This is to certify that the

thesis entitled

A NEUTRON ACTIVATION ANALYSIS PROCEDURE DEVELOPED FOR USE AT THE MICHIGAN STATE UNIVERSITY NUCLEAR REACTOR LABORATORY

presented by

STEVEN A. LELEWER

has been accepted towards fulfillment
of the requirements for

M.S. degree in Chemical Engineering

Bruce W. Wilkinson
Major professor

Date May 16, 1978

O-7639

A NEUTRON ACTIVATION ANALYSIS PROCEDURE DEVELOPED FOR USE AT THE MICHIGAN STATE UNIVERSITY NUCLEAR REACTOR LABORATORY

By

Steven Arthur Lelewer

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Chemical Engineering

1978

ABSTRACT

A NEUTRON ACTIVATION ANALYSIS PROCEDURE DEVELOPED FOR USE AT THE MICHIGAN STATE UNIVERSITY NUCLEAR REACTOR LABORATORY

By

Steven Arthur Lelewer

The purpose of this thesis is to describe the development of a method of performing neutron activation analysis at the Michigan State University Nuclear Reactor Laboratory. One requirement of the method was that it could be performed in a reasonable amount of time. Manual spectrum analysis is a long and tedious procedure. Philip A. Baedecker developed a spectral analysis computer program entitled SPECTRA. Part of the project described in this thesis entailed modifying SPECTRA to operate on the Michigan State University CDC 6500 computer. SPECTRA requires that the spectrum data be input via punched cards. An interface was designed to allow the interconnection of the Nuclear Data multichannel analyzer and the engineering building IBM 1800 computer. Program ND was created to control the transfer of data from the analyzer to the IBM 1800, and to handle the formatting and punching of the data onto cards for input to program SPECTRA. A 2048

channel spectrum can be transferred from the Nuclear Data analyzer to the IBM 1800 in approximately 15 seconds. The spectral analysis can be performed by SPECTRA at a cost of $1 to $2 on the campus computer.

The final chapter of the thesis provides a detailed procedure for transferring the data from the Nuclear Data analyzer to the IBM 1800, including cable hook-ups and the oepration of the IBM 1800 computer. The procedure also incudes instructions for the usage of program SPECTRA.

ACKNOWLEDGEMENTS

The author would like to thank Professors Bruce Wilkinson and George Coulman for their guidance, time and friendship.

Thanks should be given to Les Keith of the Engineering Building Computer Laboratory for his assistance in the development of the interface circuitry and to Robert Morris of Norwalk, California, a computer science student, for his assistance in the modifications of program SPECTRA.

A special thanks goes to my wife, Debby, for her moral support, understanding, and encouragement.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iv
INTRODUCTION	1
THE INTERFACE AND THE ND PROGRAM	7
SPECTRA	15
USER'S MANUAL	23
BIBLIOGRAPHY	33
APPENDIX A: A Listing of Program ND	35
APPENDIX B: A Listing of Program Spectra	48
APPENDIX C: A Description and Listing of the Program Spectra by Philip A. Baedecker	70

LIST OF FIGURES

Figure		Page
1	Wiring Schematic for Interface.	10
2	Typical Photopeak and Compton Edge.	19
3	ND Analyzer--Interface--IBM 1800 Cable Hook-Ups.	24

INTRODUCTION

Neutron activation analysis (NAA) is a powerful laboratory technique for determining the elemental content of an unknown sample. NAA is a particularly valuable tool in that it is often possible to identify trace amounts of elements in quantities of parts per million or even parts per billion. When the unknown sample is neutron irradiated, radioactive isotopes are formed. Many of the isotopes thus formed decay by gamma emission. Examination of the energy spectrum of the gamma rays emitted will identify the elements contained in the unknown sample. A multichannel analyzer coupled with a gamma ray radiation detector is used to collect the spectral data of the irradiated sample. Elemental identification is performed by analysis of the energies of the photopeaks observed in the collected spectrum. Analysis of the photopeak height, sample irradiation time, counting time, time between irradiation and data collection, sample weights or volumes, and detector efficiency will allow the determination of the quantity of each element present in the unknown sample. Another feature of NAA is its insensitivity to chemical form of the irradiated sample.

