending channel to be used in calculating the background level on the high-energy side of the region.

Routine: GAIN (NEG. VALUE FOR ABSOLUTE)
IF = 0, PEAK AREA SIMPLY INTEGRATED

User: *xx.xxxxxx*

(positive value) specifies the "approximate" gain used in locating reference peaks to determine the gain and zero for the spectrum. The energies of two well-defined peaks in

the spectrum are then requested. The two peaks become calibration points for a two-point determination of the gain and zero. If zero is entered for peak No. 1 (PK #1), GRPANL assumes zero energy at channel zero.

0 determines only the total net counts in the region. A zero entry signifies that a fit to the peaks in the designated region is not desired.

-xx.xxxxxx

(negative value) specifies "absolute gain," i.e., no internal peaks are used for energy reference, and the final energy values are based on this gain.

Routine: ENERGY OF ZERO CHANNEL

User: *xxx.xxxx*

specifies the energy of channel zero, information useful for those spectra which have a large zero offset.

Routine: LIST PK ENERGIES IN THIS GROUP (NEG. VALUE FOR X-RAYS)
TERMINATE WITH A ZERO

User: *xxxxx.xxxx*

specifies the energy for a peak in the region. Additional peak energies may be entered, each followed by a carriage return. Terminate the list of energies with a zero entry.

-xxxxx.xxxx

(negative value) signifies that the peak is an x ray. For negative values, the intrinsic width of the x-ray line of the form *xxxx.x* is then entered after the prompt.

INTRINSIC WIDTH (EV)
IF ENTRY = 0, AN INTERNALLY ESTIMATED VALUE USED

The values to be entered are listed in Ref. 11. If zero is entered, an internal algorithm calculates an approximate width.

0 terminates the entry of peak energies.

Routine: PEAKS ENTERED FOR THIS REGION

(displays the number of peaks entered)

TYPE SEQUENCE # OF PEAKS WHOSE ENERGY ARE UNCERTAIN
TYPE 99 FOR ALL—
NEG. FOR FIXED RELATIONSHIP. END WITH 0.

User: 99 designates all peaks in the list as free in the fitting process.

nnn (sequence number) designates the centroid position of the peak as variable in the fitting process. Those not designated in this step are fixed in position to the channel corresponding to the energy of the peak entered, by the formula

peak position = (peak energy - zero channel energy) / gain

-nnn (negative sequence number) fixes the peak energy relative to another peak in the region. The sequence number of this other peak in the form nnn is entered in response to the prompt.

ENERGY FIXED RELATIVE TO PEAK #:

The centroid of the current peak is then varied as the centroid of the peak to which it has been fixed. This option is used for splitting close multiplets or for locating weak peaks.

0 terminates the entry of sequence numbers of peaks of uncertain energy.

Routine: SEQUENCE # OF PEAK WHOSE INTENSITY IS KNOWN

User: *nnn* (sequence number) indicates that the intensity of this peak is known. A zero entry indicates that all are unknown. The peak height is entered after the prompt.

PEAK HEIGHT (NEG. FOR RATIO) =

xxxxxxx.xxxx

designates the peak height for the sequence-numbered peak named above.

–xxxxxxx.xxxx

(negative entry) specifies that the peak height is to be ratioed to another peak in the region, a procedure useful when both peaks are from the same isotope and their branching ratios are well known. The absolute value of this entry is the ratio of the current peak height to the height of the reference peak.

Enter the sequence number of the reference peak in the form nnn following the prompt RATIOED TO PEAK #: .

Routine: IS THERE A PARAMETER FILE?

User: Y reads in a file of peak-shape parameters available on disk. Enter the file name in the form *device:file* after the prompt FILE NAME?

After the file name is entered, the routine asks

DO YOU WANT TO FREE ANY SHAPE PARAMETERS?

Y presents each shape parameter for a further Y or N decision.

Y — frees this shape parameter. The maximum number of free parameters, including peak positions and heights, that are allowed for fitting each peak grouping is 20.

N — fixes the parameter to the value determined from the parameter file, in accordance with Eqs. (3) and (4), above.

N — indicates there is no parameter file. If there is none, enter a value for each peak-shape parameter, as listed below. For each, a response to the prompt TO BE FREED? is required.

Y — frees the parameter.

N — fixes the parameter to the value entered.

The peak-shape parameters that can be freed for the entire peak grouping are:

FWHM (IN KEV)
SHORT-TERM TAIL AMPLITUDE
SHORT-TERM TAILING SLOPE
LONG-TERM TAIL AMPLITUDE
LONG-TERM TAIL SLOPE

The height and the position may be freed for each peak in the multiplet. A 20-peak multiplet may be fit if peak height is the only free parameter. A seven-peak multiplet can be fit if all parameters are free.

Routine: LOW ENERGY BKGRD SLOPE EXPRESSED IN %/CHAN =

User: *x.xxx* — specifies the value for the slope before the region of interest, expressed as % change in counts per channel relative to the average counts per channel in the region. This option is useful for peaks which lie in a region where the Compton continuum is not flat, but has a nonzero slope.

Routine: HIGH ENERGY BKGRD SLOPE EXPRESSED IN %/CHAN =

User: *x.xxx* — specifies a similar value for the slope following the region of interest.

REFERENCES

1. J. E. Fager and F. P. Brauer, *Rapid Nondestructive Plutonium Isotopic Analysis*, U.S. National Bureau of Standards Special Publication 528, 64 (1978).
2. Currently used at Mound Laboratories, Miamisburg, Ohio.
3. R. Gunnink and J. B. Niday, *Computerized Quantitative Analysis by Gamma-Ray Spectrometer*, Lawrence Livermore National Laboratory, Livermore, Calif., UCRL-57061 (1971).
4. R. Gunnink, *Computer Techniques for Analysis of Gamma-Ray Spectra*, Lawrence Livermore National Laboratory, Livermore, Calif., UCRL-80297 (1978).
5. J. J. Routti and S. G. Prussin, "Photopeak Method for the Computer Analysis of Gamma-Ray Spectra from Semiconductor Detectors," *Nucl. Inst. and Methods* **72**, 125 (1969).
6. G. Phillips and K. W. Marlow, " Automatic Analysis of Gamma-Ray Spectra from Germanium Detectors," *Nucl. Inst. and Methods* **137**, 525 (1976).
7. H. P. Block et al, "A New Peak Search Method for an Automatic Spectrum Analysis System," *Nucl. Inst. and Methods* **128**, 545 (1975).
8. W. M. Sanders and D. M. Holmes, *An Analytical Method for Unfolding Gamma-Ray Spectra*, Los Alamos Scientific Laboratory, Los Alamos, N.M., LA-4038 (1969).
9. R. Gunnink, "An Algorithm for Fitting Lorentzian-Broadened K-Series X-Ray Peaks of the Heavy Elements," *Nucl. Inst. and Methods* **143**, 145 (1977).
10. R. Gunnink and J. B. Niday, in: *ERDA Symposium on X- and Gamma-Ray Sources and Applications* (U. S. Energy Research and Development Administration, CONF-760539, Washington, D.C., GPO, 1976) p. 55.
11. S. I. Salom and P. L. Lee, "Experimental Widths of K and L X-Ray Lines," *Atomic Data and Nuclear Data Tables* **18**(3) 233 (1976).

BED/cb

APPENDIX: SOURCE LISTINGS

GRPANL
ULOAD2
ULFIT2
ULOUT2
FILES
PKSHAP

Subroutines:

