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
**Report No.**
NASA TN D-8398

**Title and Subtitle**
SAVLOC, COMPUTER PROGRAM FOR AUTOMATIC CONTROL AND ANALYSIS OF X-RAY FLUORESCENCE EXPERIMENTS

**Author(s)**
Regis F. Leonard

**Performing Organization Name and Address**
Lewis Research Center  
National Aeronautics and Space Administration  
Cleveland, Ohio 44135

**Sponsoring Agency Name and Address**
National Aeronautics and Space Administration  
Washington, D.C. 20546

**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.

**Key Words (Suggested by Author(s))**
- Computer program
- Trace elements
- X-ray fluorescence

**Distribution Statement**
Unclassified - unlimited  
STAR Category 61

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

\[ N(E) = N_i \left( \frac{d\sigma}{d\Omega} \right)_E \rho_x \Omega \varepsilon(E) \]  \hspace{1cm} (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
- \( \rho_x \): areal density of atoms producing X-rays of energy \( E \)
- \( \Omega \): solid angle subtended by X-ray detector
- \( \varepsilon(E) \): 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

\[ N_i = K_{Rh} N_{Rh} \]  \hspace{1cm} (2)

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

\[
\rho_x = \frac{N(E)}{K_{Rh}^N_{Rh} \left( \frac{d\sigma}{d\Omega} \right)_E \Omega \epsilon(E)}
\]  

(3)

\[
F(E) = \frac{1}{K_{Rh} \left( \frac{d\sigma}{d\Omega} \right)_E \Omega \epsilon(E)}
\]  

(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

\[
\rho_x = \frac{N(E) \cdot F(E)}{N_{Rh}}
\]  

(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 fluorescence 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$_\alpha$ line, a check is made to determine whether the K$_\beta$ line can also be observed. If a line of the proper energy is located, the expected intensity of the K$_\beta$ line (determined from the intensity of the K$_\alpha$ line) is subtracted from the K$_\beta$ 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$_\alpha$ and L$_\beta$ 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
The parameters $P(1)$ 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 $P(1)$ vary with any change in detector or source geometry and must be redetermined each time such a change occurs. A typical set of parameters $P(1)$ 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

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

(7)

where $I$ and $\sigma$ (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

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

(8)

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

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

(9)

or

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

(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

\[ S_i(z, m) = \sum_{j=1-m}^{i+m} \sum_{h=l-m}^{l+m} (N_{l+1} - 2N_l + N_{l-1}) \]  

or equivalently

\[ S_i(z, m) = \sum_{all \, j} C_{ij}(z, m)N_j \]  

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

\[ C_{ij}(z, m) = \sum_{l=i-m}^{i+m} C_{lj}(z-1, m) \]  

and

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

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

\[ z = 5 \]

\[ m = \left(\frac{0.6\Gamma - 1}{2}\right) \]
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,

$$|S_i| > 1.6 \ |\delta S_i|$$

where $\delta S_i$ 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 $\delta N_i$ 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,

$$S_i < 0$$

Third,

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

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

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

(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 \( \sigma \) is calculated from the FWHM input (for a Gaussian distribution, \( \sigma = \text{FWHM}/1.665 \)).

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
>PGP, 16K, IZ
DEFINE RESIDENT CODE
>SIMUL, CAMADC, IPK3, PICKER
DESCRIBE LINKS & STRUCTURE
>LK1 = PEDE, RITE
>LK2 = FITALL, FITALL, GAUSS
>LK3 = ECAL1, ECAL2
>LK4 = PEGOUT, XYLD
>LK5 = LARFL, SOFT
>LK6 = S, SCALI, LOCUM
>MULT: SIT2: ERPO: ENERGY: IDENT
>
LINK TABLE

37225-37636 00412

RESIDENT CODE

SIMUL 35527-37224 00476
CAMADC 35731-35626 00576
IPK3 34742-35732 00771
PICKER 34305-34737 00433
PMBOL 34226-34304 00257
PTIME 34146-34225 00269
REC16 34126-34145 00082
SEND16 34113-34125 00013
CAMAC 33631-34112 00262
PANEL 33477-33630 00139
CROSS 33347-33476 00139
HISTO1 33296-33346 00121
PSNEW 33200-33225 00226
VPVEST 33014-33177 00164
VIDEO 32515-33013 00277
RTIME 32470-32514 00095
CLOCK 32372-32467 00076
RTEST 32358-32371 00022
GETALK 32267-32347 00061
POOL 32224-32266 00143
PEPO 32200-32223 00224
SS 32138-32177 00056
FLOAT 32117-32127 00011
FLOA? 32071-32116 00026
BR 32024-32073 00045
FIOPS 31302-32023 00522
INTEAE 31171-31301 00111
RELEASE 30057-31170 00112
BCDIO 22777-25433 02435
  .CB  22760-22776 00017

LINK -- LK5
LABEL  24757-27344 02366
SOPT    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
ARS    24673-24711 00017
LIMIT  24650-24672 00023
  .RC  24554-24647 00074
SOPT   24441-24553 00113
DIFF   24229-24440 00221

LINK -- REED
  PEED  26775-27344 00350
BDCEXT 26726-26774 00047
INPUT1 26637-26725 00067
  XPEAD 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 007616
INPUT1 26440-26526 00267
XPEAD  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 27033-27344 00312
BCDIO 24376-27032 02435
. CB 24357-24375 00017

LINK -- ENERG
ENERG 26500-27344 00645

LINK -- IDENT
IDENT 26066-27344 01257
RCDEXT 26017-26069 00047
PACKER 25716-26014 00101
4PS 25677-25715 00017
MOD 25656-25676 00021
BCDIO 23221-25655 22435
FILE 22742-23220 00257
. CR 22723-22741 00017