In order to more fully understand NAA, a few basic concepts will be discussed. An atomic nucleus is only stable when a specific number of protons and neutrons are present. The number of protons in the nucleus determines the element's identity and the number of neutrons usually determines whether that nucleus is radioactive or non-radioactive (stable). (It should be noted that there are some elements that have no stable nuclei. In some cases, there are other differences that make nuclei radioactive.)

As a sample case, all sodium atoms contain 11 protons, while only those sodium atoms containing 12 neutrons are stable. A sodium atom with fewer or with more neutrons will be radioactive. There are elements that have more than one number of neutrons that result in stability; for instance, tin has ten stable isotopes.

Nuclei can absorb additional neutrons, resulting in the conversion of stable nuclei to radioactive ones. The radioactive nuclei decay in unique ways and emit radiations that are often distinct and can be measured even in very small quantities. The measurement of the radiations can determine the kind and the number of radio-active atoms that are present.

The neutron bombardment of a sample is performed in a nuclear reactor where the neutrons that strike the target atoms have been slowed down so that they have very little energy of motion. These free neutrons react with

the target nuclei resulting in the capture of the neutron and thus creating nuclei with atomic weights one unit greater than the original atomic weight. These newly-created nuclei are typically radioactive.

Radioactive nuclei almost always decay by emitting negatively charged beta particles, usually accompanied by gamma rays. Each kind of radioactive atom decays with a unique pattern called a decay scheme. By measuring the radiation and identifying the pattern, the source of radiation can be determined. There are some problems in the measurement of gamma rays. These problems are due to the interaction of gamma rays with matter. Due to these interactions, the gamma rays measured by the detector no longer have the energy they started with at the time of emission. The most troublesome of the interactions is Compton scattering.

The Compton effect is a collision between a gamma ray and an electron. Compton scattering will generally involve the outer atomic electrons. When the gamma ray collides with the electron, it only transfers part of its energy. The gamma ray is degraded in energy and deflected from its original path. Compton scattering is the prevailing interference for measurement of gamma rays with energies greater than .5 MeV. Assuming that the radioactive isotope being detected has a 1 MeV gamma ray emission, due to the Compton effect, the detector will see gamma rays at 1 MeV

and a smear of gamma rays from 0 KeV to a value which is approximately 250 KeV less than the original gamma ray energy (for this case, 750 KeV). The end of the smear at 250 KeV less than the original gamma ray energy is called the Compton edge. This smear of gamma rays raises the background counts in the lower-energy gamma ray region. Therefore if low-energy gamma rays are being emitted from another radioactive isotope, the Compton effect from the 1 MeV gamma rays may mask them and make identification of the low-energy emitting isotope difficult. Interaction of gamma rays with matter may effect the sensitivity of the analysis, but the use of good gamma ray detectors can minimize the interference effects.

The applications for activation analysis seem unlimited. The only requirement for its use is that the element of interest must yield a gamma ray emitting isotope after irradiation. NAA has been used in the electronics industry to determine trace levels of impurities in semiconductors and the dopant levels in finished semiconductors. It has been useful in detecting trace amounts of oxygen in steel. NAA can be used to detect traces of catalyst residues in plastics. Determination of residual amounts of bromide in foodstuffs and on crops has assisted the agricultural industry.

NAA has provided a new tool for the forensic chemist. The results of activation analysis have been

admitted as evidence in a court of law. NAA can be used to trace a sample of paint, grease, tire rubber, etc. to its manufacturer by the quantities of trace elements present. It can provide a reliable method of identifying traces of gunpowder on a suspect's skin.

Another attractive use of NAA is the introduction of minor quantities of several easily activated elements as a secret coding to guard against counterfeiting of a product. This method of product identification is being considered for use in drugs, foodstuffs, cement and legal tender.