RDSK
YESNO
MAXVAL
MINVAL
NCNTS
NCTS
AVE
BKGRD
GFIT
BWF
MLR
MLRA
MLRE
GPM
CALMU
CALGPM

```
      C       PROGRAM GRPANL
      C       LINK WITH RDSK,YESNO,AVE,MAXVAL,GFIT,NCNTS,NCTS
      C
      C       THIS PROGRAM IS A GENERAL PURPOSE PROGRAM FOR FITTING
      C       MULTIPLETS (GAMMA-RAYS AND X-RAYS).
      C       IT WILL ACCOMMODATE UP TO 20 FREE PARAMETERS.  THE PEAK
      C       HEIGHTS (AREAS) ARE AUTOMATICALLY FREED.  THE OTHERS
      C       CAN BE SELECTED.  UP TO 240 DATA POINTS ARE ALLOWED.
      C
      C       THIS PROGRAM READS IN GRPANL PARAMETERS
      C       FROM TTY TERMINAL
      C
0001          INTEGER*2 ULOAD2(4)
0002          LOGICAL YESNO
0003          DIMENSION AMATRX(5888),BMATRX(256)
0004          COMMON/GNL/NUMS,NUME,NDPTS,NP,NWRDS(6),NFLG(20),NYFLG(20),
             1  NXFLG(20),FNAME(3),SP(2),GAIN,ZERO,START,ALPHA,FWHM,ATAIL,
             2  BTAIL,EXP3,EXP4,STD(2),C1,C2,AVEBG,SLP(2),EN(20),SHAPC(10),
             3  CLIN(5),HIGHT(20),GAMA(20),Y(240),DUM(4500)
0005          COMMON/PASS/ENJ(20),IST,IEND,NUMSJ,NUMEJ,GAINJ,ZEROJ,ST1J,
             1  ST2J,STRJ(20),NENUM,ALPJ,STAMP,STSLP,TLAMP,TLSLP,BKGSLP(2),
             2  CFILE(3),SSS,YES,BK1,NGAIN,IREGG,LISTFG,PAME(3)
      C
0006          EQUIVALENCE (NUMS,AMATRX(1)),(ENJ(1),BMATRX(1))
      C       LISTFG = LISTING FLAG = 0 COMPLETE PRINTOUT
      C       OF CALCULATIONS
      C      = 1 SHORT PRINTOUT
      C       THE FOLLOWING STATEMENT PROVIDES FOR CHAINING TO THE PROGRAM
      C       ULOAD2 ON LOGICAL DEVICE RK3.  IT MUST BE CHANGED ACCORDINGLY
      C       IF THE PROGRAM RESIDES ELSEWHERE.
      C
0007          DATA ULOAD2/3RRK3,3RULO,3RAD2,3RSAV/
      C
0008          LISTFG = 1
0009          IALGTH = 5888
0010          IBLGTH = 256
0011          TYPE 99
0012  99      FORMAT(' COMPLETE PRINTOUT? ',$)
0013          IF(YESNO())LISTFG = 0
0015  1       TYPE 100
0016  100     FORMAT(' SPECTRUM ID IS ',$)
0017          ACCEPT 101,FNAME
0018  101     FORMAT(3A4)
0019          TYPE 102
0020  102     FORMAT(' GROUP STARTS AT ',$)
0021          ACCEPT 106,IST
0022          TYPE 104
0023  104     FORMAT(' GROUP ENDS AT ',$)
0024          ACCEPT 106,IEND
0025          TYPE 105
0026  105     FORMAT(' # OF BKG. CHANNELS IN FRONT OF GROUP = ',$)
0027          ACCEPT 106,NUMS
0028          NUMSJ = NUMS
0029  106     FORMAT(I4)
0030          SUM(J) = 0.
0031          AMP1 = EXP1
0032          AMP2 = EXP3
0033          GAMMA = 0.
0034          XI = PKPOS(J)
0035          DO 300 L = 1,2
0036          CC = CX * XI
0037          IF(GAMA(J))201,201,100
0038  100     CALL BWF(CC)
0039  201     BG(L)=BG(L)+PKHT(J)*(.6366*GAMA(J)*CX*CC+AMP1*EXP(-EXP2*XI)
             1  +AMP2*EXP(-EXP4*XI))
```

```
0040          AMP1 = 0.
0041          AMP2 = 0.
0042          XI = XPTS - PKPOS(J)
0043   300    CONTINUE
0044   400    CONTINUE
0045          YP = 0.
0046          DBG = BG(2) - BG(1)
0047          CALL AVE(1,NDPTS,YSUM,AV,YNET)
       C      ZERO COEFFICIENT ARRAY
0048          DO 2 J=1,20
0049          DO 2 I=1,NDPTS
0050   2      ACOEF(I,J)=0.
       C      FORM MATRIX OF LINEAR EQUATIONS
0051          DO 30 I = 1,NDPTS
0052          XI = I
0053          FX = 0.0
0054          FXY=0.0
0055          DALFA = 0.0
0056          DBETA = 0.0
0057          DGAMA = 0.0
0058          DDELT = 0.0
0059          DCHI = 0.0
       C      DETERMINE WEIGHTING FACTOR
0060          WT = 1.0 / SQRT (YNET(I) + AVEBG)
0061          K = KK
       C      CALCULATE COEFFICIENTS FOR EACH OF 'NP' PEAKS
0062          DO 20 J = 1,NP
0063          DLTAX = XI - PKPOS(J)
0064          DX = DLTAX
0065          DLX = DLTAX
0066          DLXSQ = DLTAX * DLTAX
0067          EXPN1 = 0.0
0068          PHT = PKHT(J)
0069          ALFA = -2.7726 / (ASLP * PKPOS(J) + SG)
0070          EXPNT = ALFA * DLXSQ
0071          EXPN3 = 0.
0072          IF (EXPNT + 14.) 4,3,3
       C      CALCULATE GAUSSIAN COMPONENT OF PEAK
0073   3      EXPN1 = EXP (EXPNT)
0074   4      IF(DLTAX)53,54,54
0075   53     EXPN3 = EXP (0.4 * EXPNT)
0076   54     FXX = EXPN1
0077          IF(GAMA(J))52,52,51
       C      COMPUTE MODIFIED SHAPE FOR X-RAY PEAK
0078   51     WL = GAMA(J) * CX
0079          CX = SQRT(-ALFA)
0080          CD = CX * DLTAX
0081          C1 = 1. - .5642 * WL
0082          C2 = .15915 * WL
0083          C3 = .25 * WL * WL
0084          C4 = .6366 * WL
0085          RT = C1 + C3
0086          CC = ABS(CD)
0087          CALL BWF(CC)
0088          AA = (C1 - C2 *EXPNT +C3 *(1. + 2. *EXPNT))*EXPN1 +C4 * CC
0089          CD= AA/RT
0090          FXX = CD
0091   52     IF (NFLG1) 6,6,5
       C      PEAK WIDTH PARAMETER FREE
0092   5      DALFA = DALFA + PHT * DLXSQ * EXPN1
0093   6      IF (DLTAX) 7,14,14
0094   7      EXPN = EXP2 * DLTAX
0095          IF (EXPN + 10.) 12,8,8
0096   8      EXPN2 = EXP (EXPN)
0097          CB = EXPN2 * (1.0 - EXPN3)
```

```
0098          FXX = FXX + EXP1 * CB
0099          IF (NFLG3) 10,10,9
      C       SHORT TERM TAILING SLOPE FREE
0100  9       DGAMA = DGAMA + DLTAX * EXP1 * PHT * CB
0101  10      IF (NFLG2) 12,12,11
      C       SHORT TERM TAILING AMPLITUDE FREE
0102  11      DBETA = DBETA + CB * PHT
0103  12      EXPN = EXP4 * DLTAX
0104          IF(EXPN + 10.)14,18,18
0105  18      CB = EXP(EXPN) * (1. - EXPN3)
0106          CY = PHT * EXP3
0107          FXY = FXY + CY * CB
0108          IF(NFLG4)28,28,27
      C       LONG TERM TAILING AMPLITUDE FREE
0109  27      DDELT = DDELT + CB * PHT
0110  28      IF(NFLG5)14,14,29
      C       LONG TERM TAILING SLOPE FREE
0111  29      DCHI = DCHI + DLTAX * CY * CB
0112  14      IF(NYFLG(J))161,16,15
0113  161     IF(HIGHT(J))162,16,16
0114  162     LPK = -NYFLG(J)-1
0115          L = KK
0116          DO 166 M=1,LPK
0117          IF(NYFLG(M))164,164,163
0118  163     L= L+1
0119  164     IF(NXFLG(M))166,166,165
0120  165     L=L+1
0121  166     CONTINUE
0122          L= L+1
0123          VALU=-HIGHT(J)*EXPN1*WT
0124          GOTO 114
      C       PEAK HEIGHT FREE
0125  15      K = K + 1
0126          L=K
0127          VALU=EXPN1*WT
0128  114     ACOEF(I,L) =VALU + ACOEF(I,L)
0129  16      IF (NXFLG(J)) 19,19,17
      C       PEAK POSITION FREE
0130  17      K = K + 1
0131          ACOEF(I,K) = -2. *ALFA *PHT * DX *EXPN1 * WT
0132  19      FX = FX +FXY + FXX * PHT
0133          SUM(J) = SUM(J) + FXX * PHT
0134  20      CONTINUE
0135          L = 0
0136          IF (NFLG1) 22,22,21
0137  21      L = L + 1
0138          ACOEF(I,L) = DALFA * WT
0139  22      IF (NFLG2) 24,24,23
0140  23      L = L + 1
0141          ACOEF(I,L) = DBETA * WT
0142  24      IF (NFLG3) 38,38,25
0143  25      L = L + 1
0144          ACOEF(I,L) = DGAMA * WT
0145  38      IF(NFLG4)37,37,36
0146  36      L = L + 1
0147          ACOEF(I,L) = DDELT * WT
0148  37      IF(NFLG5)39,39,26
0149  26      L = L + 1
0150          ACOEF(I,L) = DCHI * WT
0151  39      YP = YP + YNET(I)
0152          DELY(I) = (YNET(I) + BG(1) +YP * DBG/YSUM -FX) * WT
0153  30      CONTINUE
0154          LFLG = 1
0155          CALL ASSIGN(1,'RK0:AMTRX.DAT',0,'NEW','CC')
0156          DO 101 J = 1,K
```

```
0157          WRITE(1) (ACOEF(I,J), I = 1,240)
0158   101    CONTINUE
0159          CALL CLOSE(1)
       C      PERFORM LEAST-SQUARES CALCULATION
0160          CALL MLR(240,NDPTS,K,DELY,DEL,ACOEF,RM,WM,VM,LFLG)
0161          IF(LFLG.EQ.2)GOTO 102
0163          CALL MLRA(240,NDPTS,K,DELY,DEL,ACOEF,RM,WM,VM,LFLG)
0164   102    CONTINUE
0165          FWHM= GAIN * SQRT(-2.7726/ALPHA)
0166          EX2 = EXP2/GAIN
0167          EX4 = EXP4/GAIN
0168          TYPE 9040
0169   9040   FORMAT(/,' ITERATION  FWHM    EXP1    EXP2    EXP3    EXP4')
0170          TYPE 9030,ITER,FWHM,EXP1,EX2,EXP3,EX4
0171   9030   FORMAT(I3,6X,F9.5,4F8.4)
0172          IF(LISTFG.EQ.1) GOTO 301
0174          TYPE 9010
0175   9010   FORMAT(/,'  CHANNEL  KEV     PEAK HEIGHT  ')
0176          DO 200 J=1,NP
0177          PKP=PKPOS(J)+START
0178          PKE =(PKP - REFCH)*GAIN + REFEN
0179          TYPE 9020,PKP,PKE,PKHT(J)
0180   200    CONTINUE
0181   9020   FORMAT(2F9.2,F14.3)
0182          GOTO(301,302,303)LFLG
0183   302    TYPE 305
0184   305    FORMAT(' MATRIX IS SINGULAR')
0185          STOP
0186   303    TYPE 306
0187   306    FORMAT(' SUM OF SQUARES OF R > THAN THOSE OF Y')
0188   310    STOP ' RETRN2 310'
0189   301    CALL MLRE(DM,240,K,ACOEF,WM)
0190          DO 304 J =1,K
0191   304    DM(J) = SQRT(DM(J))
0192          DO 307 J = 1,7
0193   307    ERR(J) = 0.
       C      INCREMENT THE TRIAL VALUES AND CHECK FOR COMPLETION OF ITERATIVE
       C      PROCESS.
0194   1000   L = 0
0195          LFLG = 0
0196          IF (NFLG1) 43,43,41
       C      INCREMENT PEAK WIDTH PARAMETER
0197   41     L = L + 1
