NASA TECHNICAL NOTE

NASA TN D-8398 c.1

NASA TN D-8398

SAVLOC, COMPUTER PROGRAM FOR AUTOMATIC CONTROL AND ANALYSIS OF X-RAY FLUORESCENCE EXPERIMENTS

Regis F. Leonard

Lewis Research Center

Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1977

1. Report No. NASA TN D-8398	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle SAVLOC, COMPUTER PROGRAM FOR AUTOMATIC CONTROL AND ANALYSIS OF X-RAY FLUORESCENCE EXPERIMENTS		5. Report Date April 1977
		6. Performing Organization Code
7. Author(s) Regis F. Leonard		8. Performing Organization Report No. E-9000
		10. Work Unit No. 506-25
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A program for a PDP-15 computer is presented which provides for control and analysis of trace element determinations by using X-ray fluorescence. The program simultaneously handles data accumulation for one sample and analysis of data from previous samples. Data accumulation consists of sample changing, timing, and data storage. Analysis requires the locating of peaks in X-ray spectra, determination of intensities of peaks, identification of origins of peaks, and determination of areal density of the element responsible for each peak. The program may be run in either a manual (supervised) mode or an automatic (unsupervised) mode.

17. Key Words (Suggested by Author(s)) Computer program; Trace elements; X-ray fluorescence	18. Distribution Statement Unclassified - unlimited; STAR Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	43	A03

* For sale by the National Technical Information Service, Springfield, Virginia 22161

SAVLOC, COMPUTER PROGRAM FOR AUTOMATIC CONTROL AND ANALYSIS OF X-RAY FLUORESCENCE EXPERIMENTS

by Regis F. Leonard

Lewis Research Center

SUMMARY

A program for a PDP-15 computer is presented which provides for control and analysis of trace element determinations by using X-ray fluorescence. The program simultaneously handles data accumulation for one sample and analysis of data from previous samples. Data accumulation consists of sample changing, timing, and data storage. Analysis requires the locating of peaks in X-ray spectra, determination of intensities of peaks, identification of origins of peaks, and determination of areal density of the element responsible for each peak. The program may be run in either a manual (supervised) mode or an automatic (unsupervised) mode.

INTRODUCTION

The recent development of high-resolution solid-state X-ray detectors has made possible the use of X-ray fluorescence as a tool for the detection of trace elements in a variety of applications. However, analysis of environmental samples by means of X-ray fluorescence generates energy spectra for which quantitative hand analysis is prohibitively tedious. In addition, the number of samples generated by any comprehensive monitoring program requires a speed of data acquisition and analysis which is possible only through the use of a computerized system. The present report describes the software designed for control and data analysis of such a system. This report should permit the reader to operate the system.

PRINCIPLES OF X-RAY FLUORESCENCE AND CHARACTER OF DATA

When properly stimulated, the atoms of any element emit X-rays whose energy (or wavelength) is strictly dependent upon the atomic number of the emitting atom. Since these energies may be quickly and accurately measured, and since the X-ray energies of all elements are well known and tabulated, it is possible to identify, with little ambiguity, the element responsible for the emission. In addition, the number of X-rays emitted is proportional to the number of atoms stimulated, so that after a proper calibration, quantitative measurements of areal densities are possible.

The X-ray detectors commonly employed provide data in the form of analog electric pulses. Data acquisition consists of the digitizing and storing of these pulses, which arrive with random timing at the rate of several thousand per second. A schematic representation of the X-ray fluorescence facility is shown in figure 1. A typical X-ray spectrum is shown in figure 2. This particular spectrum resulted from excitation of a sample by using X-rays from a tungsten X-ray tube. It contains a peak (labelled Rh) which resulted from the presence in the primary X-ray beam of a small, known amount of rhodium, whose purpose was to provide a monitor of the number of photons incident on the sample.

Data analysis consists of determining from this spectrum the number of X-ray counts contained in each peak, identifying the origin of the peak, and converting the number to an areal density by using the expression

$$N(E) = N_i \left(\frac{d\sigma}{d\Omega}\right)_E \rho_x \Omega \epsilon(E) \tag{1}$$

where

$N(E)$	number of X-ray photons of energy E detected
N_i	number of X-ray photons incident on target
$(d\sigma/d\Omega)_E$	cross section for production of X-rays of energy E by exciting X-rays
ρ_x	areal density of atoms producing X-rays of energy E
Ω	solid angle subtended by X-ray detector
$\epsilon(E)$	efficiency of detector for detection of X-rays of energy E

In practice, only the quantity $N(E)$ is determined absolutely. The number N_i is assumed to have the form

$$N_i = K_{Rh} N_{Rh} \tag{2}$$

where N_{Rh} is the number of rhodium (monitor) X-rays detected, so that the areal density may be written

$$\rho_X = \frac{N(E)}{K_{Rh} N_{Rh} \left(\frac{d\sigma}{d\Omega}\right)_E \Omega \epsilon(E)} \tag{3}$$

The overall efficiency function

$$F(E) = \frac{1}{K_{Rh} \left(\frac{d\sigma}{d\Omega}\right)_E \Omega \epsilon(E)} \tag{4}$$

is then determined experimentally by using samples of known areal density of a number of elements. For elements other than those for which $F(E)$ is measured directly, an interpolation is carried out with the assumption that the function varies smoothly with energy. The areal density then, in terms of directly measurable quantities, becomes

$$\rho_X = N(E) \frac{F(E)}{N_{Rh}} \tag{5}$$

Analysis then requires only the determination of the intensity and energy of each line in the X-ray spectrum and the intensity of the line used for normalization N_{Rh}.

DATA ACQUISITION AND ANALYSIS

The entire X-ray fluorescence operation (sample changing, data acquisition, and data analysis) is carried out under the control of a PDP-15 computer. Communication between the computer and the X-ray fluroescence hardware is by means of a CAMAC system (ref. 1). The programmable interface between the computer and the CAMAC system has been described in detail elsewhere (ref. 2). The control panel of the PDP-15 computer, shown in figure 3, allows the input of a number of parameters as well as the selection of a number of different modes of operation, so that a sufficient degree of flexibility is provided to make the system useful for a variety of applications.

Data Acquisition

The two principal control tasks which the PDP-15 must perform are the changing of targets and the control (stop, start, and data transfer) of the 1024-channel analog-to-digital converter (ADC), through which all pulse-height data pass.

A typical data-taking cycle is initiated upon input of the last piece of sample identification data by means of the computer teletype unit. The ADC is turned on, and data transfers from the ADC to the computer memory take place autonomously by way of the CAMAC interface. During data acquisition the pulse-height spectrum is displayed on a video unit, which assures the user that all systems are functioning properly.

By means of push buttons on the computer control panel, data acquisition may be terminated at any time, either by pressing B4, which causes data to be recorded on magnetic tape, or by pushing B2, which ends the run without recording of the data. Data acquisition also terminates, with all accumulated data being recorded on tape, when the elapsed time reaches the preset value as entered (in tenths of seconds) on control panel thumb switch TH2.

Following termination and depending on the mode of operation selected, either another run is started automatically (T15 up) or a halt occurs (T15 down) until the operator chooses to continue by pressing B2. In either case sample changing may or may not be done, depending on the setting of rotary switch R1 on the control panel (sample is changed when R1=2).

The data recorded upon completion of a run include the run identification (a five-character file name and a three-digit file number), the time elapsed during data acquisition, and, of course, the pulse-height spectrum itself.

Because analog-to-digital conversion and data transfers take place at data-channel level, they proceed autonomously once initiated, so that the computer central processing unit is free at main-stream level to proceed with analysis of a previous data set if so instructed.

Data Analysis

Analysis of previous data sets may either be initiated by the operator (press B1) or take place automatically when the computer is operating in an automatic mode (T15 up). Analysis begins with the recalling of two previous runs from tape: one with the data of interest and one that is a background or blank run, which is subtracted to remove events resulting from scattering or impurities in the system or sample-supporting matrix.

When analysis is initiated by an operator, the program requests a file name and file number for each of these data sets. In the automatic mode the program searches for a data set with the same name as that previously analyzed, increases the file number to obtain the next data set, and uses the same background data set. Hence, it is always necessary that at least the first set be analyzed manually.

The first step in the analysis is the determination of the number of events N_{Rh} contained in the monitor (rhodium) peak for both the sample data and background data.

This is accomplished by locating the rhodium peak and then fitting that portion of the spectrum within 20 channels of it with a Gaussian peak plus linear background. The number of events within the Gaussian peak is then the N_{Rh} of equation (5). The amount of background in this region provides the normalization for subtraction of the blank spectrum.

It is assumed that since the blank spectrum results from an unloaded sample of the supporting matrix, it is identical in shape to that encountered in the loaded sample. Consequently, after normalization and smoothing over three channel segments, the two spectra are subtracted channel by channel to arrive at the spectrum to be analyzed.

The recalled spectrum (with background subtracted) then replaces the live display of data currently being accumulated on the video unit. Figure 4 shows the recalled spectrum derived from the data of figure 2. The number of events under each peak, that is, the number of detected X-ray photons of each energy, is determined by a least-squares fitting to each peak of a Gaussian distribution of adjustable height, width, and location. Initial estimates of peak locations may be supplied by the researcher by means of the light pen and video display (T11 down), or they may be located automatically by the program (T11 up). The automatic location procedure is described in appendix A.

Depending on the setting of toggle switches, an energy calibration may be supplied at this time (T16 down). The user may supply through the teletype console the energies of the first two peaks selected with the light pen. For operation in the automatic mode (T16 and T15 up) the program retains the energy calibration from the previous run. Hence, again at least one run must be done manually. The researcher is also asked to supply through the teletype console (unless the computer is operating in the automatic mode) an estimate of the full width at one-half of the maximum (FWHM) of a typical peak.

The fitting routine does each peak in turn from the lowest to the highest energy. When two peaks are found to lie within four times the estimated FWHM, they are fitted simultaneously. Up to five peaks may be treated as a multiplet and done simultaneously. A more detailed description of the fitting procedure is given in appendix B.

Following analysis of each line for energy and intensity, an identification is made by comparing the measured energy with a table of energies and atomic symbols. The elements which are included in this table are listed in table I. If a peak is identified as a K_α line, a check is made to determine whether the K_β line can also be observed. If a line of the proper energy is located, the expected intensity of the K_β line (determined from the intensity of the K_α line) is subtracted from the K_β candidate. The remainder of the peak is tested for other identification. No peak may be identified as an L series X-ray unless both the L_α and L_β lines are observed.

The experimental intensity is converted to an areal density by using the stored efficiency function F(E). This function is assumed (for K series X-rays) to have the form

$$\log \left[\frac{1}{F(E)}\right] = \sum_{i=1}^{7} \frac{P(I)}{E^{(I-1)}} \tag{6}$$

The parameters P(I) are determined by measuring the X-ray yield for a number of samples (approximately 10 to 15) of known areal density and performing a least-squares fit of the function given in equation (6) to that data. The parameters P(I) vary with any change in detector or source geometry and must be redetermined each time such a change occurs. A typical set of parameters P(I) is listed in table II. For heavier elements, which are measured by means of D series X-rays, tabulated efficiencies are used for those elements which the program can identify.

The estimates of errors in the computed areal density are based on the number of counts in each peak, given by

$$N(E) = I(E)\sigma(E)\sqrt{\pi} \tag{7}$$

where I and σ (as defined in appendix B) are the height and width, respectively, of the Gaussian distribution derived from the data. The areal density, given by equation (5), then becomes

$$\rho_X = \frac{I(E)\sigma(E)\sqrt{\pi}F(E)}{N_{Rh}} \tag{8}$$

Therefore, assuming the error in N_{Rh} to be small gives

$$\frac{\delta\rho_X}{\rho_X} = \left|\frac{\delta I(E)}{I(E)}\right| + \left|\frac{\delta\sigma(E)}{\sigma(E)}\right| + \left|\frac{\delta F(E)}{F(E)}\right| \tag{9}$$

or

$$\frac{\delta\rho_X}{\rho_X} = \left|\frac{\delta I(E)}{I(E)}\right| + \left|\frac{\delta\sigma(E)}{\sigma(E)}\right| + \left|\frac{1}{F(E)} \frac{dF(E)}{dE} \delta E\right| \tag{10}$$

The errors in each of the quantities I(E), $\sigma(E)$, and E are derived in the usual way from the residuals of the least-squares fitting process.

Computed areal densities may be modified by an arbitrary multiplicative factor G, which may be entered by means of thumb wheel TH1. The value of G may be between

0 and 9.9999, and its effect is included in all areal densities and errors listed as output. The principal purpose of this factor is to permit the user to convert to a convenient system of units if so desired. If G is set equal to 1, the results are in units of micrograms per square inch.

Output may consist of either one or two pages of teletype output as well as a record of the results on magnetic tape. These options are selected through the control panel toggle switches (p. 1, T14 up; p. 2, T12 up; and tape, T13 up). Samples of the two pages of teletype output are shown in figures 5 and 6. More detailed information on the results of the least-squares fitting process may be obtained by replying "T" (true) when the program inquires, through the teletype console, "Do you wish extra output? Output=." A sample of this augmented output is shown in figure 7. It includes the height, width, and location, as well as the standard deviation of each, for each line fitted by the program.