The topic of this paper will be the development and usage of a computerized method of neutron activation analysis utilizing the Michigan State University Triga Mark I nuclear reactor, a Nuclear Data multichannel analyzer, an IBM 1800 computer and the Michigan State University CDC 6500 computer. Prior to the development of this procedure, NAA was performed by manual techniques. A complex spectrum could take many hours to complete. Utilizing the methods described in this paper, spectrum analysis can be completed in a few minutes with a computer usage cost of a few dollars. Included in this paper will be a discussion of the electronic interface developed to allow data transfer from the Nuclear Data multichannel analyzer to the IBM 1800 computer, a discussion of the computer program developed to control the data transfer

and to present the data on computer punch cards, and a discussion of the spectral analysis computer program developed for the CDC 6500 computer. The final topic of this paper will be a user's manual giving instructions on performing NAA utilizing the facilities available at Michigan State University.

THE INTERFACE AND THE ND PROGRAM

Prior to the design of the interface and the development of the ND program, there were only two forms of data output for the Nuclear Data analyzer. The first, a pen plotter, provided rapid data output. Its disadvantage was that when peak height analysis was performed by calculating the area beneath the peak, the result was quite inaccurate. The second form of output available was through a Teletype with a paper tape punch. The Teletype output allowed peak height analysis with a high degree of accuracy, but to output a 2048 channel spectrum required approximately 30 minutes. Therefore, the first step necessary for a project that involved experiments using the Nuclear Data multichannel analyzer to count many samples was to find a method of data output that was both rapid and accurate. A bid request for an interface to connect the Nuclear Data analyzer to the IBM 1800 computer was sent to IBM and to Nuclear Data. The cost reported was in excess of the budget for the project. It was decided at this point that the interface design was to be performed in-house.

To develop the interface, the following information had to be ascertained:

(a) location of an output port on the Nuclear Data analyzer where memory bank data was available in some binary format;

(b) the binary format of the data at the output port (i.e.: is the data in pure binary (base 2) or in 1-2-4-8 binary coded decimal, where any four bits (on-off switches) represents one decimal digit. In four bits, "0" through "9" can be represented);

(c) voltages at the output port which represented 0 and 1;

(d) location of an input port on the Nuclear Data analyzer where the memory bank data at the output port could be advanced;

(e) voltage needed at the input port to trigger the data to advance as described in (d);

(f) location of an input port on the IBM 1800 where the data could be received;

(g) the voltages required at the input port to represent 0 and 1;

(h) the location of an output port on the IBM 1800 where an advance signal could be generated to trigger the data advance;

(i) voltages generated at the output port for 0 and 1 representation.

Many hours were spent researching the operation and maintenance manuals for both the Nuclear Data analyzer and the

IBM 1800. This research, coupled with extensive experimentation, yielded the necessary information.

It was found that at the connector labeled "Printer A" on the Nuclear Data analyzer, the memory bank data could be read in 1-2-4-8 binary coded decimal form with voltages pulled down to within 0.5 volts of ground for "zero" signals and up to +6 volts or floating for "one" signals. It was also discovered that the memory bank could be advanced by simulating a "complete" signal of +12 volts. This was the voltage that the plotter generated to indicate that data had been received and that the next channel of data should be prepared for transfer. The digital inputs or "DI's" on the IBM 1800 were determined to be the best port for data input and the digital outputs (called "DO's" or "ECO's") were found to be the source for generation of the trigger signal. It was determined that for the IBM 1800, a zero was represented by -12 volts (with a range of -6 to -30 volts), and a one by 0 volts (with a range of -1 to +30 volts).

An electronic interface was required to adjust the voltages to conform with the needs of the two devices being connected. The Michigan State University Department of Engineering Research Computer Laboratory developed an interface box powered by the Nuclear Data analyzer NIM bin that provided the required voltage conversions (see Figure 1). It is important to note here that the voltage