0198          ERR(3) = DM(L)
0199          ADEL = ABS(DEL(L))
0200          IF(ADEL + .01 * ALPHA)430,430,42
0201   42     ALPHA = ALPHA + DEL(L) * (1. +0.8* ADEL/(ALPHA - ADEL))
0202          LFLG = 1
0203          GOTO 43
0204   430    ALPHA = ALPHA +DEL(L)
0205   43     IF (NFLG2) 46,46,44
0206   44     L = L + 1
0207          ERR(4) = DM(L)
0208          IF(ITER -NFLG1)45,45,115
       C      INCREMENT SHORT TERM TAILING AMPLITUDE
0209   115    IF (EXP1 + DEL(L)) 116,116,117
0210   116    DEL(L) =- .5 * EXP1
0211   117    ADEL = ABS(DEL(L))
0212          IF(ADEL - DM(L) - 0.1*EXP1)460,460,45
0213   45     EXP1 = EXP1 + DEL(L) * (1. -0.8* ADEL / (EXP1 + ADEL))
0214          LFLG = 1
0215          GOTO 47
0216   460    EXP1 = EXP1 + DEL(L)
0217   46     IF (NFLG3) 49,49,47
0218   47     L = L + 1
```

```
0219          ERR(5) = DM(L)
0220          IF(ITER-NFLG1-NFLG2)480,480,119
      C       INCREMENT SHORT TERM TAILING SLOPE
0221  119     IF (EXP2 + DEL(L)) 120,120,118
0222  120     DEL(L) =- .5 * EXP2
0223  118     ADEL = ABS(DEL(L))
0224          IF(ADEL - DM(L) - .02*EXP2)490,490,48
0225  48      EXP2 = EXP2 + DEL(L) * (1. -0.8* ADEL / (EXP2 + ADEL))
0226  480     LFLG = 1
0227          GOTO 49
0228  490     EXP2 = EXP2 + DEL(L)
0229  49      IF(NFLG4)126,126,121
      C       INCREMENT LONG TERM TAILING AMPLITUDE
0230  121     L = L + 1
0231          ERR(6) = DM(L)
0232          ADEL=ABS(DEL(L))
0233          IF(ADEL - DM(L) - .01*EXP3)1260,1260,125
0234  125     EXP3 = EXP3 +DEL(L) * ABS(1. -0.8* ADEL/(EXP3 + ADEL))
0235          LFLG = 1
0236          GOTO 126
0237  1260    EXP3 = EXP3 + DEL(L)
0238  126     IF(NFLG5)132,132,127
      C       INCREMENT LONG TERM TAILING SLOPE
0239  127     L = L + 1
0240          ERR(7) = DM(L)
0241          ADEL=ABS(DEL(L))
0242          IF(ADEL - DM(L) - .02*EXP4)131,131,130
0243  130     EXP4 = EXP4 +DEL(L)*ABS(1. -0.8*ADEL/(EXP4+ADEL))
0244          LFLG = 1
0245          GOTO 132
0246  131     EXP4 = EXP4 + DEL(L)
0247  132     K = L
0248          DO 60 J = 1,NP
0249          IF (NYFLG(J)) 152,152,150
      C       INCREMENT PEAK HEIGHTS
0250  150     K = K + 1
0251          ADEL = ABS(DEL(K))
0252          PKHT(J) = PKHT(J) + DEL(K) * (1. - ADEL /(PKHT(J) + ADEL))
0253          IF (ADEL - DM(K) - .001 * PKHT(J)) 152,152,151
0254  151     LFLG = 1
0255  152     IF (NXFLG(J)) 60,60,153
      C       INCREMENT PEAK POSITIONS
0256  153     K = K + 1
0257          ADEL = ABS(DEL(K))
0258          IF(ADEL - 1.)56,55,55
0259  55      DEL(K) = DEL(K)/ADEL
0260  56      PKPOS(J) = PKPOS(J) + DEL(K)
0261          IF(ADEL - DM(K) - .01 * PKPOS(J))60,60,154
0262  154     LFLG = 1
0263  60      CONTINUE
      C       REESTABLISH PEAK ENERGY AND INTENSITY RELATIONSHIPS
      C       SPECIFIED IN INPUT.
0264          DO 65 J = 1,NP
0265          IF(NYFLG(J))61,63,63
0266  61      PKHT(J) = HIGHT(J)
0267          IF(HIGHT(J))62,63,63
0268  62      L = -NYFLG(J)
0269          PKHT(J) = -HIGHT(J) * PKHT(L)
0270  63      IF(NXFLG(J))64,65,65
0271  64      L = -NXFLG(J)
0272          PKPOS(J) = PKPOS(L) -(EN(L) - EN(J))/GAIN
0273  65      CONTINUE
0274          IF(LFLG) 70,79,70
0275  70      ITER = ITER + 1
0276          IF (ITER - 10)78,78,77
```

```
C       ITERATE CALCULATION AGAIN
78      DO 90 I = 1,5
        IF(NFLG(I))91,90,90
91      NFLG(I) = - NFLG(I)
90      CONTINUE
        GOTO 1
77      TYPE 105
105     FORMAT ('MAXIMUM ITERATIONS EXCEEDED.')
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       TERMINATE ITERATIONS. OUTPUT RESULTS.
79      CALL ASSIGN(NDEV,'SY:GNL.DAT')
        WRITE (NDEV) (AMATRX(I), I = 1,IALGTH)
        CALL CLOSE(NDEV)
        CALL ASSIGN(NDEV,'SY:PASS.DAT')
        WRITE (NDEV) (BMATRX(I), I = 1,IBLGTH)
        CALL CLOSE(NDEV)
        CALL CHAIN(ULOUT2,0,0)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        END
```

```
C      PROGRAM ULOAD2
C      THIS SUBPROGRAM IS THE SECOND PART TO THE PROGRAM "GRPANL"
C      LINK WITH RDSK,MAXVAL,MINVAL,AVE,BKGRD,GFIT
C
       INTEGER*2 ULFIT2(4)
       DIMENSION AMATRX(5888),BMATRX(256),XC(2),REFCH(2)
       COMMON/GNL/ NUMS,NUME,NDPTS,NP,ITER,NWRDS(5),NFLG(20),
     1  NYFLG(20),NXFLG(20),FNAME(3),SP(2),
     2  GAIN,ZERO,START,ALPHA,FWHM,ATAIL,BTAIL,EXP3,EXP4,
     3  STD(2),C1,C2,AVEBG,SLP(2),EN(20),SHAPC(10),CLIN(5),
     4  HIGHT(20),GAMA(20),Y(240),PKPOS(20),PKHT(20),DUM(4460)
       COMMON/PASS/ENJ(20),IST,IEND,NUMSJ,NUMEJ,GAINJ,ZEROJ,ST1J,ST2J,
     1  STRJ(20),NENUM,ALPJ,STAMP,STSLP,TLAMP,TLSLP,BKGSLP(2),FILE(3),
     2  SSS,YES,BK1,NGAIN,IREGG,LISTFG,PAME(3)
       EQUIVALENCE (NUMS,AMATRX(1)),(ENJ(1),BMATRX(1))
C      THIS PROGRAM CHAINS TO THE PROGRAM ULFIT2 WHICH RESIDES
C      ON PHYSICAL DEVICE RK3.  THIS DATA STATEMENT MUST BE
C      CHANGED ACCORDINGLY IF THE PROGRAM RESIDES ELSEWHERE.
       DATA ULFIT2/3RRK3,3RULF,3RIT2,3RSAV/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      THE FOLLOWING STATEMENTS READ THE COMMON BLOCKS BACK INTO CORE
       NDEV = 1
       CALL CLOSE(NDEV)
       CALL ASSIGN(NDEV,'SY:GNL.DAT')
       IALGTH = 5888
       READ (NDEV) (AMATRX(I), I = 1,IALGTH)
       CALL CLOSE(NDEV)
       IBLGTH = 256
       CALL ASSIGN(NDEV,'SY:PASS.DAT')
       READ (NDEV) (BMATRX(I), I = 1,IBLGTH)
       CALL CLOSE(NDEV)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      A NEGATIVE NGAIN IS A FLAG TO INDICATE THAT INTERNAL
C      REFERENCE PEAKS ARE TO BE USED TO ESTABLISH ENERGY SCALING.
C      TWO PEAKS ARE USED.  IF FIRST PEAK ENERGY=0, INTERPRETED
C      TO MEAN CHANNEL ZERO =0 KEV.
       IF(NGAIN) 29,29,54
 29    DO 50 J=1,2
       REFCH(J)=0.
       XC(J)=0.
       IF(STD(J)) 32,50,32
 32    REF=(STD(J)- ZERO)/GAIN
       ST1=REF-30.
       NX = ST1
       ST1 = FLOAT(NX)
       IST1 = ST1 + 1.
       CALL RDSK(IST1,60,FNAME,Y)
       CALL MAXVAL (20,40,JPK,YMAX,Y)
       CALL MINVAL (1,JPK,MS,BG1,Y)
       CALL MINVAL (JPK,60,ME,BG2,Y)
       CALL AVE (MS,ME,SUM,AV,Y)
       YS = 0.0
       DBG = BG2 - BG1
       DO 35 I = MS,ME
       YS = YS + Y(I)
       Y(I) = Y(I) - BG1 - DBG * YS/SUM
 35    CONTINUE
       IF(SHAPC(1).EQ.0)GOTO 33
       GSQ = GAIN * GAIN
       SG = SHAPC(1)/GSQ + SHAPC(2) *(ST1/GAIN +ZERO/GSQ)+.462
       ALFA=-2.7726/SG
       EXP1=EXP(SHAPC(3) +STD(J) *SHAPC(4))
       EXP2=BTAIL
       GOTO 34
C      NO PARAMETER FILE; FOLLOWING VALUES ARE ONLY APPROXIMATE
```

```
0046  33    ALFA=-2.7726*GSQ/FWHM**2
0047        EXP1=ATAIL
0048        EXP2=BTAIL
0049  34    NUM = 9
0050        IF (ALFA +.2) 36,36,37
0051  36    NUM = 7
      C     DETERMINE PEAK POSITIONS OF REFERENCE PEAKS
0052  37    CALL GFIT (JPK,PKPO,PHT,NUM,ALFA,EXP1,EXP2,Y)
0053        REFCH(J) = ST1 + PKPO - 1.
0054        PP = REFCH(J)
0055        PKP = PP
0056        DO 38 I = 1,4
0057        PKP = PKP * REFCH(J)
0058  38    PP = PP + CLIN(I) * PKP
0059        XC(J) = PP
0060  50    CONTINUE
0061        GAIN = (STD(2) - STD(1)) / (XC(2) - XC(1))
0062        GAINJ =GAIN
0063        ZERO = STD(1) - GAIN * XC(1)
0064        ZEROJ = ZERO
      C     CALCULATE AFFECT OF SYSTEM NONLINEARITY ON ENERGY MEASUREMENT.
      C     CLIN(I) VALUES =0 FOR LINEAR RESPONSE.
0065  54    AVECH= 0.5 * (IST+IEND)
0066        PP = AVECH
0067        PKP = PP
0068        DO 52 I = 1,4
0069        PKP = PKP * AVECH
0070  52    PP = PP + CLIN(I) * PKP
0071        STD(1) = ZERO + GAIN * PP
0072        STD(2) = AVECH
0073        REFEN = STD(1)
0074        REFCHA = STD(2)
      C     DETERMINE PEAK GROUPING LOCATION AND READ IN DATA
0075        NDPTS = IEND - IST + 1
0076        NDATA = NDPTS + NUMS + NUME
0077        IST2= IST-NUMS+1
0078        CALL RDSK(IST2,NDATA,FNAME,Y)
0079        START = IST -1
0080        XN = 0.
0081  40    NE1 = NDPTS + NUMS
      C     DETERMINE AVERAGE BACKGROUND LEVEL IN FRONT AND BACK OF
      C     PEAK GROUPING
0082        CALL AVE (1,NUMS,BGD,AVBG1,Y)
0083        CALL AVE (NE1,NDATA,YSUM,AVBG2,Y)
0084        BGD = BGD - YSUM
0085        K = NUMS + 1
0086        KK = NE1 - 1
0087        CALL AVE (K,KK,YSUM,DD,Y)
      C     REMOVE BACKGROUND CONTINUUM FROM PEAK GROUPING.  PRESERVE
      C     ENTERING AND EXITING BACKGROUND SLOPES.
0088        SLP(1) = SLP(1) * .01 * AVBG1
0089        SLP(2)=  SLP(2) * .01 * AVBG2
0090        CALL BKGRD (NUMS,NE1,AVBG1,AVBG2,SLP(1),SLP(2),YSUM,NDPTS,Y,Y)
      C     TEST TO SEE IF TOO MANY CHANNELS
0091        IF (NDPTS - 240) 44,44,43
0092  43    CONTINUE
0093        TYPE 100,NDPTS
0094  100   FORMAT ('TOO MANY CHANNELS.  NDPTS = ',I4)
0095        TYPE 101
0096  101   FORMAT(' NEW STARTING CHANNEL = ',$)
0097        ACCEPT 102,IST
0098  102   FORMAT(I4)
0099        IF(IST)47,46,47
0100  46    IST = START + 1.