A flow chart for the entire program is shown in figure 8. A guide to the loading of the program into the PDP-15 is given in appendix C. Appendix D contains the details of the overlay structure necessary in order to fit the program into the available core. Appendix E contains a listing of the FORTRAN sections of the program.

CONCLUDING REMARKS

The program described in this report permits trace element analysis of environmental samples at a rate which would make meaningful monitoring surveys feasible. Specifically, data analysis requires between 3 and 12 minutes, depending on the complexity of the spectrum, while data acquisition requires 5 to 30 minutes, depending on the primary radiation source and associated hardware employed. The program works reliably in an automatic mode; it changes samples, takes data, locates peaks, measures intensities, and identifies elements. As a result samples may be processed without an operator in attendance, which makes around-the-clock analysis possible. In addition, when operated under the supervision of the user, the program is capable of treating a wide variety of nonroutine samples, such as those containing elements not automatically recognized and those containing a number of heavy elements and thus having extremely complex spectra. For these, the program permits the extraction of meaningful data with a minimum amount of effort on the part of the user.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, December 2, 1976,
506-25.

APPENDIX A

AUTOMATIC PEAK LOCATING PROCEDURE

If the operator so desires, initial estimates of peak locations are determined by the program, using a method described by Mariscotti (ref. 3). The method is based on the construction of smoothed second differences of the experimentally measured energy spectrum. The smoothed second difference is defined by

$$S_i(z,m) = \underbrace{\sum_{j=i-m}^{i+m} \cdots \sum_{h=l-m}^{l+m}}_{z} (N_{l+1} - 2N_l + N_{l-1}) \tag{A1}$$

or equivalently

$$S_i(z,m) = \sum_{\text{all } j} C_{ij}(z,m)N_j \tag{A2}$$

where the C_{ij} are weighting factors derived by Mariscotti, given by

$$C_{ij}(z,m) = \sum_{l=i-m}^{i+m} C_{lj}(z-1,m) \tag{A3}$$

and

$$C_{ij}(0,m) = \begin{cases} 0 & \text{if } |j - i| \geq 2 \\ 1 & \text{if } |j - i| = 1 \\ -2 & \text{if } j = i \end{cases} \tag{A4}$$

The values of z and m used here are those found by Mariscotti to be most suitable, namely,

$$z = 5$$

$$m = \frac{(0.6\Gamma - 1)}{2}$$

The resulting function S then vanishes for a linear spectrum and is similar to the second derivative of a Gaussian function if N_j is a Gaussian function. The program assumes that a peak exists in channel i if three conditions are present: First,

$$|S_i| > 1.6\ |\delta S_i| \tag{A5}$$

where δS_i is the standard deviation in the function S_i, approximated by Mariscotti as

$$\delta S_i = \left(\sum C_{ij}^2\right)^{1/2} \delta N_i$$

where δN_i is the standard deviation in the data point N_i. The program assumes

$$\delta N_i = (N_i + B_i)^{1/2}$$

where B_i is the amount of background previously subtracted from channel i of the spectrum. Second,

$$S_i < 0 \tag{A6}$$

Third,

$$S_{i-1} < |S_i| > S_{i+1} \tag{A7}$$

The code does not reject any peak on the basis of its width as suggested by Mariscotti.

APPENDIX B

LEAST-SQUARES FITTING

Once initial estimates of peak locations and widths have been obtained, a least-squares fitting is carried out by using the Gaussian function

$$f(x) = I \exp\left[\frac{(x - x_0)^2}{\sigma^2}\right] \tag{B1}$$

to represent that portion of the spectrum being treated. In the event that several peaks fall within 4FWHM of each other, they are fitted simultaneously as the sum of as many as five Gaussian distributions, each of which has independently variable width.

For initial estimates, x_0 is taken to be the channel i either found by the automatic search routine described in appendix B or supplied through the light pen. The initial estimate of the intensity I is taken to be the number of counts in that i^{th} channel, and σ is calculated from the FWHM input (for a Gaussian distribution, σ = FWHM/1.665).

Fitting is carried out by standard methods. In the event a satisfactory fit cannot be obtained within a specified number of iterations, the peak of highest energy is dropped from the multiplet, and the fitting process is restarted for the remainder of the peaks. In the event that a single peak cannot be satisfactorily fitted, a message to that effect is included on output page 1, and the peak is assigned intensity zero and a location equal to the starting channel number.

APPENDIX C

LOADING THE PROGRAM

As presented in this report the program (entitled SAVLOC) is too large to be loaded directly into a 16K PDP-15 computer. As a result it can only be used as a series of overlays, constructed by using the routine CHAIN (ref. 4). A listing of the execute file, titled SAVLOC, is given in appendix D; the listing includes the resident code as well as the structure of each of the links and a machine constructed map of the file.

At execution time the user must assign handlers to three slots in the device assignment table (dat): dat slot 1, for writing of raw data; dat slot 2, for reading of raw data; and dat slot 3, for writing of results of analysis (fig. 6). Dat slots 1 and 2 are normally assigned the same magnetic tape (dectape) unit, and dat slot 3 is assigned a second dectape unit.

An example of the entire loading sequence is as follows:

```
KMS15 V5C
$A DTE6 1,2/DTE7 3
$E SAVLOC

EXECUTE V4A
FNAME(A) =
```

where the underlined expressions are output from the computer.

APPENDIX D

OVERLAY STRUCTURE AND MEMORY MAP FOR EXECUTE FILE SAVLOC

```
\CHAIN

CHAIN V5B

NAME XCT FILE
>SAVLOC
LIST OPTIONS & PARAMETERS
>PGR,16K,SZ
DEFINE RESIDENT CODE
>SIMUL2,CAMADC,IPK3,PICKER
DESCRIBE LINKS & STRUCTURE
>LK1=REDE,RITE
>LK2=FITALL,FITAL1,GAUSS
>LK3=ECAL1,ECAL2
>LK4=REGOUT,XKYLD
>LK5=LABEL,SORT
>LK6=S,SCAL1,LOCUM
>LK1:LK2:LK3:LK4:LK5:LK6:REED:MAXER:SET1:SUBTR:OUTER:
-MULT:SIT2:ERRO:ENERG:IDENT
>
LINK TABLE
        37225-37636 ØØ412

RESIDENT CODE
SIMUL2  36527-37224 ØØ476
CAMADC  35731-36526 ØØ576
IPK3    3474Ø-3573Ø ØØ771
PICKER  343Ø5-34737 ØØ433
RELAON  34226-343Ø4 ØØØ57
RTIMER  34146-34225 ØØØ6Ø
RECV16  34126-34145 ØØØ22
SEND16  34113-34125 ØØØ13
CAMAC.  33631-34112 ØØ262
PANEL   33477-3363Ø ØØ132
CROSS   33347-33476 ØØ13Ø
HISTO1  33226-33346 ØØ121
PENSW   332ØØ-33225 ØØØ26
VPVECT  33Ø14-33177 ØØ164
VIDEO.  32515-33Ø13 ØØ277
RTIME   3247Ø-32514 ØØØ25
CLOCK.  32372-32467 ØØØ76
BITEST  3235Ø-32371 ØØØ22
GETBLK  32267-32347 ØØØ61
IBOOL   32224-32266 ØØØ43
ZERO    322ØØ-32223 ØØØ24
.SS     3213Ø-32177 ØØØ5Ø
FLOAT   32117-32127 ØØØ11
FLOAT2  32Ø71-32116 ØØØ26
.BB     32Ø24-32Ø7Ø ØØØ45
FIOPS   313Ø2-32Ø23 ØØ522
INTEAE  31171-313Ø1 ØØ111
RELEAE  3ØØ57-3117Ø Ø1112
```

```
OTSEP   27712-27777 00066
.DA     30021-30056 00036
DISPL   30016-30020 00003
ERP     30012-30015 00004
STUFF   27464-27711 00226
MUP     27445-27463 00017
FINOUT  27345-27444 00100

LINK -- LK1
REDE    27277-27344 00046
PITE    27135-27276 00142
RELAOF  27061-27134 00054
BCDEXT  27012-27060 00047
INPUT1  26723-27011 00067
XREAD   26407-26702 00314
PACKER  24306-26406 00101
MOD     26265-26305 00021
BCDIO   23630-26264 02435
BINIO   23403-23627 00225
FILE    23124-23402 00257
.CB     23105-23123 00017

LINK -- LK2
FITALL  26136-27344 01207
FITAL1  26005-26135 00131
GAUSS   25277-26004 00506
MATINV  24736-25276 00341
GENBLK  24663-24735 00053
LOC     24654-24662 00007
ABS     24635-24653 00017
RMNMX   24511-24634 00124
.BC     24415-24510 00074
EXP     24377-24414 00016
.EF     24272-24376 00105
.EC     24220-24271 00052
.RDA    24130-24217 00070
FITALC  24127-24127 00001

LINK -- LK3
ECAL1   27001-27344 00344
ECAL2   26446-27000 00333
INPUT1  26357-26445 00067
XREAD   26043-26356 00314
PACKER  25742-26042 00101
BCDIO   23305-25741 02435
.CB     23266-23304 00017

LINK -- LK4
REGOUT  26470-27344 00655
XKYLD   26343-26467 00125
.BC     26247-26342 00074
.BE     26237-26246 00010
.PR     26201-26236 00036
SQRT    26066-26200 00113
.EE     25703-26065 00163
.EF     25576-25702 00105
.EC     25524-25575 00052
.RDA    25434-25523 00070
```

```
BCDIO   22777-25433 02435
.CB     22760-22776 00017

LINK -- LK5
LABEL   24757-27344 02366
SORT    24621-24756 00136
ABS     24602-24620 00017

LINK -- LK6
S       27117-27344 00226
SCALI   24651-27116 00246
LOCUM   24730-26650 01721
IFIX    24712-24727 00016
ABS     24673-24711 00017
LIMIT   24650-24672 00023
.BC     24554-24647 00074
SQRT    24441-24553 00113
DIFF    24220-24440 00221

LINK -- REED
REED    26775-27344 00350
BCDEXT  26726-26774 00047
INPUTI  26637-26725 00067
XREAD   26323-26636 00314
PACKER  24222-26322 00101
MOD     26201-26221 00021
BCDIO   23544-26200 02435
BINIO   23317-23543 00225
FILE    23040-23316 00257
.CB     23021-23037 00017

LINK -- MAXER
MAXER   27275-27344 00050

LINK -- SETI
SETI    27003-27344 00342

LINK -- SUBTP
SUBTP   26527-27344 00616
INPUTI  26440-26526 00067
XREAD   26124-26437 00314
PACKER  26023-26123 00101
BCDIO   23366-26022 02435
.CB     23347-23365 00017

LINK -- OUTER
OUTER   27041-27344 00304
BCDIO   24404-27040 02435
.CB     24365-24403 00017

LINK -- MULT
MULT    26665-27344 00460
IFIX    26647-26664 00016

LINK -- SIT2
SIT2    26032-27344 01313
BCDIO   23375-26031 02435
.CB     23356-23374 00017
```

```
LINK -- ERRO
ERRO   27Ø33-27344 ØØ312
BCDIO  24376-27Ø32 Ø2435
.CB    24357-24375 ØØØ17

LINK -- ENERG
ENERG  265ØØ-27344 ØØ645

LINK -- IDENT
IDENT  26Ø66-27344 Ø1257
BCDEXT 26Ø17-26Ø65 ØØØ47
PACKER 25716-26Ø14 ØØ1Ø1
ABS    25677-25715 ØØØ17
MOD    25656-25676 ØØØ21
BCDIO  23221-25655 Ø2435
FILE   22742-2322Ø ØØ257
.CB    22723-22741 ØØØ17

BLANK COMMON
.XX    12723-22722 1ØØØØ

CORE REQ'D
       12723-37636 24714


KMS15  V5C
$
```