BLANK COMMON
. XX 12723-22722 10000

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),QRGN(8),DATUM(1024)
  REAL FNAME(2),LT,LT1
  LOGICAL ON,EXIST,BUTNS,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)/HSPECT,1H /
  CALL CLKON
  CALL PSTART
  CALL REDE(FNAME,NEXT,NLAST)
  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 HIST01(DATUM,NF,NX,NY,NSAW,1)
  CALL VPSTRT
  CALL ECAMCK
  CALL RTCOFF
  IF(ON) CALL ADCOFF
  MODE=0
  IFT=ITHUMB(2)
  ILT=0
  IF(NEXT.GT.NLAST) GO TO 122
  CALL ADCSTR(DATAD,GATEX,QRGN,ILT,ON,MODE)
  CALL RTCSTR(IFT,ADCOFF)
  CALL RTIME(TS)
  CALL LAMPON(1)
  CALL BUTNS(B)
  IF(IB,EQ.1) GO TO 101
  IF(IB,EQ.2) GO TO 501
  IF(.NOT.ON) GO TO 96
  IF(IB,EQ.4) GO TO 3F1
  CALL TOGGLS(IT)
  IF(BITEST(IT,15).AND.EXIST) GO TO 101
  GO TO 120
  CALL VPSTOP
  CALL IP02(EXIST)
  CALL VPSTOP
  CALL HIST01(DATUM,NF,NX,NY,NSAW,1)
  CALL JPSTRT
  IF(ON) GO TO 122
  IF(NEXT.GT.NLAST) GO TO 122
  CALL RTCOFF
  CALL ADCOFF
  CALL LAMPOF(1)
CALL TOGGLS(IT)
IF(BITEST(IT,15)) GO TO 3
CALL LAMPOF(2)
500
CALL BUTTNS(IB)
IF(IB .NE.2) GO TO 500
502
CALL LAMPOn(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 BUILD XCT FILE
INTEGER LIST(520), DATA(1024), RKGND(1024), GATE(5,30), CH0(30),
1 CHMAX, DATAD(4096), BEXT, IRT(5)
REAL FNAME(2), PARAM(15), DP(15), POINT(100), BNAME(2), XCH(2),
1 ELE(30), ENORM(30), BNORMX(30), DRC(30)
LOGICAL EXIST, DOUBLE, UNDOUR, TRIPLE, UNTRIP, QUAD, UNQUAD, GUIN,
1 UNQUIN, BUTTNS, PENSW, OUTPT, BITEST
EXTERNAL GAUSS
COMMON/DISPL/NF, NX, NY
COMMON/ERR/PHI2, IMIN, ITELL
COMMON/STUFF/GATE
COMMON/MUP/DDOUBLE, UNDOUR, TRIPLE, UNTRIP, QUAD, UNQUAD, GUIN, UNQUIN,
1 JPEAK, GAM, IDUR, ITRIP, IOU4D, IOU1N
COMMON/FINOUT/ELE, AI, A2
COMMON DATA
EQUIVALENCE (DATA(1), DATAD(1025)), (RKGND(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)), (DRC(1), DATAD(3921)),
4 (IRT(1), DATAD(4051)), (PARAM(1), DATAD(4096))
CALL LAMPOF(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, 0, LIST)
CALL OUTER(PARAM, LIST, ANORM, APNORM, CHMAX)
CALL REED(2, NAME, NEXT, EXIST)
CALL MAXER(RKGND, CHMAX)
CALL SETI(CHMAX, RKGND, POINT)
CALL FITALL(GAUSS, POINT, PARAM, DP, 0, LIST)
ITELL = 0
CALL SUSRT(APNORM, APNORM, CHMAX, OUTPT, GAM)
CALL PICKER
CALL TOGGLS(IT)
IF(BITEST(IT, 17)) CALL UPSTOP
IF(BITEST(IT,11)) CALL LOCUM
CALL LAMPOF(7)
IF(BITEST(IT,16)) GO TO 87
DO RR J=1,2
CALL ECAL1(J)
CALL FITALL(GAUSS,POINT,PARAM,DP,Q,LIST)
XCH(J)=FLOAT(CH0(J))+PARAM(2)-4.
CONTINUE
88 CALL ECAL2(XCH,A1,A2)
C.....BEGIN FITTING PEAKS
89 CALL LAMPOF(7)
DO RR J=3,JPEAK
90 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)>IMAX-IMIN+5)) LIST(4)=7
IF(PARAM(I+1)<-5.) LIST(4)=8
CONTINUE
IF(LIST(4).EQ.0) GO TO 86
CALL ERROR(J,IMIN,DP)
IF(.NOT.DOUBLE) GO TO 86
GO TO 81
85 CONTINUE
86 CALL ENERG(IMIN,A1,A2,J,DP)
CALL LABEL(J,JPEAK)
CALL REGOUT(OUTPT,ANORM,IMIN,DP,Q,J)
CONTINUE
CALL IDENT(ELE,ENORM,BNORMX,DP,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/DOURLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,
1 UNQUIN,JPEAK,GAM,IDUR,ITRIPI,QUAD,IQUN
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 ISI=NX
IS2=NY
IS3=NF
CALL SCAL1(NX,NY,NF)
CALL TOGGL5K(II)
IF(ICHK.NE.1) GO TO 66
IF(ISI.NE.NX).OR.(IS2.NE.NY).OR.(IS3.NE.NF)) GO TO 66
GO TO 65
66 CALL VPSTOP
CALL HISTOI(DATA,NF,NX,NY,NHIT,1)
CALL VPVECT(GATE,30,IHIT,2)
CALL VPSTRT
IF(BITEST(IT,16)) RETURN
IF(ICHK.EQ.1) GO TO 12
IF(ICHK.EQ.-1) GO TO 13
CALL ZERO(GATE)
CALL ZERO(CH0)
JPEAK=0
CALL CROSS(X,Y,OFF,3)
JPEAK=JPEAK+1
IF(PENSW(.TRUE.)) GO TO 9
ICHK=1
GO TO 6
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
CALL LAMPON(7)
CALL RUTTNS(IB)
IF(IR.EQ.128) GO TO 11
IF(IB.EQ.32) GO TO 14
IF(IR.EQ.64) RETURN
GO TO 10
JPEAK=JPEAK-1
GO TO 8
END