0101  47    ST1 = IST
```

```
      GSQ = GAIN * GAIN
      SG = SHAPC(1)/GSQ + SHAPC(2) * ST1/GAIN + .462
      ALPHA = -2.7726/SG
      TYPE 103
103   FORMAT(' NEW ENDING CHANNEL = ',$)
      ACCEPT 102,IEND
      IF(IEND)49,48,49
48    IEND = START + FLOAT(NDPTS) - 1.
49    NDPTS = IEND - IST + 1
      L = ST1 - START
      DO 60 I =1,NDPTS
      Y(I) = Y(L)
      L = L + 1
60    CONTINUE
      START = ST1 - 1.
44    AVEBG = 0.5 * (AVBG1 + AVBG2)
C     DETERMINE PEAK POSITIONS FROM GAIN AND ZERO VALUES
      DO 42 J = 1,NP
      PKPOS(J) = (EN(J) - REFEN)/ GAIN + REFCHA - START
      NPK = PKPOS(J) + 0.5
      PKHT(J) = Y(NPK)
42    CONTINUE
      FPKHT=0.
C     FIX THE INTENSITY RELATIONSHIPS THAT HAVE BEEN SPECIFIED
C     IN THE INPUT.  READJUST PEAK HEIGHT
C     ESTIMATES FOR INTERFERENCES FROM NEIGHBORING PEAKS.
      DO 70 J=1,NP
      L=J-1
      IF(L)65,65,61
61    DX=PKPOS(J)-PKPOS(L)
      IF(NYFLG(L).LT.0)PKHT(L)=FPKHT
      IF(FPKHT.EQ.0.)GOTO 51
      DPKHT=PKHT(L)*EXP(ALPHA*DX*DX)
      PKHT(J)=PKHT(J)-DPKHT
65    L=J+1
      IF(L-NP)63,63,70
63    DX=PKPOS(J)-PKPOS(L)
      IF(NYFLG(L))51,53,53
51    FPKHT=HIGHT(L)
      IF(HIGHT(L))62,53,53
62    K=-NYFLG(L)
      DXX=PKPOS(L)-PKPOS(K)
      FCT=-HIGHT(L)*EXP(ALPHA*DXX*DXX)
      PKHT(K)=PKHT(K)*(1.-FCT/(1.+FCT))
      FPKHT=-HIGHT(L)*PKHT(K)
      PKHT(L)=FPKHT
53    DPKHT=PKHT(L)*EXP(ALPHA*DX*DX)
      PKHT(J)=PKHT(J)-DPKHT
70    CONTINUE
C     FIX THE ENERGY RELATIONSHIPS THAT HAVE BEEN
C     SPECIFIED IN THE INPUT.
      DO 55 J =1,NP
      IF(NXFLG(J))64,55,55
64    L = -NXFLG(J)
      PKPOS(J) = PKPOS(L) -(EN(L) - EN(J))/GAIN
55    CONTINUE
      ITER = 1
      NFLG(6) = 0
      SLP(1) = SHAPC(2)/GAIN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CALL ASSIGN(NDEV,'SY:GNL.DAT')
      WRITE (NDEV) (AMATRX(I) , I = 1,IALGTH)
      CALL CLOSE(NDEV)
      CALL ASSIGN(NDEV,'SY:PASS.DAT')
      WRITE (NDEV) (BMATRX(I), I = 1,IBLGTH)
```

```
      CALL CLOSE(NDEV)
      CALL CHAIN(ULFIT2,0,0)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      END
```

```
C       PROGRAM ULFIT2
C       LINK WITH BWF,AVE,MLR,MLRA,MLRE
C       THIS SUBPROGRAM IS THE THIRD PART TO THE PROGRAM 'GRPANL'.
C       IT PERFORMS A TAYLOR'S EXPANSION ABOUT THE TRIAL VALUES
C       TO OBTAIN A SET OF LINEAR EQUATIONS, WHICH ARE SOLVED BY
C       THE METHOD OF LEAST SQUARES.  THE TRIAL VALUES ARE THEN
C       INCREMENTED BY THE RESULTS OF THE FIT.  IF THE INCREMENT(S)
C       EXCEEDS A SPECIFIED VALUE, THE SET OF EQUATIONS IS REFORMED,
C       SOLVED AND INCREMENTED.  THE PROCESS CONTINUES UNTIL ALL
C       TESTS ARE SATISFIED OR 10 ITERATIONS HAVE BEEN MADE,
C       WHICHEVER OCCURS FIRST.
        INTEGER*2 ULOUT2(4)
        DIMENSION AMATRX(5888),BMATRX(256)
        COMMON/GNL/ K,KK,NDPTS,NP,ITER,NWRDS(5),NFLG1,NFLG2,NFLG3,
      1    NFLG4,NFLG5,NXF,MWRDS(14),NYFLG(20),NXFLG(20),FNAME(3),ATAIL,
      2  BTAIL,GAIN,ALFA,START,ALPHA,FWHM,EXP1,EXP2,EXP3,EXP4,REFEN,
      3  REFCH,BG(2),AVEBG,ASLP,SLP,EN(20),SHAPC(10),CLIN(5),
      4  HIGHT(20),GAMA(20),YNET(240),PKPOS(20),PKHT(20),DELY(240),
      5  DEL(20),RM(240),VM(20),ACOEF(240,20),DM(20),ERR(7)
        COMMON/PASS/ENJ(20),IST,IEND,NUMSJ,NUMEJ,GAINJ,ZEROJ,ST1J,ST2J,
      1  STRJ(20),NENUM,ALPJ,TAMP,TSLP,TLAMP,TLSLP,BKGSLP(2),FILE(3),SSS,
      2  YES,BK1,NGAIN,IREGG,LISTFG,PAME(3),SUM(20)
        DIMENSION WM(20),NFLG(5)
        EQUIVALENCE (K,AMATRX(1)),(ENJ(1),BMATRX(1)),(NFLG(1),NFLG1)
C       THIS PROGRAM CHAINS TO THE SUBPROGRAM ULOUT2 ON RK3
        DATA ULOUT2/3RRK3,3RULO,3RUT2,3RSAV/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       READ IN THE COMMON BLOCKS FROM THE SYS DEVICE
        IALGTH = 5888
        IBLGTH = 256
        NDEV = 1
        CALL ASSIGN(NDEV,'SY:GNL.DAT')
        READ (NDEV) (AMATRX(I), I = 1,IALGTH)
        CALL CLOSE(NDEV)
        CALL ASSIGN(NDEV,'SY:PASS.DAT')
        READ (NDEV) (BMATRX(I), I = 1,IBLGTH)
        CALL CLOSE(NDEV)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
 1      KK = NFLG1 + NFLG2 + NFLG3 + NFLG4 + NFLG5
        IF(ITER - NFLG4 -NFLG5)32,33,33
 32     KK = KK -NFLG2
        NFLG2 = -NFLG2
 33     IF(ITER - NFLG3 -NFLG4 - NFLG5)34,35,35
 34     KK = KK - NFLG3
        NFLG3 = - NFLG3
 35     SG = -2.7726 / ALPHA
        CX = SQRT(-ALPHA)
        XPTS = NDPTS
C       CALCULATE PEAK TAILING INTENSITIES AT THE GROUP BOUNDARIES.
C       THESE VALUES USED LATER TO ADJUST NET COUNTS OF PEAK
C       GROUPING.
        BG(1) = 0.
        BG(2) = 0.
        DO 400 J = 1,NP
        TYPE 107
 107    FORMAT(' # OF BKG. CHANNELS IN BACK OF GROUP  = ',$)
        ACCEPT 106,NUME
        NUMEJ = NUME
        TYPE 108
 108    FORMAT(' GAIN (NEG. VALUE FOR ABSOLUTE' /
      1       '                  IF =0, PEAK AREA SIMPLY INTEGRATED) = ',$)
        ACCEPT 109,GAIN
 109    FORMAT(F9.6)
        NGAIN = 0
        TYPE 110
```

```
0040  110    FORMAT(' ENERGY OF ZERO CHAN =',$)
0041         ACCEPT 111,ZERO
0042         ZEROJ=ZERO
0043  111    FORMAT(F8.4)
0044         IF(GAIN)60,59,61
      C      IF GAIN=0, SIMPLY INTEGRATE NET COUNTS ABOVE BACKGROUND
0045  59     NDPTS=IEND-IST+1
0046         NDATA=NDPTS+NUMS+NUME
0047         XNUMS=NUMS
0048         ST2=ST-XNUMS
0049         IST2=ST2+1.
0050         CALL RDSK(IST2,NDATA,FNAME,Y)
0051         NE1=NDPTS+NUMS
0052         CALL MAXVAL(1,NDATA,JPK,YMAX,Y)
0053         CALL GFIT(JPK,PKPO,PHT,5,0.,0.,0.,Y)
0054         PKPO=PKPO+ST2 - 1.
0055         CALL NCNTS(1,NUMS,NE1,NDATA,ND,BGD,SUM,ER,AVEBG,Y,Y)
0056         ER=ER/SUM*100.
0057         PRINT 1089,PKPO
0058  1089   FORMAT(' PEAK POSITION:',F8.3)
0059         PRINT 1090,SUM,ER
0060  1090   FORMAT(' NET COUNTS FOR GIVEN REGION:',F8.0,'+- ',F7.2,'%')
0061         CALL EXIT
      C      GAIN IS ABSOLUTE
0062  60     GAIN = - GAIN
0063         GAINJ = GAIN
0064         NGAIN = 1
0065         RKEV = GAIN
0066         GOTO 62
      C      GET REFERENCE PEAKS FOR REDETERMINATION OF GAIN AND ZERO
0067  61     TYPE 112
0068  112    FORMAT(' REFERENCE PK #1 ENERGY = ',$)
0069         ACCEPT 113,STD(1)
0070         ST1J = STD(1)
0071  113    FORMAT(F8.3)
0072         TYPE 114
0073  114    FORMAT(' REFERENCE PK #2 ENERGY = ',$)
0074         ACCEPT 113,STD(2)
0075         ST2J = STD(2)
0076  62     TYPE 115
0077  115    FORMAT(' LIST PK ENERGIES IN THIS GROUP (NEG. VALUES FOR XRAYS)'
            1  / ' TERMINATE WITH A 0')
0078         I = 0
0079  2      I = I + 1
0080         TYPE 1150,I
0081  1150   FORMAT(I3,'  ',$)
0082         NXFLG(I) = 0
0083         GAMA(I) = 0.
0084         ACCEPT 116,EN(I)
0085  116    FORMAT(F10.4)
0086         NYFLG(I) = 1
0087         ENJ(I)= EN(I)
0088         STRJ(I) = ANO
0089         IF(EN(I))3,4,2
0090  3      TYPE 117
0091  117    FORMAT(' INTRINSIC WIDTH (EV) = ',/
            1  ' IF ENTRY=0, ESTIMATED VALUE USED ',$)
0092         ACCEPT 118,GAMA(I)
0093  118    FORMAT(F6.1)
0094         EN(I)=-EN(I)
0095         IF(GAMA(I).NE.0)GOTO 1181
0097         GAMA(I)=-0.4+0.4*EN(I)
0098         IF(EN(I).GT.28.)GAMA(I)=GAMA(I)-27.22+.972*EN(I)
0100  1181   GAMA(I) = GAMA(I) * .001/GAIN
0101         GOTO 2
```

```
4       NP = I -1
        NENUM = NP
        TYPE 1190,NP
1190    FORMAT(//,I3,' PEAKS WERE ENTERED FOR THIS REGION')
        TYPE 119
119     FORMAT(' TYPE SEQUENCE # OF PEAKS WHOSE ENERGY ARE UNCERTAIN'/
       1  ' TYPE 99 FOR ALL -- NEG. FOR FIXED RELATIONSHIP. END WITH 0')
12      ACCEPT 120,N
120     FORMAT(I3)
        IF(N)5,10,6
5       N = -N
        TYPE 121
121     FORMAT(' ENERGY FIXED RELATIVE TO PEAK #: ',$)
        ACCEPT 120,M
        NXFLG(N) = -M
        GOTO 12
6       IF(N-99)7,8,7
7       NXFLG(N) = 1
        STRJ(N) = AYES
        GOTO 12
8       DO 9 N = 1,NP
        STRJ(N) = AYES
9       NXFLG(N) = 1
10      TYPE 122
122     FORMAT(' SEQUENCE # OF PEAK WHOSE INTENSITY IS KNOWN '/
       1' 0 FOR NONE')
40      ACCEPT 120,N
        IF(N)41,48,41
41      NYFLG(N) = -N
        TYPE 123
123     FORMAT(' PEAK HEIGHT (NEG. FOR RATIO ) = ',$)
        ACCEPT 124,HIGHT(N)
124     FORMAT(E12.4)
        IF(HIGHT(N))42,43,43
42      TYPE 125
125     FORMAT(' RATIOEED TO PEAK #: ',$)
        ACCEPT 120,M
        NYFLG(N) = -M
43      GOTO 40
48      DO 133 I = 1,5
        SHAPC(I)=0.
133     CLIN(I) = 0.
        TYPE 126
126     FORMAT(' IS THERE A PARAMETER FILE? ',$)
        IF(.NOT.YESNO()) IVAL = 1
        IF(IVAL.EQ.1)GOTO 135
        TYPE 150
150     FORMAT(' FILE NAME? ',$)
        ACCEPT 101,CFILE
11      CALL ASSIGN(1,CFILE,10)
        READ (1) (NUM,SHAPC(I), I =1,NUM)
        CALL CLOSE(1)
        EXP3 = SHAPC(8)
        EXP4 = SHAPC(9)*GAIN
135     ALPJ = ANO
        STAMP = ANO
        STSLP = ANO
        TLAMP = ANO