APPENDIX E

SOURCE LISTING

```
C.....PROGRAM TO SIMULTANEOUSLY TAKE AND ANALYZE DATA
      INTEGER DATAD(4096),GATEX(64,7),ORGN(8),DATUM(1024)
      REAL FNAME(2),LT,LTI
      LOGICAL ON,EXIST,BUTTNS,BITEST
      EXTERNAL ADCON
      COMMON DATAD
      EQUIVALENCE (GATEX(1,1),DATAD(3573)),(DATAD(1),DATUM(1))
      DATA ND,NLAST/1024,1/,
     1     FNAME(1),FNAME(2)/5HSPECT,1H /
      CALL CLKON
      CALL PSTART
2     CALL REDE(FNAME,NEXT,NLAST)
3     I4=IROTOR(4)
      I3=IROTOR(3)
      I2=IROTOR(2)
      EXIST=.TRUE.
      NY=IROTOR(4)+6
      NF=256*I2-255
      NX=I3+7
      CALL ZERO(DATUM)
      CALL CLKON
      CALL VPSTOP
      CALL HISTOI(DATUM,NF,NX,NY,NSAW,1)
      CALL VPSTRT
      CALL ECAMCK
      CALL RTCOFF
      IF(ON) CALL ADCOFF
104   MODE=0
      IRT=ITHUMB(2)
      ILT=0
      IF(NEXT.GT.NLAST) GO TO 100
      CALL ADCSTR(DATAD,GATEX,ORGN,ILT,ON,MODE)
      CALL RTCSTR(IRT,ADCOFF)
      CALL RTIME(TS)
      CALL LAMPON(1)
100   CALL BUTTNS(IB)
      IF(IB.EQ.1) GO TO 101
      IF(IB.EQ.2) GO TO 501
      IF(.NOT.ON) GO TO 96
102   IF(IB.EQ.4) GO TO 98
      CALL TOGGLS(IT)
      IF(BITEST(IT,15).AND.EXIST) GO TO 101
      GO TO 100
101   CALL VPSTOP
      CALL IPK2(EXIST)
      CALL VPSTOP
      CALL HISTOI(DATUM,NF,NX,NY,NSAW,1)
      CALL VPSTRT
99    IF(ON) GO TO 100
96    IF(NEXT.GT.NLAST) GO TO 102
98    CALL RTCOFF
      CALL ADCOFF
      CALL LAMPOF(1)
```

```
      CALL TOGGLS(IT)
      IF(IROTOR(1).EQ.2) CALL RELAON(3)
      CALL RITE(FNAME,NEXT,TS,IRT,ILT,ND)
      NEXT=NEXT+1
      CALL TOGGLS(IT)
      IF(BITEST(IT,15)) GO TO 3
      CALL LAMPON(2)
500   CALL BUTTNS(IB)
      IF(IB.NE.2) GO TO 500
502   CALL LAMPOF(2)
501   CALL TOGGLS(IT)
      IF(BITEST(IT,17)) GO TO 2
      GO TO 3
      END


      SUBROUTINE IPK2(EXIST)
C.....SUBROUTINE TO DO PEAK ANALYSIS AFTER LIGHT PEN SELECTION,
C.....FOR USE WITH SIMUL2, TO TAKE DATA SIMULTANEOUSLY.
C.....USED TO BIULD XCT FILE
      INTEGER LIST(520),DATA(1024),BKGND(1024),GATE(5,30),CH0(30),
     1      CHMAX,DATAD(4096),BEXT,IR(5)
      REAL FNAME(2),PARAM(15),DP(15),POINT(100),BNAME(2),XCH(2),
     1      ELE(30),ENORM(30),BNORMX(30),DR(30)
      LOGICAL EXIST,DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,
     1      UNQUIN,BUTTNS,PENSW,OUTPT,BITEST
      EXTERNAL GAUSS
      COMMON/DISPL/NF,NX,NY
      COMMON/ERR/PHI2,IMIN,ITELL
      COMMON/STUFF/GATE
      COMMON/MUP/DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,UNQUIN,
     1      JPEAK,GAM,IDUB,ITRIP,IQUAD,IQUIN
      COMMON/FINOUT/ELE,A1,A2
      COMMON DATAD
      EQUIVALENCE (DATA(1),DATAD(1025)),(BKGND(1),DATAD(2049)),
     1      (LIST(1),DATAD(3073)),(CH0(1),DATAD(4021)),
     2      (POINT(1),DATAD(3593)),(ENORM(1),DATAD(3800)),
     3      (BNORMX(1),DATAD(3861)),(DR(1),DATAD(3921)),
     4      (IR(1),DATAD(4051)),(PARAM(1),DATAD(4056))
      CALL LAMPON(3)
      CALL REED(1,FNAME,NEXT,EXIST)
      IF(EXIST) GO TO 3
      CALL LAMPOF(3)
      RETURN
3     NEXT=NEXT+1
2     CALL MAXER(DATA,CHMAX)
      CALL SETI(CHMAX,DATA,POINT)
      ITELL=1
      CALL FITALL(GAUSS,POINT,PARAM,DP,Q,LIST)
      CALL OUTER(PARAM,LIST,ANORM,ABNORM,CHMAX)
      CALL REED(2,BNAME,BEXT,EXIST)
      CALL MAXER(BKGND,CHMAX)
      CALL SETI(CHMAX,BKGND,POINT)
      CALL FITALL(GAUSS,POINT,PARAM,DP,Q,LIST)
      ITELL=0
      CALL SUBTR(ANORM,ABNORM,CHMAX,OUTPT,GAM)
      CALL PICKER
      CALL TOGGLS(IT)
      IF(BITEST(IT,10)) CALL VPSTOP
```

```
      IF(BITEST(IT,11)) CALL LOCUM
      CALL LAMPOF(7)
      IF(BITEST(IT,16)) GO TO 87
      DO 88 J=1,2
      CALL ECAL1(J)
      CALL FITALL(GAUSS,POINT,PARAM,DP,Q,LIST)
      XCH(J)=FLOAT(CH0(J))+PARAM(2)-4.
88    CONTINUE
87    CALL ECAL2(XCH,A1,A2)
C.....BEGIN FITTING PEAKS
80    CALL LAMPOF(7)
      DO 89 J=3,JPEAK
81    CALL MULT(IMAX,IMIN,J)
      CALL SIT2(IMAX,IMIN,J,ANORM)
      CALL FITALL(GAUSS,POINT,PARAM,DP,Q,LIST)
      NPARAM=LIST(6)
      DO 85 I=1,NPARAM,3
      IF(PARAM(I+1).GT.FLOAT(IMAX-IMIN+5)) LIST(4)=7
      IF(PARAM(I+1).LT.-5.) LIST(4)=8
85    CONTINUE
      IF(LIST(4).EQ.0) GO TO 86
      CALL ERRO(J,IMIN,DP)
      IF(.NOT.DOUBLE) GO TO 86
      GO TO 81
86    CALL ENERG(IMIN,A1,A2,J,DP)
      CALL LABEL(J,JPEAK)
      CALL REGOUT(OUTPT,ANORM,IMIN,DP,Q,J)
89    CONTINUE
      CALL IDENT(ELE,ENORM,BNORMX,DB,JPEAK,FNAME,NEXT)
      CALL LAMPOF(3)
      CALL VPSTRT
      RETURN
      END


      SUBROUTINE PICKER
      INTEGER GATE(5,30),CH0(30),DATA(1024),DATAD(4096),OFF,X,Y,
     1     CHMAX
      LOGICAL DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,UNQUIN,
     1     BUTTNS,PENSW,BITEST
      COMMON/STUFF/GATE
      COMMON/MUP/DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,
     1     UNQUIN,JPEAK,GAM,IDUB,ITRIP,IQUAD,IQUIN
      COMMON/DISPL/NF,NX,NY
      COMMON DATAD
      EQUIVALENCE (DATA(1),DATAD(1025)),(CH0(1),DATAD(4021))
      DATA OFF/-1/
100   FORMAT(1X,2HOK)
      ICHK=0
6     IS1=NX
      IS2=NY
      IS3=NF
      CALL SCAL1(NX,NY,NF)
      CALL TOGGLS(IT)
      IF(ICHK.NE.1) GO TO 66
      IF((IS1.NE.NX).OR.(IS2.NE.NY).OR.(IS3.NE.NF)) GO TO 66
      GO TO 65
66    CALL VPSTOP
      CALL HISTO1(DATA,NF,NX,NY,NHIT,1)
```

```
      CALL VPVECT(GATE,30,IHIT,2)
      CALL VPSTRT
65    IF(BITEST(IT,16)) RETURN
      IF(ICHK.EQ.1) GO TO 12
      IF(ICHK.EQ.-1) GO TO 13
7     CALL ZERO(GATE)
      CALL ZERO(CH0)
      JPEAK=0
      CALL CROSS(X,Y,OFF,3)
8     JPEAK=JPEAK+1
12    IF(PENSW(.TRUE.)) GO TO 9
      ICHK=1
      GO TO 6
9     GATE(1,JPEAK)=NHIT
      CH0(JPEAK)=NHIT
      GATE(2,JPEAK)=DATA(NHIT)
      GATE(3,JPEAK)=0
      GATE(4,JPEAK)=128
      GATE(5,JPEAK)=-10
      ICHK=-1
      GO TO 6
13    CALL LAMPON(7)
10    CALL BUTTNS(IB)
      IF(IB.EQ.128) GO TO 11
      IF(IB.EQ.32) GO TO 14
      IF(IB.EQ.64) RETURN
      GO TO 10
14    JPEAK=JPEAK-1
11    CALL LAMPOF(7)
      GO TO 8
      END


      SUBROUTINE REDE(FNAME,NEXT,NLAST)
      REAL FNAME(2)
      CALL INPUTI(FNAME(1),5HFNAME,6)
      CALL INPUTI(NEXT,4HNEXT,1)
      CALL INPUTI(NLAST,5HNLAST,1)
      RETURN
      END


      FUNCTION S(ICHNL)
      INTEGER DATA(1024),DATAD(4096)
      REAL C(36),B(36)
      COMMON/DIFF/C,B,M
      COMMON DATAD
      EQUIVALENCE (DATA(1),DATAD(1025))
      M1=5*M+2
      X=C(1)*FLOAT(DATA(ICHNL))
      DO 41 J=1,M1
      I1=ICHNL+J
      I2=ICHNL-J
      J1=J+1
      IF(I1.GT.1024) GO TO 42
      IF(I2.LT.0) GO TO 42
      X=X+C(J1)*FLOAT(DATA(I1))+C(J1)*FLOAT(DATA(I2))
      GO TO 41
```

```
42      X=0
41      CONTINUE
        S=X
        RETURN


        SUBROUTINE RITE(FNAME,NEXT,TS,IRT,ILT,ND)
        REAL FNAME(2)
        INTEGER DATAD(4096)
        COMMON DATAD
        CALL BCDEXT(FNAME,NEXT)
        CALL ENTER(2,FNAME)
        WRITE(2) TS,IRT,ILT,ND,(DATAD(I),I=1,ND)
        CALL RELAOF(3)
        WRITE(6,106) FNAME,TS,IRT,ILT
106     FORMAT(1X,2A5,5X,F10.5,I6,3X,I6)
        CALL CLOSE(2)
        RETURN
        END


        SUBROUTINE GAUSS(R,X,P,DP,V,KK,J)
C.....SUBROUTINE FOR US EWITH FITALL TO FIT UP TO 4
C..... GAUSSIAN PAKS WITH OR WITHOUT BACKGROUND
        REAL X(100),P(15),DP(15)
        INTEGER BKGND(1024),DATA(1024),DATAD(4096)
        COMMON/ERR/PHI2,ILOWER,ITELL
        COMMON DATAD
        EQUIVALENCE (DATA(1),DATAD(1025)),(BKGND(1),DATAD(2049))
        NP=KK/3
        IF(KK-3*NP.EQ.0) GO TO 1
        R=X(J)-P(1)*FLOAT(J)-P(2)
        DP(1)=-FLOAT(J)
        DP(2)=-1.
        N=3
        GO TO 3
1       N=1
        R=X(J)
3       DO 2 I=N,KK,3
        TP=P(I)*EXP(-(FLOAT(J)-P(I+1))**2/P(I+2)**2)
        R=R-TP
        DP(I)=-TP/P(I)
        DP(I+1)=-TP*2.*(FLOAT(J)-P(I+1))/P(I+2)**2
        DP(I+2)=DP(I+1)*(FLOAT(J)-P(I+1))/P(I+2)
2       CONTINUE
        IERR=ILOWER+J-3
        IF(ITELL.EQ.0) V=(FLOAT(BKGND(IERR)+1))
        IF(ITELL.EQ.1) V=1.
        RETURN
        END


        SUBROUTINE ECAL1(J)
C.......SUBROUTINE CALL ED BY IPK2 TO DETERMINE ENERGY CALIBRATION.
        INTEGER CH0(30),LIST(520),DATAD(4096),DATA(1024)
        LOGICAL BITEST