SUBROUTINE REDE(FNAME,NEXT,NLAST)
REAL FNAME(2)
CALL INPUT1(FNAME(1),5HFNAME,6)
CALL INPUT1(NEXT,4HNEXT,1)
CALL INPUT1(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))
MI=5*M+2
X=C(1)*FLOAT(DATA(ICHNL))
DO 41 J=1,MI
I1=ICHNL+J
I2=ICHNL-J
JI=JI+1
IF(I1.GT.1024) GO TO 42
IF(I2.LT.0) GO TO 42
X=X+C(JI)*FLOAT(DATA(I1))+C(JI)*FLOAT(DATA(I2))
GO TO 41

19
SUBROUTINE RITE(FNAME,NEXT,TS,IRT,ILT,ND)
 REAL FNAME(2)
 INTEGER DATAD(4096)
 COMMON DATAD
 CALL BCDEX(FNAME,NEXT)
 CALL ENTER(2,FNAME)
 WRITE(2) TS,IRT,ILT,ND,(DATA(I),I=1,ND)
 CALL RLAOF(3)
 WRITE(6,106) FNAME,TS,IRT,ILT
106 FORMAT(1X,2A5,5X,F12.5,16X,3X,16)
 CALL CLOSE(2)
 RETURN
END

SUBROUTINE GAUSSS(R,X,P,DP,V,KK,J)
 C......SUBROUTINE FOR US WITH 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(1255),(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 NP=K
 P=X(J)
3 DO 2 I=N,KK,3
 TP=P(I)*EXP(-(FLOAT(J)-P(I+1))**2/P(I+2)**2)
 P=P-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+2)*(FLOAT(J)-P(I+1))/P(I+2)
2 CONTINUE
 IERR=ILOWER+J-3
 IF(ITELL.EQ.0) V=(FLOAT(BKGND(ERR)*1))
 IF(ITELL.EQ.1) V=1.
 RETURN
END

SUBROUTINE ECAL1(J)
 C......SUBROUTINE CALL ED BY IPK2 TO DETERMINE ENERGY CALIBRATION.
 INTEGER CHE(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 (CH(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) = 2
LIST(5) = 3
LIST(6) = 3
IK = CH(1, J)
PARAM(1) = FLOAT(DATA(1K))
PARAM(2) = 4,
PARAM(3) = 4.5/1.665
IMIN = CH(1, J) - 3
DO 3 I = 1, 7
IC = CH(1, J) - 4 + I
POINT(I) = FLOAT(DATA(IC))
3 CONTINUE
RETURN
END

SUBROUTINE FCAL2(X, A1, A2)
REAL X(2)
LOGICAL BITEST
CALL TOGGLS(IT)
IF (BITEST(IT, 16)) GO TO 99
WRITE(6, 100)
CALL INPUT(E1, 2HE1, 2)
CALL INPUT(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, 6HG = 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, O, 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)), (DR(1), DATAD(3921)),
2 (PARAM(1), DATAD(4056))
NPARAM = LIST(6)
DO 92 I=1,NPARAM
DPC(I)=SQRT(TH(I)*DP(I))

92 CONTINUE
DO 91 I=1,NPARAM,3
IP2=I+1
ANORMX=PARAM(I)*PARAM(I+2)*1.7725/ANORM
BNORMX(J)=ANORM*XKYLD*ENORM(J)*FLOAT(I1TH(I))/10000.
DB(J)=(DP(I)/PARAM(I)+DP(I+2)/PARAM(I+2))
DY=AI*DP(IP2)/XKYLD*ENORM(J)
DZ=I(DJ)+DY)*BNORMX(J)
CALL TOGGLSC(J)
IF(BITEST(J,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,2F12.5,3X,F8.4,3X,F12.5,3X,1A5>
RETURN
END