        TLSLP = ANO
        NFLG(1) = 0
        NFLG(2) = 0
        NFLG(3) = 0
        NFLG(4) = 0
        NFLG(5) = 0
        IF(IVAL.EQ.1)GOTO 25
```

```
0168          TYPE 127
0169   127    FORMAT(' DO YOU WANT TO FREE ANY SHAPE PARAMETERS? ',$)
0170          IF(.NOT.YESNO())GOTO 24
0172          TYPE 128
0173   128    FORMAT(' ALPHA? ',$)
0174          IF(.NOT.YESNO())GOTO 15
0176          NFLG(1) = 1
0177          ALPJ = AYES
0178   15     TYPE 129
0179   129    FORMAT(' SHORT TERM TAIL AMPLITUDE? ',$)
0180          IF(.NOT.YESNO())GOTO 17
0182          NFLG(2) = 1
0183          STAMP = AYES
0184   17     TYPE 130
0185   130    FORMAT(' SHORT TERM TAIL SLOPE? ',$)
0186          IF(.NOT.YESNO())GOTO 22
0188          NFLG(3) = 1
0189          STSLP = AYES
0190   22     TYPE 140
0191   140    FORMAT( ' LONG TERM TAIL AMPLITUDE?'.$)
0192          IF(.NOT.YESNO())GOTO 23
0194          NFLG(4) = 1
0195          TLAMP = AYES
0196   23     TYPE 141
0197   141    FORMAT(' LONG TERM TAIL SLOPE?',$)
0198          IF(.NOT.YESNO())GOTO 24
0200          NFLG(5) = 1
0201          TLSLP = AYES
0202          GOTO 24
0203   25     TYPE 142
0204   142    FORMAT(' FWHM (IN KEV) =',$)
0205          ACCEPT 143,FWHM
0206   143    FORMAT(F7.4)
0207          TYPE 144
0208   144    FORMAT(' TO BE FREED? ',$)
0209          IF(.NOT.YESNO())GOTO 26
0211          NFLG(1) = 1
0212          ALPJ = AYES
0213   26     TYPE 145
0214   145    FORMAT(' SHORT TERM TAIL AMPLITUDE = ',$)
0215          ACCEPT 143,ATAIL
0216          TYPE 144
0217          IF(.NOT.YESNO())GOTO 27
0219          NFLG(2) = 1
0220          STAMP = AYES
0221   27     TYPE 146
0222   146    FORMAT(' SHORT TERM TAILING SLOPE = ',$)
0223          ACCEPT 143,BTAIL
0224          BTAIL = BTAIL * GAIN
0225          TYPE 144
0226          IF(.NOT.YESNO())GOTO 28
0228          NFLG(3) = 1
0229          STSLP = AYES
0230   28     TYPE 147
0231   147    FORMAT(' LONG TERM TAILING AMPLITUDE = ',$)
0232          ACCEPT 143,EXP3
0233          TYPE 144
0234          IF(.NOT.YESNO())GOTO 29
0236          NFLG(4) = 1
0237          TLAMP = AYES
0238   29     TYPE 148
0239   148    FORMAT(' LONG TERM TAILING SLOPE = ',$)
0240          ACCEPT 143,EXP4
0241          EXP4 = EXP4 * GAIN
0242          TYPE 144
```

```
      IF(.NOT.YESNO())GOTO 30
      NFLG(5) = 1
      TLSLP = AYES
30    SG = FWHM ** 2
      ALPHA = -2.7726 * GAIN * GAIN / SG
      SHAPC(2) = .0017!THIS IS AN APPROXIMATE VALUE USED FOR ASLP
      GOTO 31
24    GSQ = GAIN * GAIN
      AVECH = 0.5 * (IST + IEND)
      AVEN = AVECH * GAIN + ZERO
      SG = SHAPC(1)/GSQ + SHAPC(2) * (IST/GAIN+ZERO/GSQ) + .462
      FWHM = SQRT(SG)
      ATAIL = EXP(SHAPC(3) + AVEN * SHAPC(4))
      BTAIL = SHAPC(5) * GAIN
      ALPHA = -2.7726/ SG
31    TYPE 131
131   FORMAT(' LOW ENERGY BKGRD SLOPE EXPRESSED IN %/CHAN =',$)
      ACCEPT 132,SLP(1)
      BKGSLP(1) = SLP(1)
      TYPE 1331
1331  FORMAT(' HIGH ENERGY BKGRD SLOPE EXPRESSED IN %/CHAN =',$)
      ACCEPT 132,SLP(2)
      BKGSLP(2)=SLP(2)
132   FORMAT(F5.3)
C     THE FOLLOWING STATEMENTS WRITE THE COMMON BLOCKS TO
C     THE SYSTEM DEVICE.  THESE FILES ARE THEN READ BY ULOAD2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      NDEV = 1
      CALL ASSIGN(NDEV,'SY:GNL.DAT')
      WRITE (NDEV) (AMATRX(I), I = 1,IALGTH)
      CALL CLOSE(NDEV)
      CALL ASSIGN(NDEV,'SY:PASS.DAT')
      WRITE (NDEV) (BMATRX(I), I = 1,IBLGTH)
      CALL CLOSE(NDEV)
      CALL CHAIN(ULOAD2,0,0)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      END
```

```
C       PROGRAM ULOUT2
C       LINK WITH RDSK,AVE
C       ALSO LINK WITH CALMU,CALGPM,GPM IF
C       GAMMA EMISSIONS ARE INCLUDED
        LOGICAL*1 DTE(10),YESNO
        REAL*8 STIME,COFMU
        DIMENSION B(20),AMATRX(5888),BMATRX(256)
        COMMON/GNL/ K,KK,NDPTS,NP,NWRDS(26),NYFLG(20),NXFLG(20),
       1  FNAME(3),ATAIL,BTAIL,GAIN,ALFA,START,ALPHA,FWHM,EXP1,EXP2,
       2  EXP3,EXP4,REFEN,REFCH,BG(2),AVEBG,ASLP,SLP,EN(20),SHAPC(10),
       3  CLIN(5),HIGHT(20),GAMA(20),YNET(240),
       4  PKPOS(20),PKHT(20),RES(240),DEL(20),RM(240),VM(20),
       5  BM(4800),DM(20),ERR(7)
        COMMON/PASS/ENJ(20),IST,IEND,NUMSJ,NUMEJ,GAINJ,ZEROJ,ST1J,
       1  ST2J,STRJ(20),NENUM,ALPJ,TAMP,TSLP,TLAMP,TLSLP,BKGSLP(2),
       2  CFILE(3),SSS,YES,BK1,NGAIN,IREGG,LISTFG,DFILE(3),
       3  YSUM(20),QFT(100),SDOF(20),SUMR(20),EDGE(20),NSMBL(20),
       4  COFMU(8,20),MTLZ(5),CMPOS(5),NABS(5),ABSRB(5),DCNST(16),
       5  N1,N2,GEOM,SURFC,SMPWT,DEPTH
        EQUIVALENCE (K,AMATRX(1)),(ENJ(1),BMATRX(1))
        DATA Y/4HYES /
        DATA ANO/4HNO  /
        DATA BL/4H    /
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       READ IN COMMON BLOCKS FROM SYS DEVICE
        IALGTH = 5888
        IBLGTH = 256
        NDEV = 1
        CALL ASSIGN(NDEV,'SY:GNL.DAT')
        READ (NDEV) (AMATRX(I), I = 1,IALGTH)
        CALL CLOSE(NDEV)
        CALL ASSIGN(NDEV,'SY:PASS.DAT')
        READ (NDEV) (BMATRX(I), I = 1,IBLGTH)
        CALL CLOSE(NDEV)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        CALL DATE(DTE)
        SG = -2.7726/ ALPHA
        ALFA = -2.7726/SG
        FWHM = SQRT(SG)
        BTAIL = EXP2/GAIN
        EXP4 = EXP4/GAIN
        DO 4 I=1,100
4       QFT(I)=0.0
        QFT(50)=1.
        DX=0.
        L=50
        DO 6 I=51,100
        DX=DX+1.
        L=L-1
        QFT(I)=SQRT(EXP(ALFA*DX*DX))
        QFT(L)=QFT(I)
        IF(QFT(I)-.01)8,6,6
6       CONTINUE
8       DO 9 J=1,NP
        SDOF(J)=0.0
9       SUMR(J)=0.0
        SUM=0.0
        DO 11 I = 1,NDPTS
        RES(I) = RM(I) * SQRT (YNET(I) + AVEBG)
        RR = RM(I)**2
        SUM = SUM + RR
        DO 10 J=1,NP
        NPK=PKPOS(J)+0.5
        K=I-NPK+50
        IF(K.LT.0)GOTO 10
```

```
0050          IF(K.GT.100)GOTO 10
0052          SDOF(J)=SDOF(J)+QFT(K)
0053          SUMR(J)=SUMR(J)+QFT(K)*RR
0054   10     CONTINUE
0055   11     CONTINUE
0056          DOF = NDPTS - KK - NP
0057          QFIT = SUM / DOF
0058          FWHM = FWHM * GAIN
0059          CALL RDSK(1,20,FNAME,B)
0060          TIME = B(1)
0061          TOC = B(2)
0062          CALL GTIM(STIME)
0063          CALL CVTTIM(STIME,IHR,IMIN,ISEC,ITIC)
0064          WRITE(6,9010)(DTE(J),J=1,9),IHR,IMIN,FNAME,CFILE,
             1TIME,TIMOC,GAIN
0065   9010   FORMAT('1 PROGRAM GRPANL LLL VERSION 1',10X,9A1,I4,':',I2,//
             1' FILE NAME',2X,3A4,20X,' PARAMETER FILE',2X,3A4,//
             1' LIVE TIME',F8.1,' SECONDS',/
             1' START DAY    ',F7.3,/
             1' KEV/CHANNEL',F10.6,/)
0066          WRITE(6,9020)IST,IEND,NUMSJ,NUMEJ,ZEROJ,ST1J,ST2J,ALPJ,TAMP,
             1  TSLP,TLAMP,TLSLP,BKGSLP(1),BKGSLP(2)
0067   9020   FORMAT(/
             1' GROUP STARTS AT CHANNEL   ',I5/
             1' GROUP ENDS AT CHANNEL     ',I5/
             1' BACKGROUND LOW ENERGY     ',I5/
             1' BACKGROUND HIGH ENERGY    ',I5/
             1' ZERO          '13X,F8.4//
             1' REFERENCE PK #1 ENERGY =',F8.3/
             1' REFERENCE PK #2 ENERGY =',F8.3//
             1' PARAMETER TABLE            FREE'/
             1' ALPHA           ',12X,A4/
             1' SHORT TAIL AMPLITUDE'6X,A4/
             1' SHORT TAIL SLOPE     '6X,A4//
             1' LONG TERM TAIL AMPLITUDE',2X,A4/
             1' LONG TERM TAIL SLOPE',6X,A4//
             1' LOW ENERGY REGION BACKGROUND SLOPE   ',F8.4,/
             1' HIGH ENERGY REGION BACKGROUND SLOPE ',F8.4/)
0068          WRITE (6,9030)
0069   9030   FORMAT(/
             1' PEAK    ENERGY  WIDTH   UNCERTAIN  FIXED  UNCERTAIN   RATIOED
             1     RATIO'/
             1' NUMBER    KEV     EV    ENERGY     TO     INTENSITY   TO'/)
0070          DO 500 I=1,NENUM
0071          GAMA(I)=GAMA(I)*GAINJ/.001
0072          IF(GAMA(I).EQ.0)GAMA(I) = BL
0074          NX = BL
0075          IF(NXFLG(I).LT.0)NX=-NXFLG(I)
0077          ANY = Y
0078          NYY = BL
0079          IF(NYFLG(I).LT.0)ANY = ANO
0081          IF(NYFLG(I).LT.0)NYY = - NYFLG(I)
0083          IF(HIGHT(I).EQ.0)HIGHT(I)=BL
0085          WRITE (6,9040) I,ENJ(I),GAMA(I),STRJ(I),NX,ANY,NYY,HIGHT(I)
0086   9040   FORMAT(I6,F9.3,F7.2,5X,A4,3X,I4,9X,A4,3X,I4,9X,F8.5)
0087   500    CONTINUE
0088          WRITE(6,9050) QFIT,FWHM,ERR(3),GAIN,EXP1,ERR(4),BTAIL,ERR(5),
             1  EXP3,ERR(6),EXP4,ERR(7)
0089   9050   FORMAT(//'  ANALYSIS RESULTS',//
             1' QFIT',F6.1,' FWHM',F8.5,'+-',F6.4,' KEV/CHANNEL',F10.6,/,33X,
             1' EXP1',F9.5,'+-',F6.4,'  EXP2',F9.5,'+-',F6.4,/,33X,
             1' EXP3',F9.5,'+-',F6.4,'  EXP4',F9.5,'+-',F6.4,//)
0090          IF(EXP3.GE.0.)GOTO 38
0092          PRINT 9060
0093   9060   FORMAT('  EXP3 IS NEGATIVE---CHECK BACKGROUND')
```

```