        REAL PARAM(15),DP(15),POINT(100),X(2)
        COMMON DATAD
        COMMON/ERR/PHI2,IMIN,ITELL
```

```
      EQUIVALENCE (DATA(1),DATAD(1025)),(LIST(1),DATAD(3073)),
     1      (CH0(1),DATAD(4021)),(POINT(1),DATAD(3593)),(PARAM(1)
     1,DATAD(4056))
      CALL TOGGLS(IT)
      IF(BITEST(IT,16)) RETURN
      LIST(1)=7
      LIST(2)=3
      LIST(3)=15
      LIST(4)=0
      LIST(5)=3
      LIST(6)=3
      IK=CH0(J)
      PARAM(1)=FLOAT(DATA(IK))
      PARAM(2)=4.
      PARAM(3)=4.5/1.665
      IMIN=CH0(J)-3
      DO 3 I=1,7
      IC=CH0(J)-4+I
      POINT(I)=FLOAT(DATA(IC))
3     CONTINUE
      RETURN
      END


      SUBROUTINE ECAL2(X,A1,A2)
C.......SUBROUTINE TO DO ENERGY CALIBRAION WITH IPK2
      REAL X(2)
      LOGICAL BITEST
      CALL TOGGLS(IT)
      IF(BITEST(IT,16)) GO TO 99
      WRITE(6,100)
      CALL INPUT1(E1,2HE1,2)
      CALL INPUT1(E2,2HE2,2)
      A1=(E1-E2)/(X(1)-X(2))
      A2=E1-A1*X(1)
99    G=FLOAT(ITHUMB(1))/10000.
      WRITE(6,103) G
103   FORMAT(1X,2HG=,F6.4)
      IF(BITEST(IT,14)) WRITE(6,101)
      IF(BITEST(IT,14)) WRITE(6,102)
      RETURN
101   FORMAT(///,6H  PEAK,9H  CHANNEL,12H    STRENGTH,
     1      10H    ENERGY,16H   CONCENTRATION,10H      ERROR,
     2      8H   ELEM )
102   FORMAT(39X,14H(UGRAMS/IN**2))
100   FORMAT(1X,25HENERGIES FOR CALIBRATION:)
      END


      SUBROUTINE REGOUT(OUTPT,ANORM,IMIN,DP,Q,J)
      INTEGER LIST(520),DATAD(4096)
      REAL PARAM(15),ENORM(30),ELE(30),DP(15),BNORMX(30),DB(30)
      LOGICAL OUTPT,BITEST
      COMMON DATAD
      COMMON/FINOUT/ELE,A1,A2
      EQUIVALENCE (LIST(1),DATAD(3073)),(ENORM(1),DATAD(3800)),
     1      (BNORMX(1),DATAD(3861)),(DB(1),DATAD(3921)),
     2      (PARAM(1),DATAD(4056))
      NPARAM=LIST(6)
```

```
      DO 92 I=1,NPARAM
      DP(I)=SQRT((Q)*DP(I))
92    CONTINUE
      DO 91 I=1,NPARAM,3
      IP2=I+1
      ANORMX=PARAM(I)*PARAM(I+2)*1.7725/ANORM
      BNORMX(J)=ANORMX*XKYLD(ENORM(J))*FLOAT(ITHUMB(1))/10000.
      DB(J)=(DP(I)/PARAM(I))+(DP(I+2)/PARAM(I+2))
      DY=A1*DP(IP2)/XKYLD(ENORM(J))
      DY=DY*(XKYLD(ENORM(J))-XKYLD(ENORM(J)+0.02))*50.
      DB(J)=(DB(J)+DY)*BNORMX(J)
      CALL TOGGLS(IT)
      IF(BITEST(IT,14))WRITE(6,125) J,PARAM(IP2),ANORMX,ENORM(J),
     1     BNORMX(J),DB(J),ELE(J)
      J=J+1
91    CONTINUE
      J=J-1
      IF(OUTPT) WRITE(6,109) (I,PARAM(I),DP(I),I=1,NPARAM)
109   FORMAT(1X,2HP(,I2,2H)=,F12.3,5X,6HDELTA=,F12.3)
125   FORMAT(1X,I4,2X,F8.3,F11.6,3X,F8.3,3X,E11.5,2X,E11.5,3X,1A5)
      RETURN
      END


C.....FUNCTION XKYLD, TO TRANSFORM RATIO TO RHODIUM TO
     1     NANOGRAMS/CM**2
      FUNCTION XKYLD(E)
      REAL P(8)
      DATA P(1),P(2),P(3),P(4),P(5),P(6),P(7)/
     1     1.2347,30.8516,-306.138,1980.348,-6477.202,
     2     10661.801,-7106.853/
1     AMT=0.
      DO 2 I=1,7
      AMT=AMT+P(I)/(E)**(I-1)
2     CONTINUE
      XKYLD=10.**AMT
      RETURN
      END


      SUBROUTINE LABEL(J,JPEAK)
      REAL ENORM(30),ELE(30),K,MN,CRMN,NI,NIBET,MO,BNORMX(30),DB(30)
      INTEGER CH0(30),LIST(520),DATAD(4096)
      LOGICAL CHECK
      COMMON/FINOUT/ELE,A1,A2
      COMMON DATAD
      EQUIVALENCE (CH0(1),DATAD(4021)),(LIST(1),DATAD(3073)),
     1     (ENORM(1),DATAD(3800)),(BNORMX(1),DATAD(3861)),
     2     (DB(1),DATAD(3921))
      DATA CL,AR,K,SC,CA,TI/2HCL,2HAR,1HK,2HSC,2HCA,2HTI/,
     1     V,CR,MN,FE,CO,NI/1HV,2HCR,2HMN,0HFE,2HCO,2HNI/,
     2     CU,ZN,TLL,PBL,SE,BR/2HCU,2HZN,3HTLL,3HPBL,2HSE,2HBR/,
     3     THL,RB,SR,ZR,MO/3HTHL,2HRB,2HSR,2HZR,2HMO/,
     4     CASC,SCTI,TIV,VCR,CRMN/5HCA-SC,5HSC-TI,4HTI-V,
     5     4HV-CR,5HCR-MN/,
     6     FEBET,NIBET,CUBET/5HFEBET,5HNIBET,5HCUBET/,
     7     ZNBET,BRBET,UNKNW/5HZNBET,5HBRBET,5H-----/
      NELE=(LIST(6)/3)+J
      DO 2 I=J,NELE
```

```
      E=ENORM(I)
      ELE(I)=UNKNW
      IF(ABS(E-2.62).LE.Ø.Ø6)  ELE(I)=CL
      IF(ABS(E-2.95).LE.Ø.Ø6)  ELE(I)=AR
      IF(ABS(E-3.31).LE.Ø.Ø6)  ELE(I)=K
2Ø    IF(ABS(E-3.69).LE.Ø.Ø6)  ELE(I)=CA
      IF(ABS(E-4.Ø9).LE.Ø.Ø6) CALL SORT(CHECK,3.69,JPEAK)
      IF(.NOT. CHECK.AND.(ABS(E-4.Ø9).LT.Ø.Ø6)) ELE(I)=SC
      IF(CHECK) ELE(I)=CASC
      CHECK=.FALSE.
      IF(ABS(E-4.51).LE.Ø.Ø6) CALL SORT(CHECK,4.Ø9,JPEAK)
      IF(.NOT.CHECK.AND.(ABS(E-4.51).LT.Ø.Ø6)) ELE(I)=TI
      IF(CHECK) ELE(I)=SCTI
      CHECK=.FALSE.
      IF(ABS(E-4.95).LE.Ø.Ø6) CALL SORT(CHECK,4.51,JPEAK)
      IF(.NOT.CHECK.AND.(ABS(E-4.95).LE.Ø.Ø6)) ELE(I)=V
      IF(CHECK) ELE(I)=TIV
      CHECK=.FALSE.
      IF(ABS(E-5.41).LE.Ø.Ø6) CALL SORT(CHECK,4.95,JPEAK)
      IF(.NOT.CHECK.AND.(ABS(E-5.41).LE.Ø.Ø6)) ELE(I)=CR
      IF(CHECK) ELE(I)=VCR
      CHECK=.FALSE.
      IF(ABS(E-5.9ØØ).LE.Ø.Ø6) CALL SORT(CHECK,5.41,JPEAK)
      IF(.NOT.CHECK.AND.(ABS(E-5.9Ø).LE.Ø.Ø6)) ELE(I)=MN
      IF(CHECK) ELE(I)=CRMN
      CHECK=.FALSE.
28    IF(ABS(E-6.4Ø).LE.Ø.Ø6)  ELE(I)=FE
      IF(ABS(E-6.93).LE.Ø.Ø6)  ELE(I)=CO
      IF(ABS(E-7.Ø6).LE.Ø.Ø6)  ELE(I)=FEBET
      IF(ABS(E-7.48) .LE.Ø.Ø6)  ELE(I)=NI
      IF(ABS(E-8.Ø5).LE.Ø.Ø6)  ELE(I)=CU
      IF(ABS(E-8.26).LE.Ø.Ø6) CALL SORT(CHECK,7.48,JPEAK)
      IF(CHECK) ELE(I)=NIBET
      CHECK=.FALSE.
      IF(ABS(E-8.63).LE.Ø.Ø6)  ELE(I)=ZN
      IF(ABS(E-8.9Ø).LE.Ø.Ø6)  CALL SORT(CHECK,8.Ø5,JPEAK)
      IF(CHECK) ELE(I)=CUBET
      CHECK=.FALSE.
      IF(ABS(E-9.57).LE.Ø.Ø6)  CALL SORT(CHECK,8.63,JPEAK)
      IF(CHECK) ELE(I)=ZNBET
      CHECK=.FALSE.
      IF(ABS(E-1Ø.25).LE.Ø.Ø6)  CALL SORT(CHECK,12.21,JPEAK)
      IF(CHECK) ELE(I)=TLL
      CHECK=.FALSE.
      IF(ABS(E-1Ø.55).LE.Ø.Ø6)  CALL SORT(CHECK,12.61,JPEAK)
      IF(CHECK) ELE(I)=PBL
      CHECK=.FALSE.
34    IF(ABS(E-11.22).LE.Ø.Ø6)  ELE(I)=SE
      IF(ABS(E-11.89).LE.Ø.Ø6)  ELE(I)=BR
      IF(ABS(E-12.21).LE.Ø.Ø6)  CALL SORT(CHECK,1Ø.27,JPEAK)
      IF(CHECK) ELE(I)=TLL
      CHECK=.FALSE.
      IF(ABS(E-12.61).LE.Ø.Ø6)  CALL SORT(CHECK,1Ø.55,JPEAK)
      IF(CHECK) ELE(I)=PBL
      CHECK=.FALSE.
      IF(ABS(E-12.9Ø).LE.Ø.1Ø)  CALL SORT(CHECK,16.2Ø,JPEAK)
      IF(CHECK) ELE(I)=THL
      CHECK=.FALSE.
      IF(ABS(E-13.29).LE.Ø.Ø6)  CALL SORT(CHECK,11.92,JPEAK)
```

```
      IF(CHECK) ELE(I)=RBBET
      CHECK=.FALSE.
      IF(ABS(E-13.39).LE.0.06) ELE(I)=RB
      IF(ABS(E-14.16).LE.0.06) ELE(I)=SR
      IF(ABS(E-15.77).LE.0.06) ELE(I)=ZR
      IF(ABS(E-16.20).LE.0.10) CALL SORT(CHECK,12.90,JPEAK)
      IF(CHECK) ELE(I)=THL
      CHECK=.FALSE.
42    IF(ABS(E-17.47).LE.0.06) ELE(I)=MO
2     CONTINUE
      RETURN
      END


      SUBROUTINE SORT(CHECK,X,JPEAK)
      INTEGER CH0(30),DATAD(4096)
      LOGICAL CHECK
      REAL ELE(30),BNORMX(30),DB(30)
      COMMON/FINOUT/ELE,A1,A2
      COMMON DATAD
      EQUIVALENCE (CH0(1),DATAD(4021)),(BNORMX(1),DATAD(3861)),
     1      (DB(1),DATAD(3921))
      CHECK=.FALSE.
      DO 2 I=1,JPEAK
      E=A1*FLOAT(CH0(I))+A2
      IF(ABS(E-X).LE.0.10) GO TO 5
      GO TO 2
5     I=JPEAK
      CHECK=.TRUE.
2     CONTINUE
      RETURN
      END


      SUBROUTINE SCAL1(NX,NY,NF)
C.....SUBROUTINE TO CHANGE SCALES ON DISPLAY,
C........FOR USE WITH IPICK
      INTEGER GATE(5,30),DATA(1024),CH0(30),DATAD(4096),CHMAX
      COMMON/STUFF/GATE
      COMMON DATAD
      EQUIVALENCE (DATA(1),DATAD(1025)),(CH0(1),DATAD(4021))
      I2=IROTOR(2)
      NF=LIMIT(1,256*I2-255,1024)
      I3=IROTOP(3)
      NX=LIMIT(8,I3+7,12)
      I4=IROTOR(4)
      NY=LIMIT(8,I4+6,18)
      DO 10 I=1,30
      L1=CH0(I)
      IF(L1.EQ.0) L1=1
      GATE(1,I)=LIMIT(0,(CH0(I)-NF)*8/2**I3,1023)
      GATE(2,I)=DATA(L1)/(2**I4)*16
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE REED(ITYPE,FNAME,NEXT,EXIST)
C........SUBROUTINE TO READ DATA FROM TAPE FOR USE WITH IPK2
C............AND SIMUL2
      INTEGER DATA(1024),DATAD(4096),BKGND(1024)
      REAL FNAME(2)
      LOGICAL EXIST,BITEST
      COMMON DATAD
      EQUIVALENCE (DATA(1),DATAD(1025)),(BKGND(1),DATAD(2049))
      CALL TOGGLS(IT)
      IF(BITEST(IT,15)) GO TO 3
1     CALL INPUTI(FNAME(1),5HDNAME,6)
      CALL INPUTI(NEXT,4HNEXT,1)
3     CALL BCDEXT(FNAME,NEXT)
      CALL FSTAT(1,FNAME,EXIST)
      IF(EXIST) GO TO 2
      WRITE(6,100) FNAME