C....FUNCTION XKYLD, TO TRANSFORM RATIO TO RHODIUM TO
1 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 1.2347,38.8516,-39.6.138,19m.348,-6477.202,
2 1866.1,-7116.m3/
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,PEAK)
REAL ENORM(30),ELE(30),K,MY,CRN,MN,NIBET,MO,BNORMX(30),DB(32)
INTEGER CH8(30),LIST(528),DATAD(4096)
LOGICAL CHECK
COMMON/FINOUT/ELE,A1,A2
COMMON DATAD
EQUIVALENCE (CH8(1),DATAD(4211)),(LIST(1),DATAD(3073)),
1 (ENORM(1),DATAD(3002)),(BNORMX(1),DATAD(3B61)),
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,2HN1/,
2 Cu,ZN,TLL,PBL,SE,RR/2HCU,2HZN,3HTLL,3HPRL,2HSE,2HBP/,
3 THL,RB,SR,ZR,MO/3HTHL,2HP,2HMR,2HZR,2HMO/,
4 ASC,SCI,TIV,VCR,CRMN/5HCA-SC,5HSC-TI,5HTI-V,
5 4HV-CP,5HCR-MN/,
6 FEBET,NIBET,CURET/5HFEPET,5HNIBET,5HCURST/,
7 ZNBT,PRRET,UNKMN/5HZNPET,5HBRRET,5H---/
NELE=(LIST(6)/3)+J
DO 2 I=J,NELE
2 RETURN
E=ENORM(I)
ELE(I)=UNKNW
IF(ABS(E-2.62).LE.0.06) ELE(I)=CL
IF(ABS(E-2.96).LE.0.06) ELE(I)=AR
IF(ABS(E-3.31).LE.0.06) ELE(I)=K
IF(ABS(E-3.69).LE.0.06) ELE(I)=CA
IF(ABS(E-4.09).LE.0.06) CALL SORT(CHECK,3.69,JPEAK)
IF(.NOT.CHECK.AND.(ABS(E-4.09).LT.0.06)) ELE(I)=SC
IF(CHECK) ELE(I)=CASC
CHECK=.FALSE.
IF(ABS(E-4.51).LE.0.06) CALL SORT(CHECK,4.09,JPEAK)
IF(.NOT.CHECK.AND.(ABS(E-4.51).LT.0.06)) ELE(I)=TI
IF(CHECK) ELE(I)=STI
CHECK=.FALSE.
IF(ABS(E-4.95).LE.0.06) CALL SORT(CHECK,4.51,JPEAK)
IF(.NOT.CHECK.AND.(ABS(E-4.95).LT.0.06)) ELE(I)=V
IF(CHECK) ELE(I)=TV
CHECK=.FALSE.
IF(ABS(E-5.41).LE.0.06) CALL SORT(CHECK,4.95,JPEAK)
IF(.NOT.CHECK.AND.(ABS(E-5.41).LT.0.06)) ELE(I)=CR
IF(CHECK) ELE(I)=VCR
CHECK=.FALSE.
IF(ABS(E-5.90).LE.0.06) CALL SORT(CHECK,5.41,JPEAK)
IF(.NOT.CHECK.AND.(ABS(E-5.90).LT.0.06)) ELE(I)=MN
IF(CHECK) ELE(I)=VMN
CHECK=.FALSE.
IF(ABS(E-6.40).LE.0.06) ELE(I)=FE
IF(ABS(E-6.80).LE.0.06) ELE(I)=CO
IF(ABS(E-7.06).LE.0.06) ELE(I)=FERET
IF(ABS(E-7.48).LE.0.06) ELE(I)=NI
IF(ABS(E-8.05).LE.0.06) ELE(I)=CU
IF(ABS(E-8.26).LE.0.06) CALL SORT(CHECK,7.48,JPEAK)
IF(CHECK) ELE(I)=CUBET
CHECK=.FALSE.
IF(ABS(E-8.63).LE.0.06) ELE(I)=7N
IF(ABS(E-9.09).LE.0.06) CALL SORT(CHECK,8.05,JPEAK)
IF(CHECK) ELE(I)=CUBET
CHECK=.FALSE.
IF(ABS(E-9.57).LE.0.06) CALL SORT(CHECK,8.63,JPEAK)
IF(CHECK) ELE(I)=7N3ET
CHECK=.FALSE.
IF(ABS(E-10.25).LE.0.06) CALL SORT(CHECK,12.21,JPEAK)
IF(CHECK) ELE(I)=TLL
CHECK=.FALSE.
IF(ABS(E-10.55).LE.0.06) CALL SORT(CHECK,12.61,JPEAK)
IF(CHECK) ELE(I)=PBL
CHECK=.FALSE.
IF(ABS(E-11.22).LE.0.06) ELE(I)=SE
IF(ABS(E-11.89).LE.0.06) ELE(I)=BR
IF(ABS(E-12.21).LE.0.06) CALL SORT(CHECK,12.27,JPEAK)
IF(CHECK) ELE(I)=TLL
CHECK=.FALSE.
IF(ABS(E-12.61).LE.0.06) CALL SORT(CHECK,10.55,JPEAK)
IF(CHECK) ELE(I)=PBL
CHECK=.FALSE.
IF(ABS(E-12.99).LE.0.06) CALL SORT(CHECK,16.20,JPEAK)
IF(CHECK) ELE(I)=THL
CHECK=.FALSE.
IF(ABS(E-13.29).LE.0.06) CALL SORT(CHECK,11.92,JPEAK)
IF(CHECK) ELE(I)=RR
CHECK=.FALSE.
IF(CARS(E-13.39),LE.,0.06) ELE(I)=RR
IF(CARS(E-14.16),LE.,0.06) ELE(I)=SR
IF(CARS(E-15.77),LE.,0.06) ELE(I)=ZR
IF(CARS(E-16.20),LE.,0.12) CALL SORT(CHECK,12.90,JPEAK)
IF(CHECK) ELE(I)=THL
CHECK=.FALSE.
IF(CARS(E-17.47),LE.,0.26) ELE(I)=MO
CONTINUE
RETURN
END

SUBROUTINE SORT(CHECK,X,JPEAK)
INTEGER CH0(30),DATA(4096)
LOGICAL CHECK
REAL ELE(30),RNORMX(30),DR(30)
COMMON/FINOUT/ELE,A1,A2
COMMON DATAD
EQUIVALENCE (CH0(1),DATA(1021)),(RNORMX(1),DATA(3861)),
1 (DB(1),DATA(3921))
CHECK=.FALSE.
DO 2 I=1,JPEAK
E=A1*FLOAT(CH0(I))+A2
IF(ABS(E-X),LE.,0.12) GO TO 5
GO TO 2
5 I=JPEAK
CHECK=.TRUE.
CONTINUE
RETURN
END

SUBROUTINE SCALE(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 (DATAI),DATA(1025),(CH0(1),DATA(4221))
I2=IROTOR(2)
NF=LIMIT(1,256*I2-255,1024)
I3=IROTOP(3)
NX=LIMIT(8,13+7,12)
I4=IROTOR(4)
NY=LIMIT(8,14+6,18)
DO 10 I=1,30
10 LI=CH0(I)
IF(LI.EQ.0) LI=1
GATE(I,1)=LIMIT(0,(CH0(I)-NF)*8/2**13,1023)
GATE(2,1)=DATA(LI)/(2**14)*16
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), BKGN(1024)
REAL FNAME(2)
LOGICAL EXIST, BITEST
COMMON DATAD
EQUIVALENCE (DATA(1), DATAD(1025)), (BKGN(1), DATAD(2249))
CALL TOGGLS(IT)
IF(BITEST(IT, 15)) GO TO 3
CALL INPUT1(FNAME(1), SHDNAME, G)
CALL INPUT1(NEXT, 4HNEXT, 1)
CALL BCDEXTC(FNAME, NEXT)
CALL FSTATCL, 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(G, 121) FNAME
101 FORMAT(1HI, 12H RUN NUMBER, 2A5)
IF(ITYPE.EQ.2) CALL ZERO(BKGN)
CALL SEEK1(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, (BKGN(I), I = 1, 1024)
RETURN
100 FORMAT(1X, 6H0 FILE, 2A5, 12H NOT FOUND.)
END