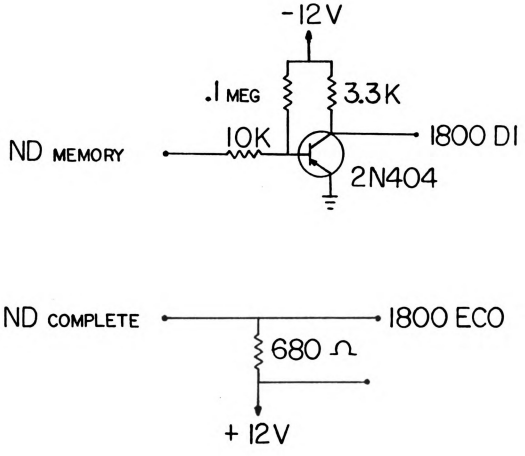


Figure 1.--Wiring Schematic for Interface.

conversions performed by the interface only shift the voltages; they are not reversed. Therefore a "1" generated by the ND analyzer, with voltage shifted by the interface, yields a "0" at the IBM 1800, and a "0" generated by the ND analyzer, with voltage shifted by the interface, yields a "1" at the IBM 1800. The correcting inversion is performed by program ND as the data is transferred.

Once the hardware was developed to enable the IBM 1800 and the Nuclear Data analyzer to communicate with each other, it was necessary to create a program to control the transfer of data from one device to the other. Program ND was written to perform this function. (A listing of ND is provided in Appendix A). Program ND is an interactive program written in Fortran IV. The program elicits information from the user, such as the number of data channels to be transferred and whether or not a line printer listing of the spectrum is desired. The program also instructs the user, with commands to load the card hopper when necessary and to make sure the ND analyzer is properly prepared for data transfer.

Program ND consists of seven parts. The first of these parts is the main program. The main program controls the operation of the rest of the code. The responses of the user to the questions asked by the computer control the title of the experiment, the run number and the form

of output. The main program calls upon the six subprograms listed below:

(1) Subroutine PRINT is called when the user requests a listing of the data transferred. This listing is outputted to the line printer. Subroutine PRINT converts the binary data to the equivalent character code and fills in leading blanks instead of zeros as needed (when the ND analyzer sends the data in a channel, six digits are transmitted. A count of 900 is sent as 000900). The program also prints the number of the first channel in each row of data listed.

(2) Subroutine PPUNC is called when the user requests a deck of cards containing the data transferred. The deck is punched in binary coded decimal form. A BCD deck contains ten channels of data on each card. Each column on the card contains either a blank or one numerical character. Subroutine PPUNC performs the same conversion from binary that subroutine PRINT performs, but no "first channel" is punched on the cards.

(3) Function MINO is called by the main program to calculate which of two numbers is the minimum.

(Note: The following three subroutines are written in IBM 1800 Assembly language. Because of the necessity to do bit manipulation, the use of Assembly language is more practical and efficient than the use of Fortran.)

(4) Subroutine DI is called by the main program to read the digital inputs from the ND analyzer and to send the "complete" signal to the ND analyzer to advance the memory stack. Subroutine DI also performs the conversion of the input bits from 0 to 1 and 1 to 0. This routine also loads the 1-2-4-8 BCD data into a special portion of the IBM 1800 computer memory called "COMMON," converts the 1-2-4-8 BCD data digits to binary and loads the binary data into "COMMON" also.

(5) Subroutine ECO is called by the main program to ensure that the digital output port is initialized prior to the ND analyzer being turned to the "Read in/out" mode. This is required so that the first channel of data is not missed.

(6) Subroutine CRUNC is called by the main program when the user requests a "crunch" deck. A crunch deck contains 40 channels per card. The data is in binary form. Each column on the card contains 12 bits of information. Since each word or channel of data requires a maximum of 16 bits when converted to binary, each word requires $1\frac{1}{3}$ columns. In other words, every four columns contains three data channels. This format is similar to the "crunch" format used in the MSU Cyclotron Laboratory. The CDC 6500 remote input station located in the engineering building is not capable of transmitting binary data to the computer. The spectrum analysis routine, SPECTRA,

available on the CDC 6500 is not provided with the ability to handle the "crunch" decks. The "crunch" feature is provided here in Program ND, but further development of Spectra would be necessary in order to make use of this feature.