      IF(BITEST(IT,15))RETURN
      GO TO 1
2     IF(ITYPE.EQ.1)CALL ZERO(DATA)
      IF(ITYPE.EQ.1) WRITE(6,101) FNAME
101   FORMAT(1H1,12H RUN NUMBER ,2A5)
      IF(ITYPE.EQ.2) CALL ZERO(BKGND)
      CALL SEEK(1,FNAME)
      IF(ITYPE.EQ.1)READ(1) TS,IRT,ILT,ND,(DATA(I),I=1,1024)
      IF(ITYPE.EQ.2) READ(1) TS,IRT,ILT,ND,(BKGND(I),I=1,1024)
      RETURN
100   FORMAT(1X,6H0FILE ,2A5,10HNOT FOUND.)
      END


      SUBROUTINE MAXER(DATA,CHMAX)
      INTEGER DATA(1024),CHMAX
      CHMAX=450
      DO 400 I=450,1024
      IF(DATA(I).GT.DATA(CHMAX))CHMAX=I
400   CONTINUE
      RETURN
      END


      SUBROUTINE SET1(CHMAX,IDAT,POINT)
C.......ROUTINE TO SET UP CALL TO FITALL TO FIND HEIGHT OF
C            NORMALIZATION PEAK
      INTEGER CHMAX,IDAT(1024),DATAD(4096),LIST(520)
      REAL POINT(100),PARAM(15)
      COMMON DATAD
      EQUIVALENCE (LIST(1),DATAD(3073)),(PARAM(1),DATAD(4056))
      IMAX=CHMAX+20
      IMIN=CHMAX-20
      BG2=IDAT(IMAX)
      BG1=IDAT(IMIN)
      DO 200 I=1,41
      ICO=IMIN+I-1
      POINT(I)=IDAT(ICO)
200   CONTINUE
      PARAM(2)=(BG1)
      PARAM(1)=(BG2-BG1)/40.
      PAPAM(5)=6./1.665
```

```
      PARAM(3)=FLOAT(IDAT(CHMAX))-PARAM(2)-PARAM(1)*20.
      PAPAM(4)=FLOAT(CHMAX-IMIN+1)
      LIST(1)=41
      LIST(2)=5
      LIST(3)=15
      LIST(4)=0
      LIST(5)=5
      LIST(6)=5
      RETURN
      END


      SUBROUTINE OUTER(PARAM,LIST,ANORM,ABNORM,CHMAX)
      INTEGER CHMAX,LIST(520)
      REAL PARAM(15)
      PARAM(4)=PARAM(4)+FLOAT(CHMAX-21)
      IF(LIST(4).NE.0) WRITE(6,111) LIST(4)
111   FORMAT(1X,11HERROR,TYPE ,I2)
      ANORM=PARAM(3)*PARAM(5)*1.7725
      ABNORM=40.*PARAM(2)+800.*PARAM(1)
      WRITE(6,123) PARAM(4),ANORM
      WRITE(6,124) ABNORM
124   FORMAT(1X,15HBACKGROUND HAS ,F10.3,8H COUNTS.)
123   FORMAT(/,1X,29HNORMALIZATION PEAK IN CHANNE
     1L,F8.2,4H HAS,F10.2,8H COUNTS.)
      RETURN
      END


      SUBROUTINE SUBTR(ANORM,ABNORM,CHMAX,OUTPT,GAM)
      INTEGER DATA(1024),CHMAX,DATAD(4096),BKGND(1024)
      REAL PARAM(15)
      LOGICAL OUTPT,BITEST
      COMMON DATAD
      EQUIVALENCE (DATA(1),DATAD(1025)),(BKGND(1),DATAD(2049)),
     1      (PARAM(1),DATAD(4056))
      PARAM(4)=PARAM(4)+FLOAT(CHMAX-21)
      CNORM=PARAM(3)*PARAM(5)*1.7725
      CBNORM=40.*PARAM(2)+800.*PARAM(1)
      WRITE(6,123) PARAM(4),CNORM
      WRITE(6,124) CBNORM
124   FORMAT(1X,15HBACKGROUND HAS ,F10.3,8H COUNTS.)
      RAT=ABNORM/CBNORM
      DO 5 I=3,1023
      IS=I-2
      DATA(I)=FLOAT(DATA(I))-RAT*FLOAT(BKGND(I-1)+BKGND(I)+
     1      BKGND(I+1))/3.
      BKGND(IS)=FLOAT(DATA(I))+2.*RAT*FLOAT(BKGND(I-1)+
     1      BKGND(I)+BKGND(I+1))/3.
      IF(DATA(I).LT.0) DATA(I)=0
5     CONTINUE
      BKGND(1023)=0
      BKGND(1024)=0
      CALL TOGGLS(IT)
      IF(BITEST(IT,15)) RETURN
      WRITE(6,105)
      CALL INPUTI(OUTPT,5HOUTPT,4)
      CALL INPUTI(GAM,5HGAMMA,2)
```

```
123     FORMAT(//,1X,29HNORMALIZATION PEAK IN CHANNE
        1L,F8.2,4H HAS,F10.2,8H COUNTS.)
105     FORMAT(1X,25HDO YOU WISH EXTRA OUTPUT?)
        RETURN
        END


        SUBROUTINE MULT(IMAX,IMIN,J)
        INTEGER CH0(30),DATAD(4096)
        LOGICAL DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,UNQUIN
        COMMON/MUP/ DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,
        1     UNQUIN,JPEAK,
        1      GAM,IDUB,ITRIP,IQUAD,IQUIN
        COMMON DATAD
        EQUIVALENCE (CH0(1),DATAD(4021))
        DOUBLE=.FALSE.
        TRIPLE=.FALSE.
        QUAD=.FALSE.
        QUIN=.FALSE.
        I0=CH0(J)
        IMAX=CH0(J)+IFIX(GAM)
        IMIN=CH0(J)-IFIX(GAM)
        IF(UNDOUB) GO TO 82
        IF(J.EQ.JPEAK) GO TO 82
        IF(CH0(J+1).LT.(I0+IFIX(4.*GAM))) DOUBLE=.TRUE.
        IDUB=CH0(J+1)
        IF(UNTRIP) GO TO 82
        IF(J.EQ.JPEAK-1) GO TO 82
        IF(CH0(J+2).LT.(IDUB+IFIX(4.*GAM))) TRIPLE=.TRUE.
        TRIPLE=DOUBLE.AND.TRIPLE
        ITRIP=CH0(J+2)
        IF(UNQUAD) GO TO 82
        IF(J.EQ.JPEAK-2) GO TO 82
        IF(CH0(J+3).LT.(ITRIP+IFIX(4.*GAM))) QUAD=.TRUE.
        QUAD=TRIPLE.AND.QUAD
        IQUAD=CH0(J+3)
        IF(UNQUIN) GO TO 82
        IF(J.EQ.JPEAK-3) GO TO 82
        IF(CH0(J+4).LT.(IQUAD+IFIX(4.*GAM))) QUIN=.TRUE.
        QUIN=QUIN.AND.QUAD
        IQUIN=CH0(J+4)
82      UNDOUB=.FALSE.
        UNTRIP=.FALSE.
        UNQUAD=.FALSE.
        UNQUIN=.FALSE.
        IF(DOUBLE) IMAX=IDUB+IFIX(GAM)
        IF(TRIPLE) IMAX=ITRIP+IFIX(GAM)
        IF(QUAD) IMAX=IQUAD+IFIX(GAM)
        IF(QUIN) IMAX=IQUIN+IFIX(GAM)
        RETURN
        END


        SUBROUTINE SIT2(IMAX,IMIN,J,ANORM)
C.....TO PREPARE FNIAL FIT TO DATA
        REAL PARAM(15),POINT(100)
        INTEGER DATA(1024),LIST(520),CH0(30),IR(5),DATAD(4096)
        LOGICAL DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,UNQUIN,
        1      BITEST
```

```
      COMMON/MUP/DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,
     1     UNQUIN,JPEAK,GAM,IDUB,ITRIP,IQUAD,IQUIN
      COMMON DATAD
      EQUIVALENCE (DATA(1),DATAD(1025)),(LIST(1),DATAD(3073)),
     1     (CH0(1),DATAD(4021)),(POINT(1),DATAD(3593)),
     2     (IR(1),DATAD(4051)),(PARAM(1),DATAD(4056))
      ITOT=IMAX-IMIN+1
      DO 84 I=1,ITOT
      IDOWN=IMIN+I-1
      POINT(I)=FLOAT(DATA(IDOWN))
84    CONTINUE
      I0=CH0(J)
      PARAM(14)=FLOAT(IQUIN-IMIN+1)
      PARAM(15)=GAM/1.665
      PARAM(13)=FLOAT(DATA(IQUIN))
      PARAM(11)=FLOAT(IQUAD-IMIN+1)
      PARAM(12)=GAM/1.665
      PARAM(10)=FLOAT(DATA(IQUAD))
85    PARAM(7)=FLOAT(DATA(ITRIP))
      PARAM(8)=FLOAT(ITRIP-IMIN+1)
      PARAM(9)=GAM/1.665
86    PARAM(4)=FLOAT(DATA(IDUB))
      PARAM(5)=FLOAT(IDUB-IMIN+1)
      PARAM(6)=GAM/1.665
87    PARAM(1)=FLOAT(DATA(I0))
      PARAM(2)=FLOAT(I0-IMIN+1)
      PARAM(3)=GAM/1.665
      LIST(1)=ITOT
      LIST(2)=3
      LIST(3)=15
      LIST(4)=0
      LIST(5)=12
      LIST(6)=3
      IF(.NOT.DOUBLE) GO TO 88
      LIST(2)=6
      LIST(6)=6
      IF(.NOT.TRIPLE) GO TO 88
      LIST(2)=9
      LIST(6)=9
      IF(.NOT.QUAD) GO TO 88
      LIST(2)=12
      LIST(6)=12
      IF(.NOT.QUIN) GO TO 88
      LIST(2)=15
      LIST(6)=15
      LIST(5)=15
88    CONTINUE
      NPARAM=LIST(6)
      CALL ZERO(IR)
      ISUM=0
      K=0
      DO 95 I=1,NPARAM,3
      K=K+1
      IF(PARAM(I).LT.0.0000300*ANORM) IR(K)=1
      ISUM=ISUM+IR(K)
95    CONTINUE
      LIST(2)=LIST(2)-3*ISUM
      LIST(5)=LIST(5)-ISUM*3
      LIST(6)=LIST(6)-3*ISUM
```

```
      I1=1
      DO 97 I=1,K
      IF(IR(I).NE.0) GO TO 97
96    PARAM(I1)=PARAM(3*I-2)
      PARAM(I1+1)=PARAM(3*I-1)
      PARAM(I1+2)=PARAM(3*I)
98    I1=I1+3
97    CONTINUE
      CALL TOGGLS(IT)
      IF(DOUBLE.AND.(.NOT.TRIPLE).AND.BITEST(IT,14)) WRITE(6,115)
      IF(TRIPLE.AND.(.NOT.QUAD).AND.BITEST(IT,14)) WRITE(6,116)
      IF(QUAD.AND.(.NOT.QUIN).AND.BITEST(IT,14)) WRITE(6,118)
      IF(QUIN.AND.BITEST(IT,14)) WRITE(6,119)
      RETURN
115   FORMAT(1X,8H DOUBLET)
116   FORMAT(1X,8H TRIPLET)
118   FORMAT(1X,11H QUADRUPLET)
119   FORMAT(1X,11H QUINTUPLET)
      END


      SUBROUTINE IDENT(ELE,ENORM,BNORMX,DB,JPEAK,FNAME,NEXT)
C.......SUBROUTINE TO MAKE FINAL IDENTIFICATION OF ELEMENTS,
C.......AND MAKE CORRECTIONS FOR BETA PEAKS.
      REAL ELE(30),ENORM(30),BNORMX(30),DB(30),FNAME(2),
     1     ES1(24),ES2(24),ATOB(24),NAM(24),EFF(4)
      LOGICAL BITEST,BETA
      DATA ES1(1),ES2(1),ATOB(1),NAM(1)/2.62,2.81,0.,2HCL/,
     1    ES1(2),ES2(2),ATOB(2),NAM(2)/2.96,3.19,0.,2HAR/,
     2    ES1(3),ES2(3),ATOB(3),NAM(3)/3.313,3.589,0.114,1HK/,
     3    ES1(4),ES2(4),ATOB(4),NAM(4)/3.691,4.012,0.0909,2HCA/,
     4    ES1(5),ES2(5),ATOB(5),NAM(5)/4.090,4.460,0.105,2HSC/,
     5    ES1(6),ES2(6),ATOB(6),NAM(6)/4.510,4.931,0.105,2HTI/,
     6    ES1(7),ES2(7),ATOB(7),NAM(7)/4.952,5.427,0.105,1HV/
      DATA ES1(8),ES2(8),ATOB(8),NAM(8)/5.414,5.946,0.114,2HCR/,
     1    ES1(9),ES2(9),ATOB(9),NAM(9)/5.898,6.490,0.108,2HMN/,
     2    ES1(10),ES2(10),ATOB(10),NAM(10)/6.403,7.057,0.106,2HFE/,
     3    ES1(11),ES2(11),ATOB(11),NAM(11)/6.930,7.649,0.118,2HCO/,
     4    ES1(12),ES2(12),ATOB(12),NAM(12)/7.477,8.264,0.105,2HNI/,