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

SUBROUTINE SET1(CHMAX, IDAT, POINT)
C........ROUTE 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, 141
ICO = IMIN + I - 1
POINT(I) = IDAT(ICO)
200 CONTINUE
PARAM(2) = (BG1)
PARAM(1) = (BG2 - BG1) / 40.
PARAM(5) = 6.1 665
PARAM(3) = FLOAT(IDAT(CHMAX)) - PARAM(2) - PARAM(1)*20.
PARAM(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, APNORM, 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(X,11HERROR, TYPE,12I)
ANORM = PARAM(3)*PARAM(5)*1.7725
ABNORM = 40.*PARAM(2)+08.*PARAM(1)
WRITE(6,123) PARAM(4), ANORM
WRITE(6,124) ABNORM
123 FORMAT('X,15HBACKGROUND HAS, F10.3,8H COUNTS.)
124 FORMAT('X,15HNORMALIZATION PEAK IN CHANNE
1L, FR.2,4H HAS, F10.2,8H COUNTS.)
RETURN
END

SUBROUTINE SUBTR(ANORM, APNORM, CHMAX, OUTPT, GAM)
INTEGER DATA(1024), CHMAX, DATA(4096), BKGND(1024)
REAL PARAM(15)
LOGICAL OUTPT, BITEST
COMMON DATAD
EQUIVALENCE (DATA(1), DATA(1025)), (BKGND(1), DATA(2049)),
( PARAM(1), DATA(4056))
PARAM(4) = PARAM(4) + FLOAT(CHMAX-21)
CNORM = PARAM(3)*PARAM(5)*1.7725
CBNORM = 40.*PARAM(2)+08.*PARAM(1)
WRITE(6,123) PARAM(4), CNORM
WRITE(6,124) CBNORM
124 FORMAT('X,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
CONTINUE
BKGND(1023) = 0
BKGND(1024) = 0
CALL TOGLS(IT)
IF (BITEST(IT,15)) RETURN
WRITE(6,105)
CALL INPUT1(OUTPT, SHOUTPT, 4)
CALL INPUT1(GAM, SHGAMMA, 2)
SUBROUTINE MULT(IMAX,IMIN,J)
  INTEGER CH0(30),DATAD(4096)
  LOGICAL DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,UNQUIN
  COMMON/DOUBLE/ DOUBLE,UNDOUB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,
  1 QUIN,JPEAK,
  1 GAM.IDUB,ITRIP,IQUAD,IQUIN
  COMMON DATAD
  EQUIVALENCE (CH0(1),DATAD(4021))
  DOUBLE=.FALSE.
  TRIPLE=.FALSE.
  QUAD=.FALSE.
  QUIN=.FALSE.
  IMAX=CH0(J)+IFIX(GAM)
  IMIN=CH0(J)-IFIX(GAM)
  IF(UNDOUB) GO TO R2
  IF(J.EQ.JPEAK) GO TO R2
  IF(CH0(J+1).LT.(I0+IFIX(4.*GAM))) DOUBLE=.TRUE.
  IDUB=CH0(J+1)
  IF(UNTRIP) GO TO R2
  IF(J.EQ.JPEAK-1) GO TO R2
  IF(CH0(J+2).LT.(IDUB+IFIX(4.*GAM))) TRIPLE=.TRUE.
  TRIPLE=DOUBLE.AND.TRIPLE
  ITRIP=CH0(J+2)
  IF(UNQUAD) GO TO R2
  IF(J.EQ.JPEAK-2) GO TO R2
  IF(CH0(J+3).LT.(ITRIP+IFIX(4.*GAM))) QUAD=.TRUE.
  QUAD=TRIPLE.AND.QUAD
  IQUAD=CH0(J+3)
  IF(UNQUIN) GO TO R2
  IF(J.EQ.JPEAK-3) GO TO R2
  IF(CH0(J+4).LT.(IQUAD+IFIX(4.*GAM))) QUIN=.TRUE.
  QUIN=QUAD.AND.QUAD
  IQUIN=CH0(J+4)
R2
  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)
  REAL PARAM(15),POINT(100)
  INTEGER DATA(1024),LIST(528),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 DATA  
EGUVALENCIE (DATA(1), DATAD(1025)), (LIST(1), DATAD(3073)),  
1  (CH0(I), DATAD(4021)), (POINT(I), DATAD(3093)),  
2  (IR(I), DATAD(4051)), (PARAM(I), DATAD(4056))  
ITOT=IMAX-IMIN+1  
DO 84 I=1, ITOT  
IDOWN=IMIN+I-1  
POINT(I)=FLOAT(DATA(IDOWN))  
84 CONTINUE  
IO=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  
PARAM(6)=FLOAT(IDUB))  
PARAM(5)=FLOAT(IDUB-IMIN+1)  
PARAM(6)=GAM/1.665  
87 PARAM(1)=FLOAT(DATA(IO))  
PARAM(2)=FLOAT(IO-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 RR  
LIST(2)=6  
LIST(6)=6  
IF(.NOT.TRIPLE) GO TO RR  
LIST(2)=9  
LIST(6)=9  
IF(.NOT QUIAD) GO TO RR  
LIST(2)=12  
LIST(6)=12  
IF(.NOT QUIIN) GO TO RR  
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
SUBROUTINE IDENT(ELE, ENORM, B NORMX, DR, JPEAK, FNAME, NEXT)

C........... SUBROUTINE TO MAKE FINAL IDENTIFICATION OF ELEMENTS
C........... AND MAKE CORRECTIONS FOR BETA PEAKS.