C      THIS PORTION OF CODING FOR THE CALCULATION OF ABSOLUTE
C      GAMMA INTENSITIES
C      EFLG=0                !INITIALIZE FLAG
C      TYPE 110
C110   FORMAT (' DO YOU WANT PHOTON EMISSION RATES CALCULATED? ',$)
C      IF(.NOT.YESNO())GOTO 29
C      EFLG=1
C      TYPE 111
C111   FORMAT (' FILE NAME FOR DETECTOR CONSTANTS IS ',$)
C      ACCEPT 9111,DFILE
C9111  FORMAT(3A4)
C      CALL ASSIGN(1,DFILE,16)
C      READ(1)(NUM,DCNST(I),I=1,NUM)
C      CALL CLOSE(1)
C      TYPE 113
C113   FORMAT (' SAMPLE DISTANCE FROM DETECTOR =',$)
C      ACCEPT 9113,GEOM
C9113  FORMAT(F6.2)
C      WRITE(6,8113)GEOM
C8113  FORMAT(' SAMPLE DISTANCE FROM DETECTOR =',F6.2)
C      GEOM = GEOM + DCNST(15)
C      TYPE 114
C114   FORMAT (' SAMPLE AREA =',$)
C      ACCEPT 9113,SURFC
C      WRITE(6,8114)SURFC
C8114  FORMAT(' SAMPLE SURFACE AREA =',F6.2)
C      IF (SURFC) 21,22,21
C21    TYPE 115
C115   FORMAT (' SAMPLE WEIGHT =',$)
C      ACCEPT 9115,SMPWT
C9115  FORMAT(F8.3)
C      WRITE(6,8115)SMPWT
C8115  FORMAT(' SAMPLE WEIGHT =',F8.3)
C22    DEPTH = .00001
C      N1 = 0
C      IF (SMPWT) 24,25,24
C24    TYPE 116
C116   FORMAT (' SAMPLE DEPTH =',$)
C      ACCEPT 9116,DEPTH
C9116  FORMAT(F6.3)
C      WRITE(6,8116)DEPTH
C8116  FORMAT(' SAMPLE DEPTH =',F6.3)
C      TYPE 117
C117   FORMAT (' SPECIFY SAMPLE COMPOSITION (TERMINATE WITH 0 AMOUNT)'
C      1  /' ELEMENT AMOUNT(%)')
C      WRITE(6,117)
C      K = 0
C26    K = K + 1
C      READ(5,118)MTLZ(K),CMPOS(K)
C118   FORMAT (A2,F10.2)
C      WRITE(6,8118)MTLZ(K),CMPOS(K)
C8118  FORMAT(1X,A2,F10.2)
C      CMPOS(K) = 0.005 * CMPOS(K) * SMPWT / SURFC
C      IF (CMPOS(K)) 26,27,26
C27    N1 = K - 1
C25    TYPE 119
C119   FORMAT (' SPECIFY ABSORBERS USED',/' ELEMENT AMOUNT (G/SQ-CM)')
C      WRITE(6,119)
C      K = 0
C28    K = K + 1
C      READ (5,118) NABS(K),ABSRB(K)
C      WRITE(6,8118)NABS(K),ABSRB(K)
C      IF (ABSRB(K)) 28,29,28
C29    N2 = K - 1
C      IF (N1 + N2) 37,38,37
```

```
      C37   CALL ASSIGN(1,'SY:ABSCOF.DAT','OLD')
      C     READ(1,120)(NSMBL(J),EDGE(J),(COFMU(I,J),I=1,8),J=1,20)
      C120  FORMAT(1X,A2,F7.3,4F12.9/10X,4F12.9)
0094  38    L = KK
0095        WRITE (6,121)
0096  121   FORMAT (/3X,' CHANNEL    ENERGY  COUNTS  GAMMAS/MIN  PCTERR')
0097        DO 122 J = 1,NP
0098        SG = -2.7726/ALPHA + ASLP * PKPOS(J)
0099        ALFA = -2.7726/SG
      C     CALCULATE PORTION OF PEAK AREAS OUTSIDE BOUNDARIES
0100        FRACT1 = 0.
0101        FRACT2 = 0.
0102        IF(GAMA(J))48,48,50
0103  50    GAMA(J)=GAMA(J)*.001/GAIN
0104        FRAC1 = ATAN(2. * PKPOS(J)/GAMA(J))
0105        FRACT1 = 1. - .6366 * FRAC1
0106        DELT = NDPTS - PKPOS(J)
0107        FRAC2 = ATAN(2. * DELT/GAMA(J))
0108        FRACT2 = 1. - .6366 * FRAC2
0109  48    CTS = YSUM(J) + YSUM(J) * (FRACT1 + FRACT2)
0110        PCERR = 0.0
0111        IF(NYFLG(J))36,36,31
0112  31    L = L+1
0113        QFTL=SUMR(J)/(SDOF(J)-1.)
0114        PCERR = (100. * DM(L) * SQRT(QFTL))/PKHT(J)
0115  36    CHAN = PKPOS(J) + START
0116        EPK = REFEN + (CHAN  - REFCH) * GAIN
      C     THIS PORTION OF CODING FOR CALCULATION OF ABSOLUTE
      C     GAMMA INTENSITIES
      C     GAMAS =0.
      C     VALUE=CTS
      C     IF(EFLG.EQ.0)GOTO 40
      C     CALL CALGPM(EPK,VALUE)
      C     GAMAS=VALUE
      C     GAMAS = GAMAS / TIME * 60.
0117  40    WRITE (6,102) J,CHAN,EPK,CTS,GAMAS,PCERR
0118  102   FORMAT (I3,F9.3,F10.3,F9.0,E11.4,F6.1)
0119        IF (NXFLG(J)) 122,122,90
0120  90    L = L + 1
0121  122   CONTINUE
0122        IF(LISTFG.EQ.1) RETURN
0124        WRITE(6,9100)
0125  9100  FORMAT(' CHANNEL  ENERGY  INDEX   YNET    RESIDUALS  STD DEV')
0126        YP=0.
0127        DO 103 I=1,NDPTS
0128        DBG = BG(2)-BG(1)
0129        CALL AVE(1,NDPTS,TAREA,AV,YNET)
0130        IICH=I+START
0131        EENG = (FLOAT(IICH) - REFCH)* GAIN + REFEN
0132        YP = YP + YNET(I)
0133        YNET(I) = YNET(I) + BG(1) + DBG * YP/TAREA
0134        WRITE(6,9110) IICH,EENG,I,YNET(I),RES(I),RM(I)
0135  9110  FORMAT(I5,F11.2,I5,3F10.2)
0136  103   CONTINUE
0137        CALL CLOSE(6)
0138        STOP 'END OF JOB'
0139        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       PROGRAM FILES
C
C       THIS IS A GENERAL PURPOSE ROUTINE FOR THE PREPARATION OR EDITING
C       UNFORMATTED SEQUENTIAL DISK FILES
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        COMMON CON(1000)
        DIMENSION FNAME(5)
C
        DATA MY,MC,MI,MD/'Y','C','I','D'/
C
        TYPE 100
100     FORMAT(' FILE NAME IS ',$)
        ACCEPT 200,FNAME
200     FORMAT(5A4)
        NDEV =1
        TYPE 102
102     FORMAT(' IS THIS A NEW FILE? ',$)
        ACCEPT 202,NM
202     FORMAT(A1)
        IF(NM.EQ.MY) GOTO 10
C
C       FETCH EXISTING FILE
C
        CALL ASSIGN(NDEV,FNAME)
27      READ(NDEV)(LENGTH,CON(I), I= 1,LENGTH)
        CALL CLOSE(NDEV)
        TYPE 125,LENGTH
125     FORMAT(' FILE LENGTH =',I5)
        TYPE 106
106     FORMAT(' HOW MANY VALUES DO YOU WANT LISTED? ',$)
        ACCEPT 206,NUM
206     FORMAT(I4)
        IF(NUM)29,29,28
28      TYPE 107
107     FORMAT(' STARTING AT ',$)
        ACCEPT 206,NM
        NUM = NUM + NM - 1
        PRINT 208, FNAME(1),FNAME(2),NDEV
208     FORMAT(' FILE NAME IS ',2A4,' ON DEVICE # ',I1)
        WRITE(6,108)(I,CON(I), I = NM,NUM)
108     FORMAT(/(I4,1PE14.6))
        CALL CLOSE(6)
29      TYPE 110
110     FORMAT(' DO YOU WANT TO EDIT THE FILE? ',$)
        ACCEPT 202,NM
        IF(NM.NE.MY)GOTO 30
2       TYPE 111
111     FORMAT(' CHANGE(C),INSERT(I), OR DELETE(D)? ',$)
        ACCEPT 202,JD
        IF(JD.EQ.MC)GOTO 4
        IF(JD.EQ.MI)GOTO 6
        IF(JD.EQ.MD)GOTO 15
        GOTO 20
C       CHANGE VALUES IN EXISTING FILE
4       TYPE 112
112     FORMAT(' WHICH VALUE? ',$)
        ACCEPT 206,N
        IF(N) 7,8,8
C       A NEGATIVE VALUE INDICATES ENTRY POSITION IN THE FILE FOR A
C       SEQUENCE OF CHANGES
7       N = -N -1
        TYPE 121
121     FORMAT(' HOW MANY VALUES WILL YOU CHANGE? ',$)
```

```
0056          ACCEPT 206,NUM
0057          GOTO 13
0058  8       TYPE 114
0059  114     FORMAT(' NEW VALUE = ',$)
0060          ACCEPT 214,AA
0061  214     FORMAT(E12.5)
0062          CON(N)= AA
0063          TYPE 116
0064  116     FORMAT(' MORE CHANGES? ',$)
0065          ACCEPT 202,NM
0066          IF(NM.EQ.MY)GOTO 2
0068          GOTO 20
      C       INSERT A NEW VALUE
0069  6       TYPE 118
0070  118     FORMAT(' INSERT BEFORE WHICH VALUE? ',$)
0071          ACCEPT 206,N
0072          L = LENGTH
0073          DO 9 I = N,LENGTH
0074          CON(L + 1)=CON(L)
0075  9       L = L - 1
0076          LENGTH = LENGTH + 1
0077          GOTO 8
      C       DELETE A VALUE FROM THE FILE
0078  15      TYPE 112
0079          ACCEPT 206,N
0080          LENGTH = LENGTH - 1
0081          DO 17 I = N,LENGTH
0082  17      CON(I) = CON(I+1)
0083          GOTO 2
      C       PREPARE FOR FORMATION OF A NEW FILE
0084  10      TYPE 117
0085  117     FORMAT(' FILE LENGTH =',$)
0086          ACCEPT 206,LENGTH
      C       ZERO OUT THE ARRAY BEFORE ENTERING VALUES
0087          NUM = LENGTH
0088          DO 11 I = 1,LENGTH
0089  11      CON(I) = 0.
0090          N = 0
0091  13      NUM = NUM + N
0092          TYPE 120
0093  120     FORMAT(' TYPE VALUES IN SEQUENCE.')
0094  12      N = N + 1
0095          TYPE 122,N
0096  122     FORMAT(I4,2X,$)
0097          ACCEPT 214,CON(N)
0098          IF(N.NE.NUM)GOTO 12
0100  20      CALL ASSIGN(NDEV,FNAME)
0101          WRITE (NDEV) (LENGTH,CON(I), I = 1,LENGTH)
0102          CALL CLOSE(NDEV)
0103  30      STOP 'ALL DONE'
0104          END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       Program PKSHAP
C
C       This program determines a set of parameters used to describe the
C       peak shapes of a gamma spectrum. Input comes from shape
C       characteristics of a low and high energy peak, each from the
C       GRPANL program.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        DIMENSION EN(2),CH(2),FWHM(2),ATAIL(2),WP(2),TP(2),SHAPC(10)
C
        TYPE 100
100     FORMAT( ' Peakshape V1...use parameters from GRPANL.'/
       +  ' Gain (keV./ch) = ',$)
        ACCEPT 102, GAIN
        PRINT 200, GAIN
200     FORMAT (' Gain = 'F5.3,' keV./channel')
        TYPE 201
201     FORMAT (' Low energy peak (keV.) = ',$)
        ACCEPT 101, EN(1)
101     FORMAT (F8.3)
        TYPE 202
202     FORMAT (' FWHM = ',$)
        ACCEPT 102, FWHM(1)
102     FORMAT (F5.3)
        TYPE 203
203     FORMAT (' EXP1 = ',$)
        ACCEPT 103, ATAIL(1)
103     FORMAT (F5.3)
        TYPE 204
204     FORMAT (' Hi energy peak (keV.) = ',$)
        ACCEPT 101, EN(2)
        TYPE 202
        ACCEPT 102, FWHM(2)
        TYPE 203
        ACCEPT 103, ATAIL(2)
        TYPE 206
206     FORMAT (' EXP2 = ',$)