     5    ES1(13),ES2(13),ATOB(13),NAM(13)/8.047,8.904,0.113,2HCU/
      DATA ES1(14),ES2(14),ATOB(14),NAM(14)/8.638,9.571,0.117,2HZN/,
     1    ES1(15),ES2(15),ATOB(15),NAM(15)/10.543,11.725,0.083,2HAS/,
     2    ES1(16),ES2(16),ATOB(16),NAM(16)/11.221,12.495,0.13,2HSE/,
     3    ES1(17),ES2(17),ATOB(17),NAM(17)/11.923,13.290,0.136,2HBR/,
     4    ES1(18),ES2(18),ATOB(18),NAM(18)/14.164,15.834,0.151,2HSR/,
     5    ES1(19),ES2(19),ATOB(19),NAM(19)/15.774,17.666,0.172,2HZR/
      DATA ES1(20),ES2(20),ATOB(20),NAM(20)/17.478,19.607,0.18,2HMO/,
     1    ES1(21),ES2(21),ATOB(21),NAM(21)/8.396,9.670,0.775,1HW/,
     2    ES1(22),ES2(22),ATOB(22),NAM(22)/10.266,12.210,0.672,2HTL/,
     3    ES1(23),ES2(23),ATOB(23),NAM(23)/10.549,12.611,0.645,2HPB/,
     4    ES1(24),ES2(24),ATOB(24),NAM(24)/12.966,16.200,0.553,2HTH/
      DATA EPS,UNKNW/0.060,2H--/
      DATA EFF(1),EFF(2),EFF(3),EFF(4)/1.775,1.66,1.66,2.00/
      MEXT=NEXT-1
      CALL BCDEXT(FNAME,MEXT)
      DO 4 I=3,JPEAK
      ELE(I)=UNKNW
      DO 2 J=1,24
      IF(ABS(ENORM(I)-ES1(J)).LT.EPS) GO TO 1
```

```
      GO TO 2
1     BETA=.FALSE.
      DO 3 K=I,JPEAK
      IF(ABS(ENORM(K)-ES2(J)).GT.0.090) GO TO 3
      IF(BNORMX(K).GT.0.3*BNORMX(I)) BETA=.TRUE.
      BNORMX(K)=BNORMX(K)-ATOB(J)*BNORMX(I)
      IF(BNORMX(K).LT.0.) BNORMX(K)=0.
3     CONTINUE
      IF(J.LT.21) ELE(I)=NAM(J)
      IF(J.GE.21 .AND. BETA) GO TO 6
      GO TO 2
6     ELE(I)=NAM(J)
      BNORMX(I)=BNORMX(I)*EFF(J-20)
      DB(I)=DB(I)*EFF(H-20)
2     CONTINUE
4     CONTINUE
      CALL TOGGLS(IT)
      IF(.NOT.BITEST(IT,13))  GO TO 5
      CALL ENTER(3,FNAME)
      WRITE(3,100) (ELE(I),BNORMX(I),DB(I),I=3,JPEAK)
100   FORMAT(1X,1A5,2E11.5)
      CALL CLOSE(3)
5     CALL TOGGLS(IT)
      IF(.NOT.BITEST(IT,12)) RETURN
       WRITE(6,103) FNAME
C.......WRITE(6,104) G
C.......FORMAT(1X,2HG=,F6.4)
      WRITE(6,101)
      WRITE(6,102) (ELE(I),BNORMX(I),DB(I),I=3,JPEAK)
      RETURN
101   FORMAT(1X,7HELEMENT,2X,13HCONCENTRATION,9H     ERROR)
102   FORMAT(4X,1A5,E12.6,E12.6)
103   FORMAT(1H1,5X,2A5)
      END


      SUBROUTINE ERRO(J,IMIN,DP)
      REAL ELE(30),BNORMX(30),DP(15),PARAM(15)
      INTEGER LIST(520),DATAD(4096),IR(5),CH0(30)
      LOGICAL DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,UNQUIN,
      1     BITEST
      COMMON /MUP/DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,
      1     UNQUIN,JPEAK,GAM,IDUB,ITRIP,IQUAD
      COMMON DATAD
      COMMON/FINOUT/ELE,A1,A2
      EQUIVALENCE (LIST(1),DATAD(3073)),(BNORMX(1),DATAD(3861)),
      1     (DB(1),DATAD(3921)),(IR(1),DATAD(4051)),(DATAD(4021),
      2     CH0(1)),(PARAM(1),DATAD(4056))
      DATA RNONE/4HNONE/
      CALL TOGGLS(IT)
      IF(BITEST(IT,14))WRITE(6,111) LIST(4)
111   FORMAT(11H ERROR,TYPE,I2)
      IF(.NOT.DOUBLE) GO TO 89
      IF(.NOT.TRIPLE) GO TO 90
      IF(.NOT.QUAD) GO TO 91
      IF(.NOT.QUIN) GO TO 92
      UNQUIN=.TRUE.
      RETURN
```

```
92      UNQUAD=.TRUE.
        RETURN
91      UNTRIP=.TRUE.
        RETURN
90      UNDOUB=.TRUE.
        RETURN
89      PARAM(1)=0.
        PARAM(3)=0.
        PARAM(2)=FLOAT(CH0(J)-IMIN+1)
        DP(1)=0.
        DP(2)=0.
        DP(3)=0.
        RETURN
        END


        SUBROUTINE ENERG(IMIN,A1,A2,J,DP)
        REAL PARAM(15),ENORM(30),DP(15)
        INTEGER DATAD(4096),LIST(520),IR(5),CH0(30)
        COMMON DATAD
        EQUIVALENCE (DATAD(3800),ENORM(1)),(LIST(1),DATAD(3073)),
        1      (CH0(1),DATAD(4021)),(IR(1),DATAD(4051)),
        2      (PARAM(1),DATAD(4056))
        K=J
        ISUM=0
        DO 2 I=1,5
        IF(IR(I).EQ.2) GO TO 2
        ISUM=ISUM+IR(I)
2       CONTINUE
        K1=0
        NPPEAK=LIST(6)/3
        NPEAK=NPPEAK+ISUM
        DO 3 I=1,NPEAK
        MPEAK=NPEAK-I+1
        M3PEAK=3*MPEAK
        IF(IR(MPEAK).EQ.0) GO TO 4
        JMPEAK=J+MPEAK-1
        PARAM(M3PEAK)=0.
        DP(M3PEAK)=0.
        PARAM(M3PEAK-1)=FLOAT(CH0(JMPEAK))-FLOAT(IMIN-1)
        DP(M3PEAK-1)=0.
        PARAM(M3PEAK-2)=0.
        DP(M3PEAK-2)=0.
        GO TO 3
4       N3PEAK=3*(NPPEAK-K1)
        PARAM(M3PEAK)=PARAM(N3PEAK)
        DP(M3PEAK)=DP(N3PEAK)
        PARAM(M3PEAK-1)=PARAM(N3PEAK-1)
        DP(M3PEAK-1)=DP(N3PEAK-1)
        PARAM(M3PEAK-2)=PARAM(N3PEAK-2)
        DP(M3PEAK-2)=DP(N3PEAK-2)
        K1=K1+1
3       CONTINUE
        LIST(6)=LIST(6)+3*ISUM
        NPARAM=LIST(6)
        DO 5 I=1,NPARAM,3
        IP2=I+1
        PARAM(IP2)=PARAM(IP2)+FLOAT(IMIN-1)
        ENORM(K)=A1*PARAM(IP2)+A2
```

```
      K=K+1
5     CONTINUE
      RETURN
      END


      SUBROUTINE LOCUM
C.....TO LOCATE PEAKS IN X-RAY FLUORESCENT ENERGY SPECTRA..
C.....TO BE USED WITH IPK3, AS PART OF AN EXECUTE FILE..SAVLOC
      INTEGER DATA(1024),GATE(5,30),CH0(30),DATAD(4096),CHMAX,
     1      CH1,CH2,CH3,CH4,CH5,BKGND(1024),GAT1(140)
      REAL C(36),B(36),PHI2,PHI
      LOGICAL DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,
     1      QUIN,UNQUIN
      COMMON/STUFF/GATE
      COMMON/MUP/DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,
     1      QUIN,UNQUIN,JPEAK,GAM,IDUB,ITRIP,IQUAD,IQUIN
      COMMON DATAD
      COMMON/DIFF/C,B,M
      EQUIVALENCE (DATA(1),DATAD(1025)),(CH0(1),DATAD(4021)),
     1      (BKGND(1),DATAD(2049)),(GAT1(1),GATE(1,3))
      CALL ZERO(GAT1)
      NHMAX=470
      M=IFIX((0.6*GAM-1.)/2.)
      IF(M.EQ.0) M=1
C.....COMPUTE COEFFICIENT C(J)
      CALL ZERO(C)
      CALL ZERO(B)
      C(1)=-0.2
      C(2)=0.1
      C(3)=0.
      DO 31 L=1,5
      DO 30 J=1,36
      B(J)=C(J)
      DO 30 K=1,M
      K1=J+K
      K2=J-K
      IF(J-K.LE.0) K2=K-J+2
      B(J)=B(J)+C(K1)+C(K2)
      IF(K1.GT.36) B(J)=0
30    CONTINUE
      DO 31 J=1,36
      C(J)=B(J)
31    CONTINUE
C.....COMPUTE GENERALIZED SECOND DIFFERENCE, S
C.....COMPUTE STANDARD DEVIATIONS IN S
      PHI=C(1)**2
      M1=5*M+2
      DO 50 J=2,M1
      PHI=PHI+2.*C(J)**2
50    CONTINUE
      PHI2=SQRT(PHI)
C.....COMPUTE CRITERIA FOR PEAK LOCATION
C.....N1=EXPECTED NUMBER OF CHANNELS IN NEGATIVE PEAK
      N1=IFIX(1.22*(GAM+0.5))
      N1MAX=N1+4
      N1MIN=N1-2
C.....BEGIN TO SEARCH FOR PEAKS
      CH5=25
```

```
      JPEAK=2
58    JPEAK=JPEAK+1
59    ICHNL=CH5
60    IF(ICHNL.EQ.NHMAX) GO TO 70
      ICHNL=ICHNL+1
      IF(S(ICHNL).GE.-1.6*PHI2*SQRT(FLOAT(BKGND(ICHNL-2)+1))) GO TO 60
      GO TO 62
61    IF(ICHNL.EQ.NHMAX) GO TO 70
      ICHNL=ICHNL+1
62    IF(S(ICHNL+1).LT.S(ICHNL).OR.S(ICHNL-1).LT.S(ICHNL)) GO TO 61
      IF(S(ICHNL).LT.-1.6*PHI2*SQRT(FLOAT(BKGND(ICHNL-2)+1))) GO TO 62
      CH5=ICHNL
      GO TO 59
625   CH4=ICHNL
      GO TO 64
63    IF(ICHNL.EQ.NHMAX) GO TO 70
      ICHNL=ICHNL+1
64    IF(S(ICHNL).LE.0.8*S(CH4)) GO TO 63
      CH5=ICHNL-1
      ICHNL=CH4
65    IF(ICHNL.EQ.1) GO TO 59
      ICHNL=ICHNL-1
      IF(S(ICHNL).LT.0.) GO TO 65
      CH3=ICHNL+1
      GO TO 67
66    IF(ICHNL.EQ.1) GO TO 59
      ICHNL=ICHNL-1
67    IF(S(ICHNL).LT.PHI2*SQPT(FLOAT(BKGND(ICHNL-2)+1))) GO TO 66
      CH2=ICHNL
      GO TO 69
68    IF(ICHNL.EQ.1) GO TO 59
      ICHNL=ICHNL-1
69    IF(S(ICHNL).GT.PHI2*SQRT(FLOAT(BKGND(ICHNL-2)+1))) GO TO 68
      CH1=ICHNL+1
C.....COMPUTE EXPECTED WIDTHS
      N2MIN=IFIX((PHI2*SQRT(FLOAT(DATA(CH4)+1))/ABS(S(CH4)))
     1     *0.5*FLOAT(N1MIN)+0.5)
      N2MAX=IFIX((PHI2*SQRT(FLOAT(DATA(CH4)+1))/ABS(S(CH4)))
     1     *0.5*FLOAT(N1MAX)+0.5)
      N3MIN=IFIX(FLOAT(N1MIN)*(1.-2.*(PHI2*SQRT(FLOAT(DATA(CH4)+1))/
     1     ABS(S(CH4))))+0.5)
      N3MAX=IFIX(FLOAT(N1MAX)*(1.-2.*(PHI2*SQRT(FLOAT(DATA(CH4)+1))/
     1     ABS(S(CH4))))+0.5)
C.......IF(CH5-CH3+1.LT.N1MIN) GO TO 59
C.......IF(CH5-CH3+1.GT.N1MAX) GO TO 59
C.......IF(N2MAX.EQ.0) GO TO 72
C.......IF(CH3-CH2-1.GT.N2MAX) GO TO 59
      GO TO 73
72    IF(CH3-CH2-1.GT.1) CONTINUE
73    IF(CH2-CH1+1.LT.N3MIN) CONTINUE
      CH0(JPEAK)=CH4
      GATE(1,JPEAK)=CH4
      GATE(2,JPEAK)=DATA(CH4)
      GATE(3,JPEAK)=0
      GATE(4,JPEAK)=128
      GATE(5,JPEAK)=-10
      CALL SCALI(NX,NY,NF)
      GO TO 58
70    JPEAK=JPEAK-1
      RETURN
```