REAL ELE(30), ENORM(30), B NORMX(30), DR(30), FNAME(2),
1 ESI(24), ES2(24), ATOR(24), NAM(24), EFF(4)

LOGICAL BITEST, BETA

DATA ESI(1), ES2(1), ATOR(1), NAM(1)/1.62, 241, 0, 2HCL/
1 ESI(2), ES2(2), ATOR(2), NAM(2)/2.96, 319, 0, 2HAR/
2 ESI(3), ES2(3), ATOR(3), NAM(3)/3.513, 3589, 0, 1HK/
3 ESI(4), ES2(4), ATOR(4), NAM(4)/4.691, 4012, 0, 2HCA/
4 ESI(5), ES2(5), ATOR(5), NAM(5)/4.090, 4460, 0, 1HSC/
5 ESI(6), ES2(6), ATOR(6), NAM(6)/4.579, 5164, 0, 2HCT/
6 ESI(7), ES2(7), ATOR(7), NAM(7)/4.952, 5427, 0, 1HV/

DATA ESI(8), ES2(8), ATOR(8), NAM(8)/5.414, 5946, 0, 114, 2HCP/
1 ESI(9), ES2(9), ATOR(9), NAM(9)/5.898, 6490, 0, 10R, 2HNN/
2 ESI(10), ES2(10), ATOR(10), NAM(10)/6.403, 7057, 0, 106, 2HFE/
3 ESI(11), ES2(11), ATOR(11), NAM(11)/6.930, 7649, 0, 118, 2HCO/
4 ESI(12), ES2(12), ATOR(12), NAM(12)/7.477, 8264, 0, 105, 2HNI/
5 ESI(13), ES2(13), ATOR(13), NAM(13)/8.047, 8904, 0, 113, 2HCU/

DATA ESI(14), ES2(14), ATOR(14), NAM(14)/8.638, 9571, 0, 117, 2HZN/
1 ESI(15), ES2(15), ATOR(15), NAM(15)/9.543, 11725, 0, 283, 2HAS/
2 ESI(16), ES2(16), ATOR(16), NAM(16)/11.221, 12495, 0, 13, 2HSE/
3 ESI(17), ES2(17), ATOR(17), NAM(17)/11.923, 13290, 0, 136, 2HBR/
4 ESI(18), ES2(18), ATOR(18), NAM(18)/14.164, 15834, 0, 151, 2HSR/
5 ESI(19), ES2(19), ATOR(19), NAM(19)/15.774, 17666, 0, 172, 2HZN/

DATA ESI(20), ES2(20), ATOR(20), NAM(20)/17.478, 19607, 0, 18, 2HMO/
1 ESI(21), ES2(21), ATOR(21), NAM(21)/8.396, 9670, 0, 775, 1HW/
2 ESI(22), ES2(22), ATOR(22), NAM(22)/10.266, 12210, 0, 672, 2HTL/
3 ESI(23), ES2(23), ATOR(23), NAM(23)/12.549, 12611, 0, 645, 2HPI/
4 ESI(24), ES2(24), ATOR(24), NAM(24)/12.966, 16209, 0, 553, 2HTH/

DATA EPS, UNKNW/R, 060, 2H--/

DATA EFF(1), EFF(2), EFF(3), EFF(4)/1.775, 1.66, 1.66, 2.72/

NEXT=NEXT+1

CALL BCDEXT(FNAME, NEXT)

DO 4 I=3, JPEAK

ELE(I)=UNKNW

DO 2 J=1,24

IF(ARS(ENORM(I)-ESI(I)), LT, EPS) GO TO 1
GO TO 2  
BETA=.FALSE.  
DO 3 K=I, JPEAK  
IF(ARSKENORM(K) .GT. 0.090) GO TO 3  
IF(ARSKENORM(K) .GT. 0.3*RNOFMX(I)) BETA=.TRUE.  
BNORMMX(K) = RNOFMX(K) - ATOR(J)*BNORMMX(I)  
IF(ARSKENORM(K) .LT. 0.) BNORMMX(K) = 0.  
CONTINUE  
IF(J,LT,21) ELE(I)=NAM(J)  
IF(J,GE,21 .AND. BETA) GO TO 6  
GO TO 2  
ELE(I)=NAM(J)  
BNORMMX(I)=BNORMMX(I)*EFF(J-20)  
DR(I)=DB(I)*EFF(H-20)  
2 CONTINUE  
4 CONTINUE  
CALL TOGGLSC(IT)  
IF(.NOT.BITEST(IT,13)) GO TO 5  
CALL ENTER(3,FNAME)  
WRITE(3,100) (ELE(I),BNORMMX(I),DR(I),I=3,JPEAK)  
100 FORMAT(1X,IA5,2EI11.5)  
CALL CLOSE(3)  
5 CALL TOGGLSC(IT)  
IF(.NOT.BITEST(IT,12)) RETURN  
WRITE(6,103) FNAME  
C........ WRITE(6,104) G  
C........ FORMAT(1X,2HG= 9F6.4)  
WRITE(6,101)  
WRITE(6,102) (ELE(I),BNORMMX(I),DR(I),I=3,JPEAK)  
RETURN  
101 FORMAT(1X,7HELEMENT,ZX, 13HCONCENTRATION,9H ERROR)  
102 FORMAT(4X,1A5,E12.6,E12.6)  
103 FORMAT(1HI,5X,2A5)  
END