        ACCEPT 102, BTAIL
        GSQ = GAIN * GAIN
        DO 1 J = 1,2
         WP(J) = FWHM(J) * FWHM(J) - 0.462 * GSQ
         TP(J) = ALOG (ATAIL(J))
1       CONTINUE
        DO 2 I = 1,10
         SHAPC(I) = 0.0
2       CONTINUE
C       Calculate the area / peak-height ratio for a 100 keV. peak.
        DEN = 1. / (EN(1) - EN(2))
        SA = (WP(1)- WP(2)) * DEN
        SHAPC(1) = WP(1) - EN(1) * SA
        SHAPC(2) = SA
        SA = (TP(1) - TP(2)) * DEN
        SHAPC(3) = TP(1) - EN(1) * SA
        SHAPC(4) = SA
        SHAPC(5) = BTAIL
C
        WRITE (6,110) SHAPC(1),SHAPC(2),SHAPC(3),SHAPC(4),SHAPC(5)
110     FORMAT (' Shape parameters are :',//' S(1) =',F8.5,/' S(2) =',
       1  E11.4,/' S(3) =',E10.3,/' S(4) =',E10.3,/
       2  ' S(5) ='F6.3,//)
        CALL EXIT
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      SUBROUTINE RDSK
C
C      THIS IS A ROUTINE FOR READING UNFORMATTED DIRECT ACCESS FILES
C      CONTAINING 64 FLOATING POINT VALUES PER RECORD.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE RDSK (NST,NDP,FNAM,ARRAY)
C
       DIMENSION BUF(64),FNAM(3),ARRAY(2)
C
       I = 1
       NDEV = 3
       CALL ASSIGN (NDEV,FNAM)
       DEFINE FILE NDEV (64,128,U,NREC)
       NREC = (NST - 1) / 64
       N1 = NST - 64 * NREC
       NREC = NREC + 1
10     READ (NDEV'NREC) (BUF(L), L = 1,64)
15     ARRAY(I) = BUF(N1)
       N1 = N1 + 1
       I = I + 1
       IF (I .GT. NDP) GO TO 30
       IF (N1 - 64) 15,15,20
20     N1 = 1
       GO TO 10
30     CALL CLOSE (NDEV)
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     LOGICAL FUNCTION YESNO
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      LOGICAL FUNCTION YESNO*1 (IANS)
      LOGICAL*1 IST(6),ISTR(4)
C
5     CALL GETSTR(5,IST,3,ERR)
      IF(ERR.EQ..TRUE.)GO TO 10
      CALL SUBSTR(IST,ISTR,1,1)
      CALL SCOMP('Y',ISTR,IVAL)
      IF(IVAL.EQ.0)GO TO 20
      CALL SCOMP('N',ISTR,IVAL)
      IF(IVAL.EQ.0)GO TO 30
10    TYPE 100
100   FORMAT('  PLEASE ANSWER Y OR N')
      GOTO 5
20    YESNO = .TRUE.
      RETURN
C
30    YESNO = .FALSE.
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     SUBROUTINE MAXVAL
C
C
C     THIS ROUTINE RETURNS THE POSITION AND VALUE OF THE MAXIMUM VALUE
C     IN A GROUP OF DATA POINTS FROM NS TO NE INCLUSIVE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MAXVAL (NS,NE,JMAX,YMAX,Y)
      DIMENSION Y(2)
      YMAX = 0.0
      DO 2 I = NS,NE
      IF (Y(I) - YMAX) 2,2,4
    4   YMAX = Y(I)
      JMAX = I
    2   CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     SUBROUTINE MINVAL
C
C     THIS ROUTINE RETURNS THE POSITION AND VALUE OF THE MINIMUM VALUE
C     IN A GROUP OF DATA POINTS FROM NS TO NE INCLUSIVE.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MINVAL (NS,NE,JMIN,YMIN,Y)
      DIMENSION Y(2)
      YMIN=1.0E38
      DO 4 I=NS,NE
      IF (Y(I)-YMIN)2,4,4
2     JMIN=I
      YMIN=Y(I)
4     CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Subroutine NCNTS
C
C      This routine is similar to NCTS except it returns an average
C      background and an error associated with the total net count.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE NCNTS (N1,N2,M1,M2,ND,S1,S2,ER,AVEBG,Y1,Y2)
       DIMENSION Y1(2),Y2(2)
C
       CHANS = M1 - N2 + 1
       XPTS = M2 + N2 - M1 - N1 + 1
       CALL NCTS (N1,N2,M1,M2,ND,S1,S2,AV1,AV2,Y1,Y2)
       AVEBG = 0.5 * (AV1 + AV2)
       IF (S2.LT.0.0)S2 = 0.0
       ER = SQRT (S2 + AVEBG * CHANS * (1. + CHANS / XPTS))
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Subroutine NCTS
C
C      This routine return the net channel counts in a peak, the sum of
C      the net counts, the average background levels in front and back
C      of a peak group, and the number of data points in the net count
C      array.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE NCTS (N1,N2,M1,M2,ND,S1,S2,AV1,AV2,Y1,Y2)
       DIMENSION Y1(2),Y2(2)
       CALL AVE (N1,N2,S1,AV1,Y1)
       CALL AVE (M1,M2,S2,AV2,Y1)
       S1 = S1 + S2
       K = N2 + 1
       KK = M1 - 1
       CALL AVE (K,KK,S2,DUM,Y1)
       DBG = AV2 - AV1
       L = 0
       YS = 0.0
       XS = 0.0
       YSS = 0.0
       ITER = 0
       CHANS = KK - K + 1
       S2 = S2 - AV2 * CHANS
       IF (DBG) 1,4,4
1      DO 2 I = K,KK
       YS = YS + Y1(I) - AV2
       BG = AV1 + DBG * YS / S2
       IF (ITER) 9,9,7
7      YSS = YSS + Y1(I) - BG
       BG = AV1 + DBG * YSS / S3
9      CNET = Y1(I) - BG
       IF (ITER) 2,2,11
11     L = L + 1
       Y2(L) = CNET
2      XS = XS + CNET
       IF (ITER) 5,5,8
5      S3 = XS
       XS = 0.0
       YS = 0.0
       ITER = 1
       GO TO 1
4      XK = KK - K + 1
       YS = AV1
       DBG = DBG / XK
       DO 6 I = K,KK
       L = L + 1
       YS=YS + DBG
       CNET = Y1(I) - YS
       Y2(L) = CNET
6      XS = XS + CNET
8      S2 = XS
       ND = L
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     SUBROUTINE AVE
C
C     THIS ROUTINE DETERMINES THE AVERAGE COUNT OF A GROUP OF DATA
C     VALUES FROM NS TO NE INCLUSIVE.  IT ALSO RETURNS THE SUM.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE AVE (NS,NE,SUM,AV,Y)
      DIMENSION Y(2)
      SUM = 0.0
      SM = NE - NS + 1
      DO 2 I = NS,NE
    2   SUM = SUM + Y(I)
      AV = SUM / SM
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       SUBROUTINE BKGRD
C
C
C       THIS SUBROUTINE CALCULATES THE BACKGROUND DISTRIBUTION
C       FOR THE GIVEN REGION
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE BKGRD(NS,NE,AVBG1,AVBG2,SLP1,SLP2,SUM,L,Y,YNET)
        DIMENSION Y(2),YNET(2)
        XPTS = NE - NS
        DBG = (AVBG2 - AVBG1) - 0.5 * (SLP1 + SLP2) * XPTS
        XPTS = XPTS - 1.0
        NS = NS + 1
        NE = NE - 1
        BG2 = AVBG2
        SM = 0.0
        DO 2 I = NS,NE
2       SM = SM + Y(I)
        SM = SM - BG2 * XPTS
        XS = 0.0
        YS = 0.0
        FDY = 0.0
        L = 0
        XBG = AVBG1
        DSLP = SLP2 - SLP1
        S1 = SLP1
        DO 10 I = NS,NE
        L = L + 1
        DY = Y(I) - BG2
        HDY = 0.5 * DY
        YS = YS +HDY + FDY
        RT1 = YS/SM
        XBG = XBG + DBG * DY /SM + S1 + RT1 * DSLP
        FDY = HDY
        YNET(L) = Y(I) - XBG
10      XS = XS + YNET(L)
        SUM = XS
        RETURN
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SUBROUTINE GFIT
C
C
C      THIS ROUTINE PERFORMS A GAUSSIAN FIT TO 5, 7, OR 9 POINTS
C      SURROUNDING THE L TH CHANNEL AND RETURNS THE EXACT PEAK POSITION
C      AND THE PEAK HEIGHT.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE GFIT (L,PKPOS,PKHT,NC,ALFA,EXP1,EXP2,Y)
       DIMENSION Y(2)
       SY = 0.0
       SXY = 0.0
       SXXY = 0.0
       LX = - NC / 2
       Y1 = Y(L+1)
       Y2 = Y(L-1)
       TPOS = .5 * (Y2-Y1) / (Y1 - 2. * Y(L) + Y2)
       Y0 = Y(L)
       DO 2 I = 1,NC
       K = L + LX
       XL = LX
       DLX = XL - TPOS
       Y1 = Y(K)
       IF (DLX) 1,7,7
1      Y1 =Y1-Y0 * EXP1* EXP(EXP2 * DLX) * (1. -EXP(.4*ALFA*DLX*DLX))
7      IF (Y1) 5,5,3
5      Y1 = 1.0
3      YLOG = ALOG (Y1)
       YX = XL * YLOG
       SXY = SXY + YX
       SY = SY + YLOG
       SXXY = SXXY + XL * YX
       LX = LX + 1
2      CONTINUE
       PKP = L
       IF (NC - 7) 4,6,8
4      PKPOS = SXY / (2.8571 * SY - 1.4286 * SXXY)
       PKHT = 0.4857 * SY - 0.14286 * SXXY + .05 * SXY * PKPOS
       GO TO 10
6      PKPOS = SXY / (2.6666 * SY - 0.6666 * SXXY)
       PKHT = 0.3333 * SY - 0.04762 * SXXY + .017857 * SXY * PKPOS
       GO TO 10
8      PKPOS = SXY / (2.597 * SY - 0.3896 * SXXY)
       PKHT = 0.2554 * SY - 0.021645 * SXXY + .0083333 * SXY * PKPOS
10     PKHT = EXP (PKHT)
       PKPOS = PKPOS + PKP
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      SUBROUTINE BWF
C
C      THIS SUBROUTINE CALCULATES THE LORENTZIAN BROADENED
C      DISTRIBUTION FOR X-RAYS.  THE VALUES GIVEN FOR THE CA AND CB
C      ARRAYS ARE HEAVY ELEMENTS.  THE VALUES GIVEN AT THE RIGHT SHOULD
C      BE ENTERED IN THE ARRAYS FOR LIGHT ELEMENTS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       SUBROUTINE BWF (X)
       DIMENSION CA(5),CB(6)
C      HEAVY ELEMENTS                                   LIGHT ELEMENTS
       CA(1) = .729281                                  !.681634
       CA(2) = .0344557                                 !.323828
       CA(3) = .127639                                  !.138754
       CA(4) = -.0325239                                !-.0360606
       CA(5) = .00642257                                !.00687045
       CB(1) = .933301                                  !1.09498
       CB(2) = -4.45603                                 !-4.6894
       CB(3) = 1.16191                                  !1.18068
       CB(4) = -.141124                                 !.0241993