REFERENCES

1. Costrell, Louis, ed.: CAMAC: A Modular Instrumentation System for Data Handling. TID-25875, Atomic Energy Commission, 1972.

2. Bercaw, Robert W.; Fessler, Theodore E.; and Arnold, Jeffrey M.: A Progammable Computer Interface for CAMAC. NASA TN D-7148, 1973.

3. Mariscotti, M. A.: A Method for Automatic Identification of Peaks in the Presence of Background and Its Application to Spectrum Analysis. Nuclear Instruments and Methods., vol. 50, no. 2, May 1967, pp. 309-320.

4. CHAIN AND EXECUTE Utility Programs. DEC-15-YWZA-DNZ, Digital Equipment Corp., 1970.

TABLE I. - ELEMENTS RECOGNIZED BY SAVLOC

(a) K series

Element	K_{α} energy, keV
Chlorine	2.62
Argon	2.96
Potassium	3.313
Calcium	3.691
Scandium	4.090
Titanium	4.510
Vanadium	4.952
Chromium	5.414
Manganese	5.898
Iron	6.403
Cobalt	6.930
Nickel	7.477
Copper	8.047
Zinc	8.638
Arsenic	10.543
Selenium	11.221
Bromine	11.923
Strontium	14.164
Zirconium	15.774
Molybdenum	17.478

(b) L series

Element	L_{α} energy, keV	L_{β} energy, keV
Tungsten	8.396	9.670
Thallium	10.266	12.210
Lead	10.549	12.611
Thorium	12.966	16.200

TABLE II. - PARAMETERS USED TO DESCRIBE X-RAY EFFICIENCY FUNCTION F(E)

I	P(I)
1	0.855
2	54.874
3	-705.322
4	5499.011
5	-22895.21
6	48343.26
7	-40357.17

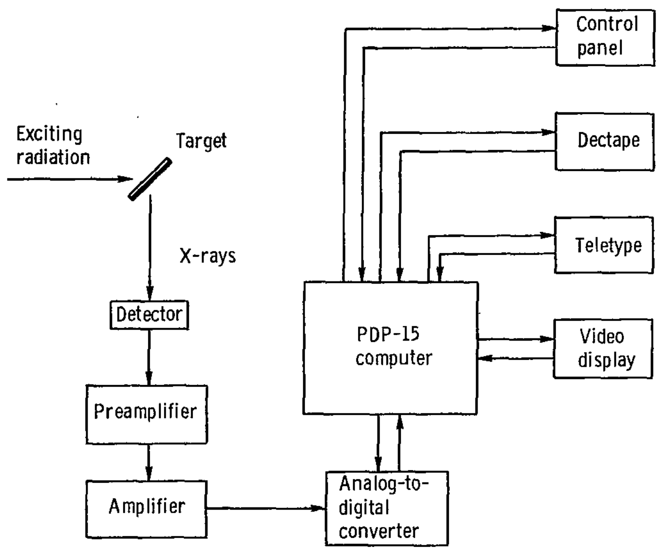

Figure 1. - X-ray fluorescence facility.

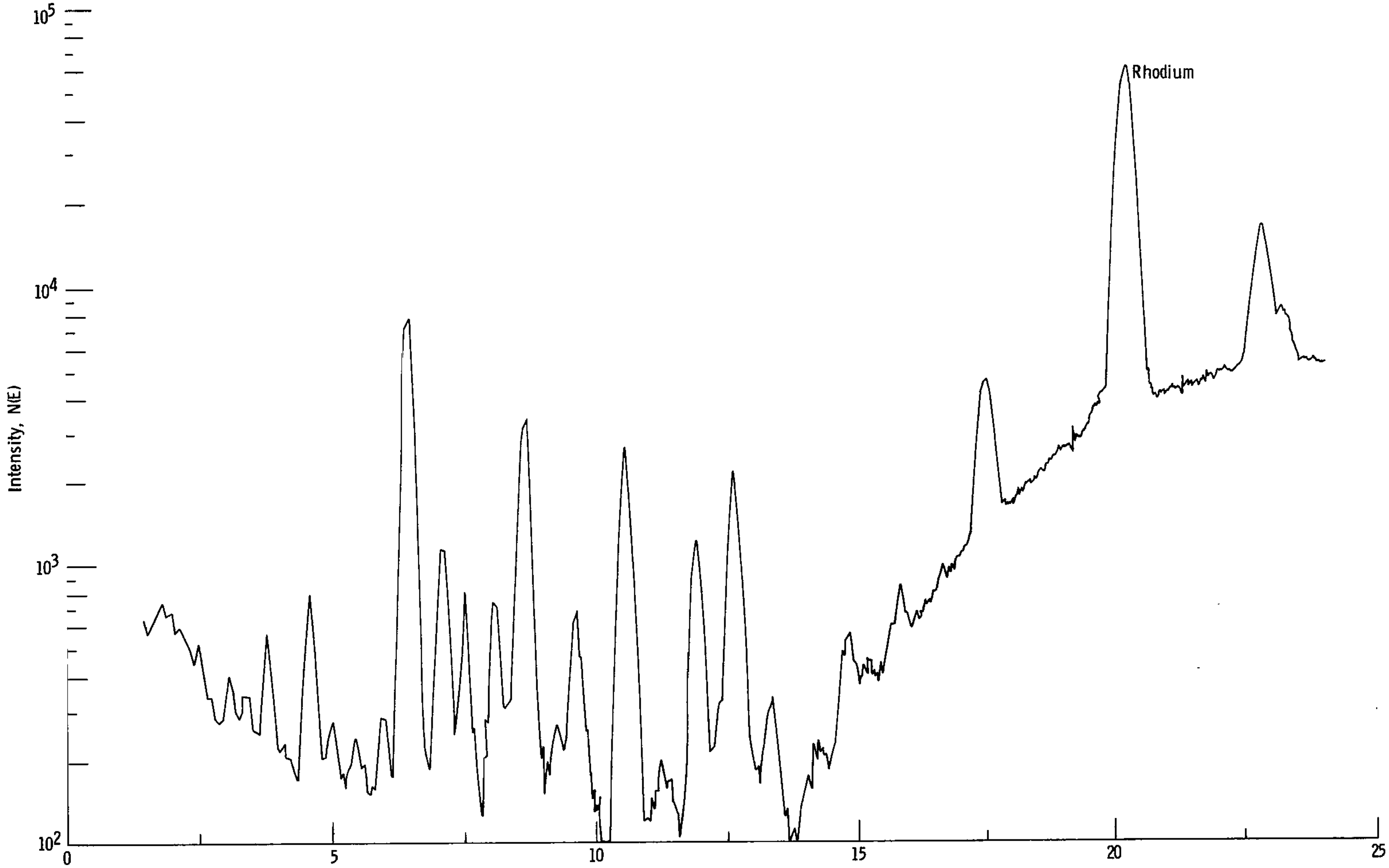

Figure 2. - Typical X-ray spectrum.

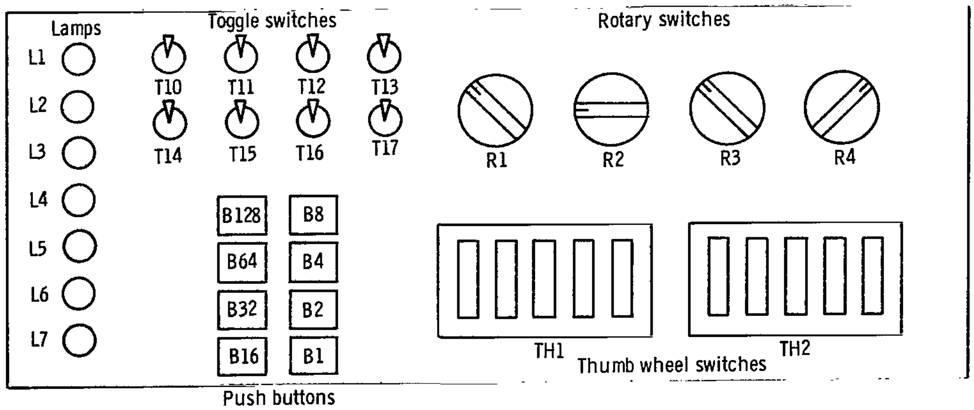

Figure 3. - Functional control panel.

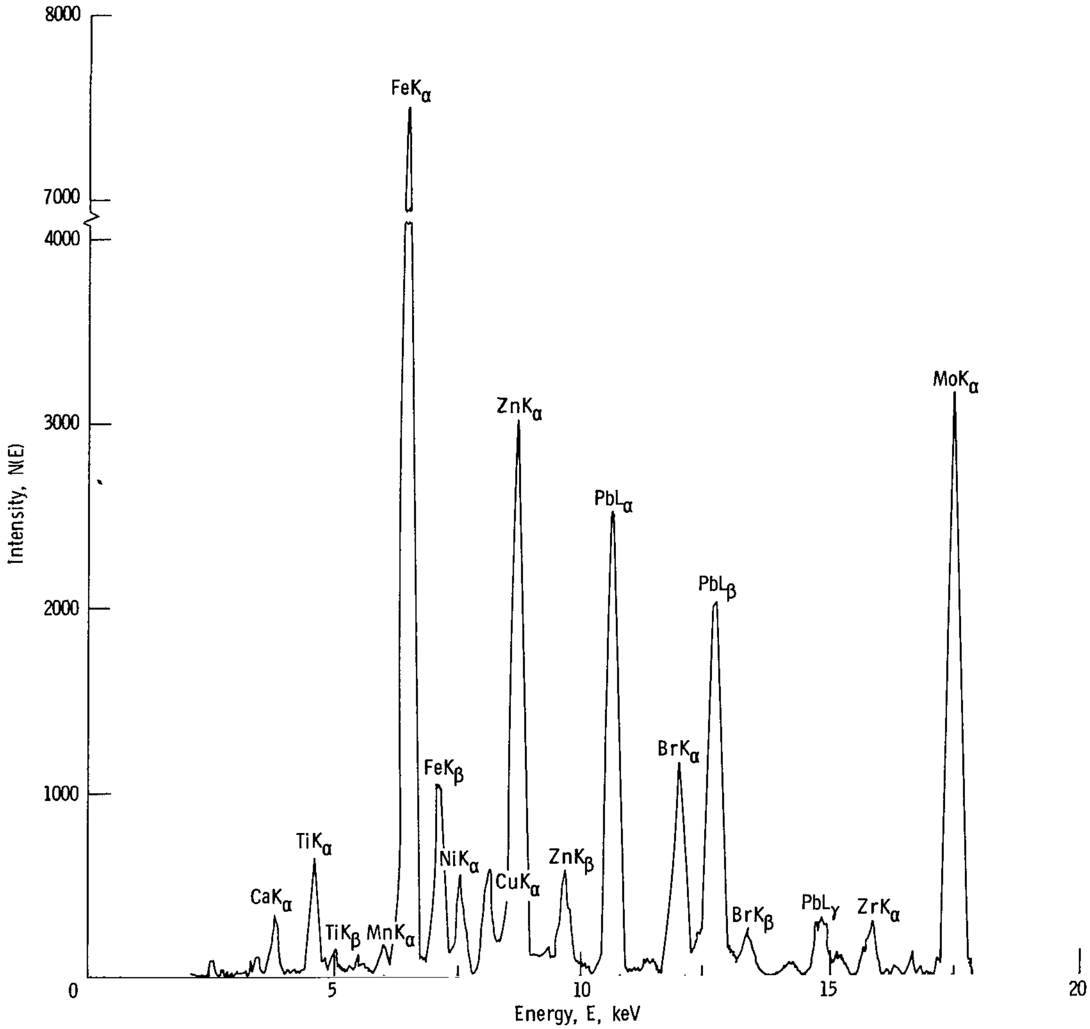

Figure 4. - Typical X-ray spectrum after background subtraction.

```
RUN NUMBER SEPT 011

NORMALIZATION PEAK IN CHANNEL  500.93 HAS 526878.78 COUNTS.
BACKGROUND HAS 152552.379 COUNTS.
DNAME(A) =SEPT
NEXT (I) =0

NORMALIZATION PEAK IN CHANNEL  501.48 HAS 648537.67 COUNTS.
BACKGROUND HAS  56075.779 COUNTS.
DO YOU WISH EXTRA OUTPUT?
OUTPT(L) =F
GAMMA(R) =4.
ENERGIES FOR CALIBRATION:
E1   (R) =6.403
E2   (R) =12.611
G=1.0000
```

PEAK	CHANNEL	STRENGTH	ENERGY	CONCENTRATION (UGRAMS/IN**2)	ERROR	ELEM
3	70.035	.000837	3.701	.29701+02	.19374+01	CA
4	91.404	.004320	4.516	.49822+02	.30408+01	TI
DOUBLET						
5	128.004	.000599	5.912	.22555+01	.78361+00	MN
6	140.871	.077698	6.403	.22115+03	.24637+01	FE
QUADRUPLET						
7	157.800	.010517	7.048	.22002+02	.15192+01	FEBET
8	168.606	.001316	7.461	.23297+01	.94050+00	NI
9	183.969	.004440	8.047	.64054+01	.82467+00	CU
10	199.256	.032763	8.630	.39711+02	.13505+01	ZN
11	224.572	.004366	9.595	.41648+01	.61117+00	ZNBET
12	249.423	.029359	10.543	.23081+02	.55729+00	PBL
13	285.057	.012955	11.902	.80868+01	.28080+00	BR
14	303.631	.025692	12.611	.14433+02	.34989+00	PBL
15	321.212	.001912	13.282	.97893+00	.19976+00	BRBET
DOUBLET						
16	359.355	.003763	14.737	.16030+01	.17434+00	-----
17	368.690	.001194	15.093	.48783+00	.15608+00	-----
18	385.632	.003646	15.739	.13832+01	.25890+00	ZR
19	407.321	.000938	16.566	.32513+00	.71051-01	-----
20	429.751	.051486	17.422	.16323+02	.29195+00	MO

Figure 5. - Sample of first page of teletype output.