SUBROUTINE ERR0(J,IIMN,DP)  
REAL ELE(30),BNORMMX(30),DP(15),PARAM(15)  
INTEGER LIST(520),DATAD(4096),IR(5),CH0(30)  
LOGICAL DOUBLE,UND0UB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN,UNQUIN, 
BITEST  
COMMON /MUP/DOUBLE,UND0UB,TRIPLE,UNTRIP,QUAD,UNQUAD,QUIN, 
UNQUIN,JPEAK,GAM,IDUB,ITRIP,IQUAD  
COMMON DATAD  
COMMON/FINOUT/ELE,A1,A2  
EQUIVALENCE (LIST(1),DATAD(3073)),(BNORMMX(1),DATAD(3861)), 
1 (DB(1),DATAD(3921)),(IR(1),DATAD(4051)),(DATAD(4021)), 
2 (CH0(1)),(PARAM(1),DATAD(4056))  
DATA RNONE/4,HNONE/  
CALL TOGGLSC(IT)  
IF(BITEST(IT,14)) WRITE(6,111) LIST(4)  
111 FORMAT(1H ERROR,TYPE,12)  
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
UNQUAD=.TRUE.
RETURN
UNTRIP=.TRUE.
RETURN
UNDOUB=.TRUE.
RETURN
PARAM(1)=0.
PARAM(3)=0.
PARAM(2)=FLOAT(CH(0)(J)-MIN+1)
DP(1)=0.
DP(2)=0.
DP(3)=0.
RETURN
END

SUBROUTINE ENERG(IMIN, AI, A2, J, DP)
REAL PARAM(15), ENORM(30), DP(15)
INTEGER DATAD(4096), LIST(520), IR(5), CH0(30)
COMMON DATAD
EQUIVALENCE (DATAD(3900), 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
NPEAK=LIST(6)/3
NPEAK=NPEAK+ISUM
DO 3 I=1, NPEAK
 MPEAK=NPEAK-I+1
 M3PEAK=3*MPEAK
 IF(IR(M3PEAK).EQ.0) GO TO 4
 J=PEAK=J+MPEAK-I
 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
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(K0)=A1*PARAM(IP2)+A2
31
K = K + 1
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), CH(30), DATA(4096), CHMAX,
1 CH1, CH2, CH3, CH4, CH5, RKGND(1024), GATI(140)
REAL C(36), R(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, IDUR, ITRIP, IQUAD, IQUIN
COMMON/DATA
COMMON/DIFF/C,P,M
EQUIVALENCE (DATA(1), DATA(125)), (CH(1), DATA(4021)),
1 (RKGND(1), DATA(2049)), (GATI(1), GATE(1,3))
CALL ZERO(GATI)
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)=-.2
C(2)=.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.P) K2=K-J+2
B(J)=B(J)+C(K1)+C(K2)
IF(K1.GT.36) R(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
MI=5*M**2
DO 50 J=2,M
PHI=PHI+2.*C(J)**2
50 CONTINUE
PHI2=SQRT(PHI)
C.....COMPUTE CRITERIA FOR PEAK LOCATION
C.....NI=EXPECTED NUMBER OF CHANNELS IN NEGATIVE PEAK
NI=IFIX(1.22*(GAM+0.5))
NI MAX=NI+4
NI MIN=NI-2
C.....BEGIN TO SEARCH FOR PEAKS
CH5=25
JPEAK=2
JPEAK=JPEAK+1
ICHNL=ICHNL
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 63
IF(S(ICHNL).LT.-1.6*PHI2*SQRT(FLOAT(BKGND(ICHNL-2)+1))) GO TO 63
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 65
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*SQRT(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
N2 MIN=IFIX((PHI2*SQRT(FLOAT(DATA(CH4)+1))/ABS(S(CH4)))*0.5*FLOAT(N1 MIN)+0.5)
N2 MAX=IFIX((PHI2*SQRT(FLOAT(DATA(CH4)+1))/ABS(S(CH4)))*0.5*FLOAT(N1 MAX)+0.5)
N3 MIN=IFIX((PHI2*SQRT(FLOAT(DATA(CH4)+1))/ABS(S(CH4)))+0.5)
N3 MAX=IFIX((PHI2*SQRT(FLOAT(DATA(CH4)+1))/ABS(S(CH4)))+0.5)
C.....IF(CH5-CH3+1.LT.N1 MIN) GO TO 59
C.....IF(CH5-CH3+1.GT.N1 MAX) GO TO 59
C.....IF(N2 MAX.EQ.0) GO TO 72
C.....IF(CH3-CH2-1.GT.N2 MAX) GO TO 59
GO TO 73
72 IF(CH3-CH2-1.GT.1) CONTINUE
73 IF(CH2-CH1+1.LT.N3 MIN) CONTINUE
CH0(JPEAK)=CH4
GATE(1,JPEAK)=CH4
GATE(2,JPEAK)=DATA(CH4)
GATE(3,JPEAK)=0
GATE(4,JPEAK)=128
GATE(5,JPEAK)=0
CALL SCAL1(NX,NY,NF)
GO TO 58
70 JPEAK=JPEAK-1
RETURN
REFERENCES


### TABLE I. - ELEMENTS RECOGNIZED BY SAVLOC

**(a) K series**

<table>
<thead>
<tr>
<th>Element</th>
<th>$K_\alpha$ energy, keV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chlorine</td>
<td>2.62</td>
</tr>
<tr>
<td>Argon</td>
<td>2.96</td>
</tr>
<tr>
<td>Potassium</td>
<td>3.313</td>
</tr>
<tr>
<td>Calcium</td>
<td>3.691</td>
</tr>
<tr>
<td>Scandium</td>
<td>4.090</td>
</tr>
<tr>
<td>Titanium</td>
<td>4.510</td>
</tr>
<tr>
<td>Vanadium</td>
<td>4.952</td>
</tr>
<tr>
<td>Chromium</td>
<td>5.414</td>
</tr>
<tr>
<td>Manganese</td>
<td>5.898</td>
</tr>
<tr>
<td>Iron</td>
<td>6.403</td>
</tr>
<tr>
<td>Cobalt</td>
<td>6.930</td>
</tr>
<tr>
<td>Nickel</td>
<td>7.477</td>
</tr>
<tr>
<td>Copper</td>
<td>8.047</td>
</tr>
<tr>
<td>Zinc</td>
<td>8.638</td>
</tr>
<tr>
<td>Arsenic</td>
<td>10.543</td>
</tr>
<tr>
<td>Selenium</td>
<td>11.221</td>
</tr>
<tr>
<td>Bromine</td>
<td>11.923</td>
</tr>
<tr>
<td>Strontium</td>
<td>14.164</td>
</tr>
<tr>
<td>Zirconium</td>
<td>15.774</td>
</tr>
<tr>
<td>Molybdenum</td>
<td>17.478</td>
</tr>
</tbody>
</table>