       CB(5) = -.0352464                                !.0952378
       CB(6) = .00744387                                !.0159129
       IF (X - 2.2) 2,10,10          !REPLACE 2.2 WITH 2.3 FOR LIGHT Z
 2     XX = X * X
       AX = 1.
       SUM = 0.0
       DO 4 I = 1,5
       AX = AX * XX
 4     SUM = SUM + CA(I) * AX
       X = SUM * EXP (-XX)
       GO TO 20
 10    XLOG = ALOG (X)
       SUM = CB(1)
       DO 12 I = 2,6
 12    SUM = SUM + CB(I) * XLOG ** (I-1)
       X = EXP (SUM)
 20    RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       SUBROUTINE MLR
C
C       THIS IS THE FIRST PART OF A LEAST-SQUARES PROGRAM WHICH WAS
C       ADAPTED FROM THE 'MLR' PROGRAM WRITTEN BY R. E. VON HOLDT,LLL.
C       THE SEQUEL IS MLRA OR MLRS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE MLR (LB,M,N,Y,X,B,R,W,V,NFLG)
      DIMENSION Y(2),B(2),R(2),X(2),W(2),V(2)
      IB = LB
        DO 70 I = 1,N
      T = 0.
      IK = (I-1) * IB
      DO 20 K = I,M
      LL = IK + K
        T = T + B(LL) * B(LL)
20    CONTINUE
      IF (T) 30,29,30
29    NFLG = 2
      GO TO 75
30    ST = SQRT (T)
      LL = IK + I
      W(I) = ST
      IF (B(LL)) 31,32,32
31      W(I) = -ST
32      B(LL) = B(LL) + W(I)
      V(I) = -B(LL) * W(I)
      IF (I-N) 35,75,35
35      IP = I + 1
      IF(IP.GT.N)GOTO 70
      DO 60 J = IP,N
      T = 0.
      JK = (J-1) * IB
      DO 40 K = I,M
      LL = IK + K
      LLL = JK + K
40      T = T + B(LL) * B(LLL)
      T = T / V(I)
      DO 50 L = I,M
      LL = JK + L
      LLL = IK + L
50      B(LL) = B(LL) + T * B(LLL)
60      CONTINUE
70      CONTINUE
75    CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       SUBROUTINE MLRA
C
C       THIS IS A SEQUEL TO THE MLR LEAST-SQUARES ROUTINE ADAPTED
C       FROM THE MLR PROGRAM WRITTEN BY R. H. VON HOLDT,LLL.
C       THIS SUBROUTINE REQUIRES LESS CORE BECAUSE THE 'A' MATRIX
C       IS STORED AS A FILE ON THE SYSTEM DEVICE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE MLRA (LB,M,N,Y,X,B,R,W,V,NFLG)
        DIMENSION Y(2),A(240),B(2),R(2),X(2),W(2),V(2)
        IB = LB
        DO 80 I = 1,M
80      R(I) = Y(I)
        DO 90 I = 1,N
90      X(I) = 0.
        S = 0.
        DO 100 L = 1,M
100     S = S + R(L) * R(L)
        YE = S
105     DO 130 I = 1,N
        T = 0.
        LK = (I-1) * IB
        DO 110 K = I,M
        LL = LK + K
110     T = T + B(LL) * R(K)
        T = T/V(I)
        DO 120 L = I,M
        LL = LK + L
120     R(L) = R(L) + T * B(LL)
130     CONTINUE
        IN = N
140     R(IN) = R(IN)/W(IN)
150     J = IN
155     J = J -1
        IF (J) 190,190,160
160     LL = (IN-1) * IB + J
        R(J) = R(J) + R(IN) * B(LL)
        GO TO 155
190     IN = IN - 1
        IF (IN) 200,200,140
200     DO 210 I = 1,N
        SC = X(I)
        SD = R(I)
        XD = SC - SD
210     X(I) = XD
        DO 215 I = 1,M
215     R(I) = Y(I)
C       READ SECTIONS OF 'A' MATRIX FROM PREVIOUSLY STORED 'AMTRX' FILES
C       STORED ON SYS
        CALL ASSIGN(1,'SYS:AMTRX.DAT',0,'OLD','CC')
        DO 220 J = 1,N
        READ (1) (A(I) , I = 1,IB)
        XD = X(J)
        DO 230 I = 1,M
230     R(I) = R(I) - A(I) * XD
220     CONTINUE
        CALL CLOSE(1)
        RE = 0.
        DO 240 I = 1,M
240     RE = RE + R(I) * R(I)
        IF (RE - S) 250,270,270
250     S = RE
        GO TO 105
```

```
0054  270    IF (RE - YE) 280,290,290
0055  290    NFLG = 3
0056  280    RETURN
0057         END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       SUBROUTINE MLRE
C
C       THIS ROUTINE OBTAINS THE DIAGONAL MATRIX ELEMENTS FOLLOWING A
C       LEAST-SQUARES CALCULATION.  THESE CAN BE USED FOR ERROR
C       ANALYSIS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE MLRE(D,IA,NN,B,W)
        DIMENSION D(2),B(2),W(2)
        N = NN
        DO 6 I =1,N
        D(I) = -1. / W(I)
        IF(N-1)4,4,1
1       JJ = I + 1
        IF(JJ.GT.N)GOTO 4
        DO 3 J =JJ,N
        S = 0.
        KJ = J -1
        DO 2 K =I,KJ
        LL = (J - 1) * IA + K
        S = S + D(K) * B(LL)
2       CONTINUE
C
C
C
        D(J) = S * 1./W(J)
3       CONTINUE
4       S = 0
        DO 5 L =I,N
        S = S + D(L) **2
5       CONTINUE
        D(I) = S
6       CONTINUE
        RETURN
        END
```

```
C       GPM
        SUBROUTINE GPM (EPK,ELOG,GEOM,SURFC,DEPTH,SMU,VALUE,GAMAS,DCNST)
        DIMENSION DCNST(2)
        EXOVR = DCNST(12)
        DETHT = DCNST(13)
        RSQ = DCNST(14) * DCNST(14)
        XMU = -2.316 + 4.2 * EXP(-.478 * ELOG - 1.434)
        XMU = EXP (XMU)
        XM = 1.0 / XMU
        SEE = EXP (-DETHT * XMU)
        XM = XM * (1. - XMU * DETHT * SEE / (1. - SEE))
        GM = GEOM + XM
        GG = GM * GM
        ROSQ = -GG + .5 * RSQ / (1. - GM/SQRT (GG+ RSQ))
        ERSQ = ROSQ * (1.1 -  XM / SQRT (DETHT))
        GGSQ = GG + ERSQ
        GLOST = 1.
        GCORR = 1.
        SQGCR = 1.
        SATN = 1.
        IF (SURFC) 21,31,21
21      RR = SURFC / 3.1416
        GM2 = GM - DEPTH * (1. - EXP (-SMU))
        GG2 = GM2 * GM2
        RR = RR * (1.4 - .4 / GM2)
        GCORR = RR / (GG2 * ALOG ((GG2 + RR)/GG2))
        SQGCR = SQRT (GCORR)
        SMU = SMU * SQGCR
        UL = SMU * SMU
        TOLOA = 2. * DEPTH / GM
        ADD = 1.
        DFACT = 1.
        ASQ = GGSQ * GCORR
        TOTAL = (ASQ /(ASQ -DEPTH*DEPTH)) + .1 * SMU * (TOLOA **3)
        DO 25 I = 1,10
        XK = 2 * I
        DFACT = DFACT * XK * (XK + 1.)
        ADD = UL * ADD
        AD = ADD / DFACT
        IF (TOTAL - AD * 200.) 24,24,27
24      TOTAL = TOTAL + AD
25      CONTINUE
27      TOTAL = TOTAL + AD
        ADD = SMU
        DFACT = 1.
        DO 28 I = 1,5
        XK = 2 * I - 1
        AD = TOLOA * ADD / ((XK + 2.) * DFACT)
        IF (TOTAL - AD *200.) 29,29,30
29      TOTAL = TOTAL + AD
        ADD = ADD * UL
        DFACT = DFACT * (XK + 1.) * (XK + 2.)
28      CONTINUE
30      TOTAL = TOTAL + AD
        SATN = EXP (-SMU)
        GLOST = TOTAL * SATN
31      GM = GM - DEPTH * (1. - SATN)
        SGSQ = SQRT (GM * GM + ROSQ)
        SMU = DCNST(16) * (SGSQ - GM) / (SGSQ + GM)
        N = 1
        M = 6
        IF (EPK - EXOVR) 32,32,34
32      N = 7
        M = 11
34      EFLOG = DCNST(N)
```

```
      L = N+1
      DO 36 I = L,M
   36 EFLOG = EFLOG + DCNST(I) * ELOG ** (I-N)
      EFF = EXP (EFLOG)
      GAMAS = VALUE * GGSQ * GCORR / (EFF * GLOST)
      RD = .7 * DCNST(14)
      VALUE = (1. + RD / GG) * SQGCR
      RETURN
      END
```

```
      C     CALMU
0001        SUBROUTINE CALMU (M,XMU,EPK,ELOG,LABZ,AMAS,KABZ,EDGE,COFMU)
0002        REAL*8 COFMU
0003        DIMENSION LABZ(2),AMAS(2),KABZ(2),EDGE(2),COFMU(8,20)
0004        XMU = 0.0
0005        DO 10 L = 1,M
0006        DO 2 K = 1,20
0007        IF (LABZ(L) - KABZ(K)) 2,4,2
0008  2     CONTINUE
0009        WRITE (6,120)
0010  120   FORMAT ('CANNOT MAKE LIBRARY MATCH WITH MATERIALS SPECIFIED')
0011        K = 1
0012  4     ABSMU = COFMU (1,K)
0013        DO 6 J = 2,8
0014  6     ABSMU = ABSMU + COFMU(J,K) * ELOG **(J -1)
0015        ABSMU = EXP (ABSMU)
0016        IF (EPK - EDGE(K)) 7,7,8
0017  7     XJUMP = .10266 + .006798 * SQRT (EDGE(K)) + .0006539 * EDGE(K)
0018        ABSMU = ABSMU * XJUMP
0019  8     XMU = XMU + ABSMU * AMAS(L)
0020  10    CONTINUE
0021        RETURN
0022        END
```

```
      C      CALGPM
0001         SUBROUTINE CALGPM (EPK,VALUE)
0002         REAL*8 COFMU
0003         COMMON/PASS/ENJ(20),IST,IEND,NUMSJ,NUMEJ,GAINJ,ZEROJ,ST1J,
            1  ST2J,STRJ(20),NENUM,ALPJ,TAMP,TSLP,TLAMP,TLSLP,BKGSLP(2),
            2  CFILE(3),SSS,YES,BK1,NGAIN,IREGG,LISTFG,DFILE(3),
            3  YSUM(20),QFT(100),SDOF(20),SUMR(20),EDGE(20),NSMBL(20),
            4  COFMU(8,20),MTLZ(5),CMPOS(5),NABS(5),ABSRB(5),DCNST(16),
            5  NF1,NF2,GEOM,SURFC,SMPWT,DEPTH
0004         DATA MTRL/'GE'/
0005         ELOG = ALOG (.001 * EPK)
0006         SMU = 0.0
0007         DEPTH = 0.5 * DEPTH
0008         IF (NF1) 14,14,13
0009   13    CALL CALMU (NF1,SMU,EPK,ELOG,MTLZ,CMPOS,NSMBL,EDGE,COFMU)
0010   14    CALL GPM (EPK,ELOG,GEOM,SURFC,DEPTH,SMU,VALUE,GAMAS,DCNST)
0011         IF (EPK - 500.) 9,9,10
      C      CORRECT FOR GE 'DEAD LAYER' ATTENUATION.
0012   9     CALL CALMU (1,XMU,EPK,ELOG,MTRL,SMU,NSMBL,EDGE,COFMU)
0013         GAMAS = GAMAS * EXP (XMU)
0014   10    IF (NF2) 15,18,15
0015   15    ATN = 0.0
0016         CALL CALMU (NF2,AMU,EPK,ELOG,NABS,ABSRB,NSMBL,EDGE,COFMU)
      C      ADJUST ABSORBER ATTENUATION TO ACCOUNT FOR GAMMAS TRAVERSING
      C      THE ABSORBER AT SKEWED ANGLES.
0017         AMU = AMU * VALUE
0018         IF (AMU - 6.9) 16,17,17
0019   16    ATN = EXP (AMU)
0020   17    GAMAS = GAMAS * ATN
0021   18    VALUE = GAMAS
0022         RETURN
0023         END
```