SEPT 011

ELEMENT	CONCENTRATION	ERROR
CA	.297007+02	.193737+01
TI	.498224+02	.304084+01
MN	.225552+01	.783605+00
FE	.220910+03	.246371+01
--	.000000+00	.151919+01
NI	.232971+01	.940501+00
CU	.640538+01	.824675+00
ZN	.397109+02	.135047+01
--	.000000+00	.611174+00
PB	.341605+02	.824787+00
BR	.808675+01	.280804+00
--	.000000+00	.349895+00
--	.000000+00	.199760+00
--	.160305+01	.174340+00
--	.487833+00	.156084+00
ZR	.138316+01	.258902+00
--	.325127+00	.710506-01
MO	.163229+02	.291951+00

Figure 6. - Sample of second page of teletype output.

```
 RUN NUMBER SEPT 011

NORMALIZATION PEAK IN CHANNEL  500.93 HAS 526878.78 COUNTS.
BACKGROUND HAS 152552.379 COUNTS.
DNAME(A) =SEPT
NEXT (I) =0


NORMALIZATION PEAK IN CHANNEL  501.48 HAS 648537.67 COUNTS.
BACKGROUND HAS  56075.779 COUNTS.
DO YOU WISH EXTRA OUTPUT?
OUTPT(L) =T
GAMMA(R) =4.
ENERGIES FOR CALIBRATION:
E1    (R) =6.403
E2    (R) =12.611
G=1.0000


 PEAK  CHANNEL    STRENGTH     ENERGY   CONCENTRATION      ERROR    ELEM
                                         (UGRAMS/IN**2)
   3    70.035    .000837      3.701      .29701+02     .19374+01    CA
  P( 1)=     213.043      DELTA=       6.546
 P( 2)=      70.035      DELTA=        .029
 P( 3)=       1.167      DELTA=        .038
   4    91.404    .004320      4.516      .49822+02     .30408+01    TI
  P( 1)=     509.417      DELTA=      14.783
 P( 2)=      91.404      DELTA=        .055
 P( 3)=       2.521      DELTA=        .075
  DOUBLET
   5   128.004    .000599      5.912      .22555+01     .78361+00    MN
   6   140.871    .077698      6.403      .22115+03     .24637+01    FE
  P( 1)=      98.433      DELTA=      15.671
 P( 2)=     128.004      DELTA=        .235
 P( 3)=       1.807      DELTA=        .330
 P( 4)=    7328.194      DELTA=      44.861
 P( 5)=     140.871      DELTA=        .012
 P( 6)=       3.152      DELTA=        .015
  QUADRUPLET
   7   157.800    .010517      7.048      .22002+02     .15192+01    FEBET
   8   168.606    .001316      7.461      .23297+01     .94050+00    NI
   9   183.969    .004440      8.047      .64054+01     .82467+00    CU
  10   199.256    .032763      8.630      .39711+02     .13505+01    ZN
  P( 1)=     959.873      DELTA=      32.626
 P( 2)=     157.800      DELTA=        .079
 P( 3)=       3.257      DELTA=        .110
 P( 4)=   P 211.486      DELTA=      41.329
 P( 5)=     168.606      DELTA=        .280
 P( 6)=       1.849      DELTA=        .378
 P( 7)=     437.471      DELTA=      28.286
 P( 8)=     183.969      DELTA=        .142
 P( 9)=       3.017      DELTA=        .188
 P(10)=    2772.324      DELTA=      48.293
 P(11)=     199.256      DELTA=        .042
 P(12)=       3.513      DELTA=        .057
  11   224.572    .004366      9.595      .41648+01     .61117+00    ZNBET
  P( 1)=     426.654      DELTA=      29.386
 P( 2)=     224.572      DELTA=        .157
 P( 3)=       3.042      DELTA=        .233
  12   249.423    .029359     10.543      .23081+02     .55729+00    PBL
  P( 1)=    2379.978      DELTA=      27.044
 P( 2)=     249.423      DELTA=        .029
 P( 3)=       3.667      DELTA=        .046
  13   285.057    .012955     11.902      .80868+01     .28080+00    BR
  P( 1)=    1016.039      DELTA=      16.254
 P( 2)=     285.057      DELTA=        .042
 P( 3)=       3.790      DELTA=        .070
  14   303.631    .025692     12.611      .14433+02     .34989+00    PBL
  P( 1)=    1891.706      DELTA=      20.430
 P( 2)=     303.631      DELTA=        .031
 P( 3)=       4.037      DELTA=        .054
  15   321.212    .001912     13.282      .97893+00     .19976+00    BRBET
  P( 1)=     165.805      DELTA=      14.989
 P( 2)=     321.212      DELTA=        .245
 P( 3)=       3.428      DELTA=        .385
  DOUBLET
  16   359.355    .003763     14.737      .16030+01     .17434+00    -----
  17   368.690    .001194     15.093      .48783+00     .15608+00    -----
  P( 1)=     294.605      DELTA=      13.175
 P( 2)=     359.355      DELTA=        .150
 P( 3)=       3.797      DELTA=        .241
 P( 4)=      99.216      DELTA=      13.019
 P( 5)=     368.690      DELTA=        .429
 P( 6)=       3.579      DELTA=        .669
  18   385.632    .003646     15.739      .13832+01     .25890+00    ZR
  P( 1)=     242.082      DELTA=      17.146
 P( 2)=     385.632      DELTA=        .266
 P( 3)=       4.477      DELTA=        .516
  19   407.321    .000938     16.566      .32513+00     .71051-01    -----
  P( 1)=     146.899      DELTA=      14.928
 P( 2)=     407.321      DELTA=        .157
 P( 3)=       1.898      DELTA=        .221
  20   429.751    .051486     17.422      .16323+02     .29195+00    MO
  P( 1)=    3371.071      DELTA=      23.320
 P( 2)=     429.751      DELTA=        .025
 P( 3)=       4.540      DELTA=        .049
```

Figure 7. - Sample of augmented page 1 teletype output.

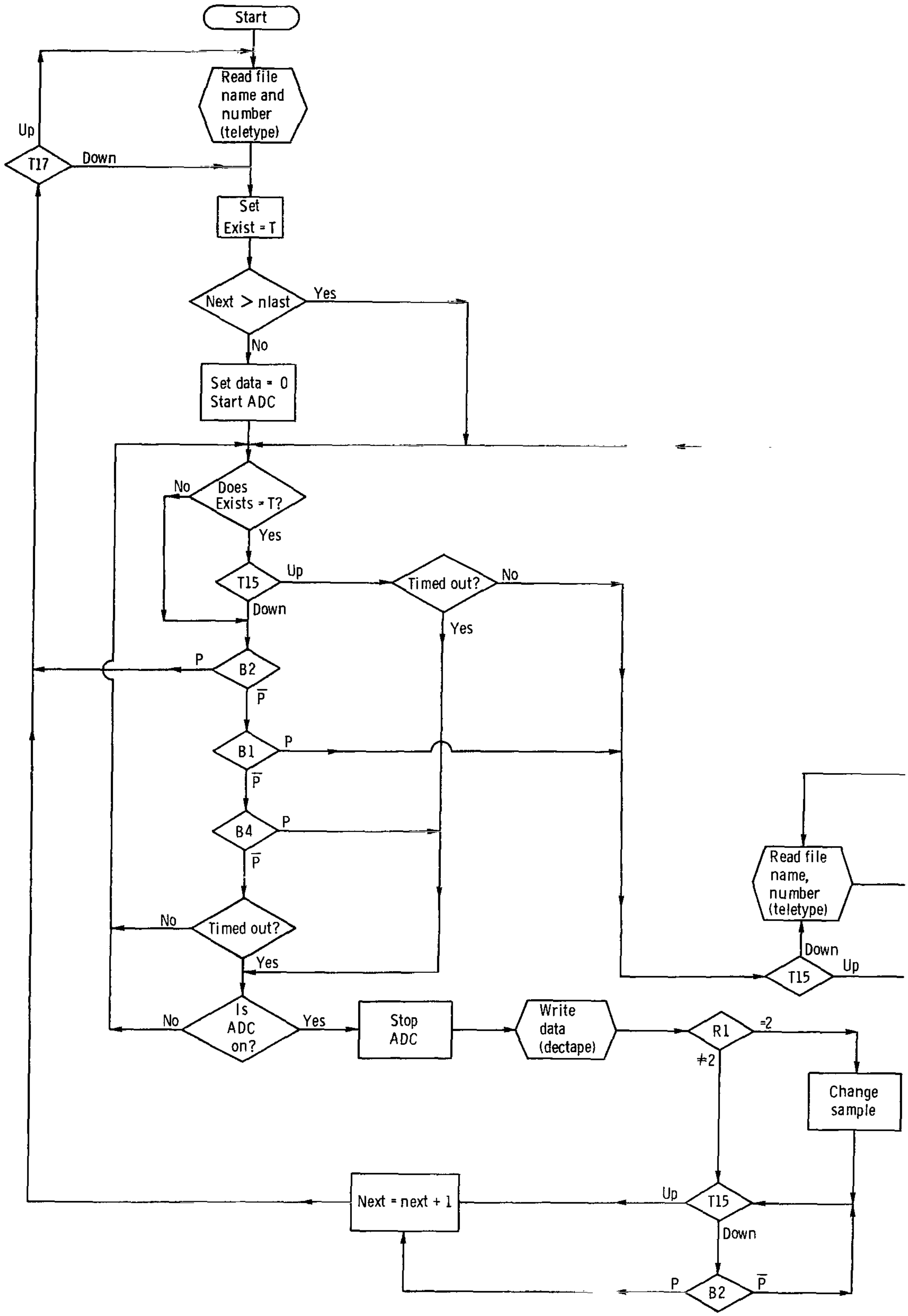

Figure 8. - Flow

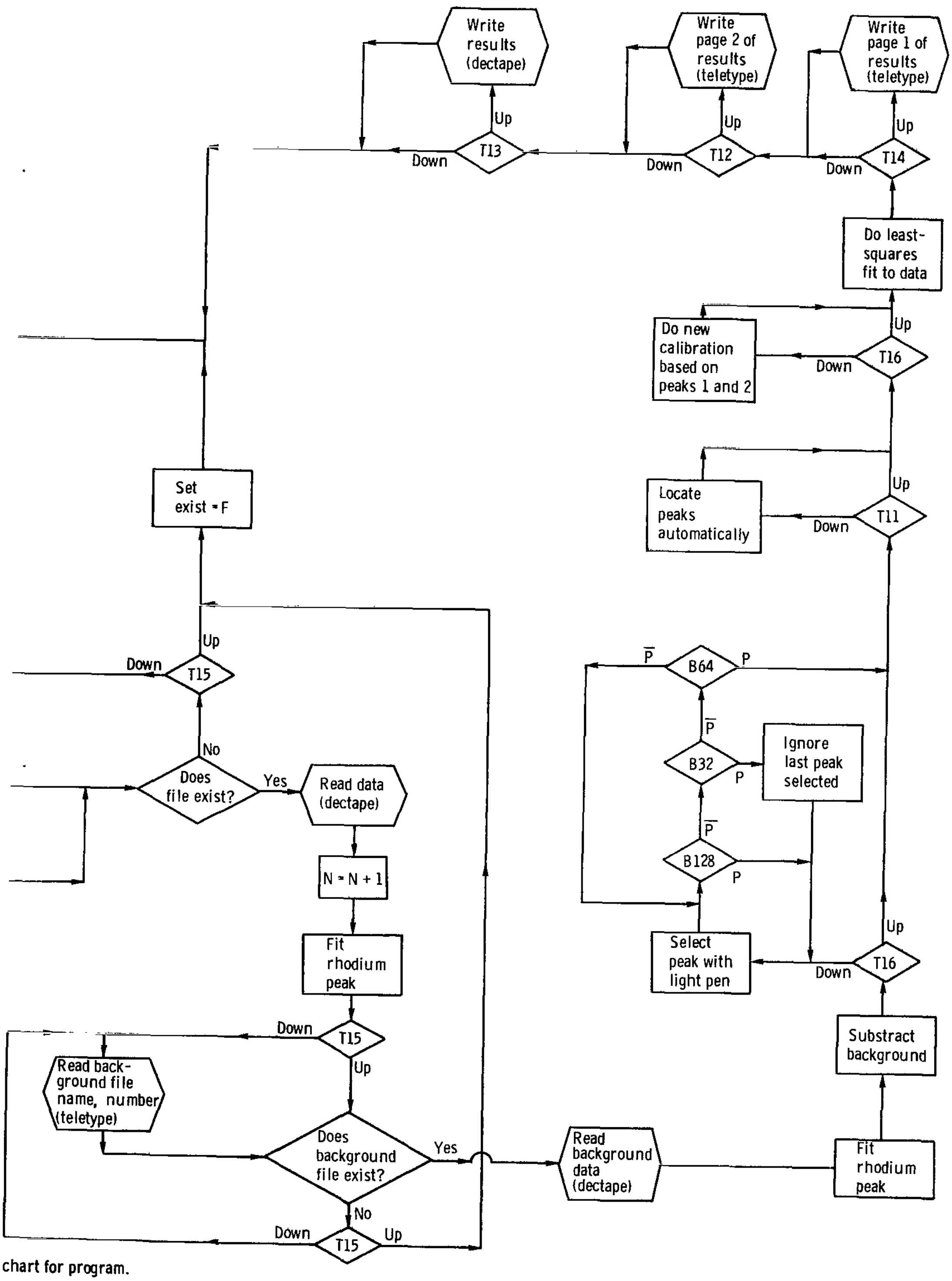

chart for program.

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546