**(b) L series**

<table>
<thead>
<tr>
<th>Element</th>
<th>$L_\alpha$ energy, keV</th>
<th>$L_\beta$ energy, keV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tungsten</td>
<td>8.396</td>
<td>9.670</td>
</tr>
<tr>
<td>Thallium</td>
<td>10.266</td>
<td>12.210</td>
</tr>
<tr>
<td>Lead</td>
<td>10.549</td>
<td>12.611</td>
</tr>
<tr>
<td>Thorium</td>
<td>12.966</td>
<td>16.200</td>
</tr>
</tbody>
</table>

### TABLE II. - PARAMETERS USED TO DESCRIBE X-RAY EFFICIENCY

**FUNCTION F(E)**

<table>
<thead>
<tr>
<th>I</th>
<th>P(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.855</td>
</tr>
<tr>
<td>2</td>
<td>54.874</td>
</tr>
<tr>
<td>3</td>
<td>-705.322</td>
</tr>
<tr>
<td>4</td>
<td>5499.011</td>
</tr>
<tr>
<td>5</td>
<td>-22895.21</td>
</tr>
<tr>
<td>6</td>
<td>48343.26</td>
</tr>
<tr>
<td>7</td>
<td>-40357.17</td>
</tr>
</tbody>
</table>
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 580.93 HAS 526878.78 COUNTS.
BACKGROUND HAS 152550.379 COUNTS.
NEXT (1) = 0

NORMALIZATION PEAK IN CHANNEL 581.48 HAS 648537.67 COUNTS.
BACKGROUND HAS 58875.779 COUNTS.
DO YOU WISH EXTRA OUTPUT?
OUTPUT (L) = F

ENERGIES FOR CALIBRATION:
TI (R) = 2.483
ZN (R) = 12.611
d = 1.0000

PEAK CHANNEL STRENGTH ENERGY CONCENTRATION ERROR ELEM

| PEAK CHANNEL STRENGTH ENERGY CONCENTRATION ERROR ELEM |
|-----------------|-----------------|-----------------|-----------------|
| 3 70.035        | 0.000037        | 3.701           | 0.29782 +00.19374 +01 CA |
| 4 91.404        | 0.004300        | 4.516           | 0.49822 +00.38403 +01 TI |
| 5 128.884       | 0.000059        | 5.912           | 0.22556 +00.78361 +00 MN |
| 6 140.871       | 0.007699        | 6.463           | 0.32115 +00.24457 +01 FE |
| 7 157.888       | 0.01587        | 7.048           | 0.22082 +00.15199 +01 FEBET |
| 8 168.666       | 0.001316        | 7.461           | 0.23297 +00.94859 +00 Ni |
| 9 183.369       | 0.004440        | 8.047           | 0.64054 +00.88547 +01 Co |
| 10 199.656      | 0.030763        | 8.620           | 0.39711 +00.13935 +01 ZN |
| 11 224.572      | 0.003466        | 9.395           | 0.41645 +00.61117 +00 ZIBET |
| 12 249.423      | 0.009359        | 10.543          | 0.23081 +00.57698 +00 PUL |
| 13 285.057      | 0.012995        | 11.982          | 0.89666 +00.28890 +00 BR |
| 14 303.631      | 0.025692        | 12.611          | 0.14432 +00.34939 +00 PUL |
| 15 301.212      | 0.001912        | 13.202          | 0.97893 +00.19976 +00 BBET |
| 16 359.355      | 0.03763        | 14.737          | 0.16038 +00.17434 +00 ----- |
| 17 369.690      | 0.001194        | 15.093          | 0.49785 +00.15608 +00 ----- |
| 18 365.562      | 0.003646        | 15.739          | 0.13832 +00.25890 +00 ZR |
| 19 407.381      | 0.009938        | 16.566          | 0.30213 +00.71951 +00 ----- |
| 20 425.751      | 0.051486        | 17.422          | 0.16383 +00.29195 +00 MO |

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

SEPT 011

ELEMENT CONCENTRATION ERROR

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>CONCENTRATION</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>0.29782 +00</td>
<td>0.19374 +01</td>
</tr>
<tr>
<td>TI</td>
<td>0.64054 +00</td>
<td>0.88547 +01</td>
</tr>
<tr>
<td>MN</td>
<td>0.23081 +00</td>
<td>0.57698 +00</td>
</tr>
<tr>
<td>FE</td>
<td>0.29711 +00</td>
<td>0.13935 +01</td>
</tr>
<tr>
<td>Ni</td>
<td>0.41645 +00</td>
<td>0.61117 +00</td>
</tr>
<tr>
<td>PUL</td>
<td>0.89666 +00</td>
<td>0.28890 +00</td>
</tr>
<tr>
<td>BBET</td>
<td>0.14432 +00</td>
<td>0.34939 +00</td>
</tr>
<tr>
<td>ZIBET</td>
<td>0.97893 +00</td>
<td>0.19976 +00</td>
</tr>
<tr>
<td>PUL</td>
<td>0.16038 +00</td>
<td>0.17434 +00</td>
</tr>
<tr>
<td>ZR</td>
<td>0.49785 +00</td>
<td>0.15608 +00</td>
</tr>
<tr>
<td>MO</td>
<td>0.16383 +00</td>
<td>0.29195 +00</td>
</tr>
</tbody>
</table>

Figure 6. - Sample of second page of teletype output.
### Figure 7.

Sample of augmented page 1 teletype output.
Read file name and number (teletype)

Set Exist = T

Next > nlast? Yes

Set data = 0
Start ADC

Does Exists > T? Yes

Round off?

Stop ADC
Write data (dectape)

Is ADC on? Yes

Next = next + 1

Figure 8. - Flow
"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