SPECTRA

Program SPECTRA is used to analyze the data decks generated by the IBM 1800 and program ND. Spectra was originally created by Philip A. Baedecker. A description and listing of the program is provided in Appendix C. Spectra was then brought to the Michigan State University Cyclotron Laboratory and modified to operate on the Sigma 7 computer. This conversion was performed by Dr. Charles Spooner of the Geology department along with members of the cyclotron staff. The original Spectra was designed to accept tape input of the spectrum. The cyclotron version of Spectra utilized a compressed binary format ("crunch") deck of data cards for input. To allow use of Spectra with the data generated by the IBM 1800 computer and program ND, the computer code was modified with the assistance of Robert Morris, a computer science student at MSU. As a result of this modification, campus-wide use of Spectra through the CDC 6500 computer became possible. The CDC 6500 version of Spectra is designed to accept BCD decks of input data as provided by the IBM 1800 and program ND. A listing of the CDC 6500 version of Spectra is provided in Appendix B.

Spectra is a computer program that analyzes gamma ray spectrum by searching out photopeaks, calculating the area under the peaks, finding the centroid of each peak and providing the energy corresponding to each peak. In activation analysis runs, when quantitative information is required, the program provides either the concentrations or the quantity of the elements in the unknown samples. Control of the program is accomplished when the user places certain control cards in front of the data decks to be analyzed. A detailed set of instructions for use of Spectra can be found in Chapter 4, the User's Manual.

The following procedure outlines the method by which the program analyzes the data. After reading in the control cards, the program reads in a data deck. The data can be smoothed by a five-point convolution method. Since smoothing can cause undershoot of the baseline on either side of very narrow peaks, smoothing should be avoided for spectrum with 1024 channels or fewer, but for spectrum with 2048 channels or more, smoothing helps to eliminate statistical fluctuations that might be confused with actual photopeaks.

The program then performs a series of operations to locate peaks. The first of these operations is a five-point convolution method of calculating first derivatives for each channel. Then, by looking for sign changes in the first derivatives, the program locates possible peaks.

The program assigns the minimum on each side of the peak as the peak limits or boundaries. The program then checks to verify that the right hand boundary channel for the peak is not too far out. By comparing the left side derivatives to the right side derivatives, the right hand boundary channel is fixed. A baseline calculation is performed next, and then a few statistical tests of peak validity are done. The first test for a valid peak involves consideration of the following criterion:

$$\text{Peak Area} > (5.0 \times \text{SIGMA})$$

where peak area is total area minus base area, and SIGMA represents the standard deviation of the peak. The calculation of SIGMA is as follows:

$$\text{SIGMA} = \sqrt{\sum_{L}^{R} D_i + (D_L + D_R) \times \left[\left(\frac{R-L-1}{2}\right)^2 - 1\right]}$$

where R represents the right hand boundary channel number, L represents the left hand boundary channel number, D_L represents the number of counts in the left hand boundary channel, D_R represents the number of counts in the right hand boundary channel, and D_i represents the number of counts in the i^{th} channel.

The next test is used to eliminate Compton edges. The slope of the left side of the peak is compared to the slope of the right side of the peak. If the absolute

values of the slopes are within a factor of 1.5, the peak is passed to the baseline calculator. By inspection of Figure 2, one can see that a Compton edge would not pass the aforementioned "1.5" test. Four channels to the outside of the peak boundaries are averaged, with those channels whose count is more than two standard deviations from the peak boundary channel count being disregarded. The right hand and left hand averages are taken as the counts in the boundary channels. The right hand and left hand boundary channels are straight-lined and a baseline is determined. The baseline is defined as the background level. Finally, the centroid of the peak is calculated.

The user has the option of two different methods of determining peak intensity. The "Total Peak Area" method yields a peak intensity equal to the sum of the counts in the channels bounding the peak as determined by the statistical tests described above. This method uses the redefined counts in the boundary channels for subtracting out background. The second peak intensity method available is the "Wasson" method. In this method, the user specifies the number of channels to either side of the center channel of the peak to be used as the limits of integration. A baseline is determined as described above, but with the boundary channels as defined by the user, and the background is then subtracted out.

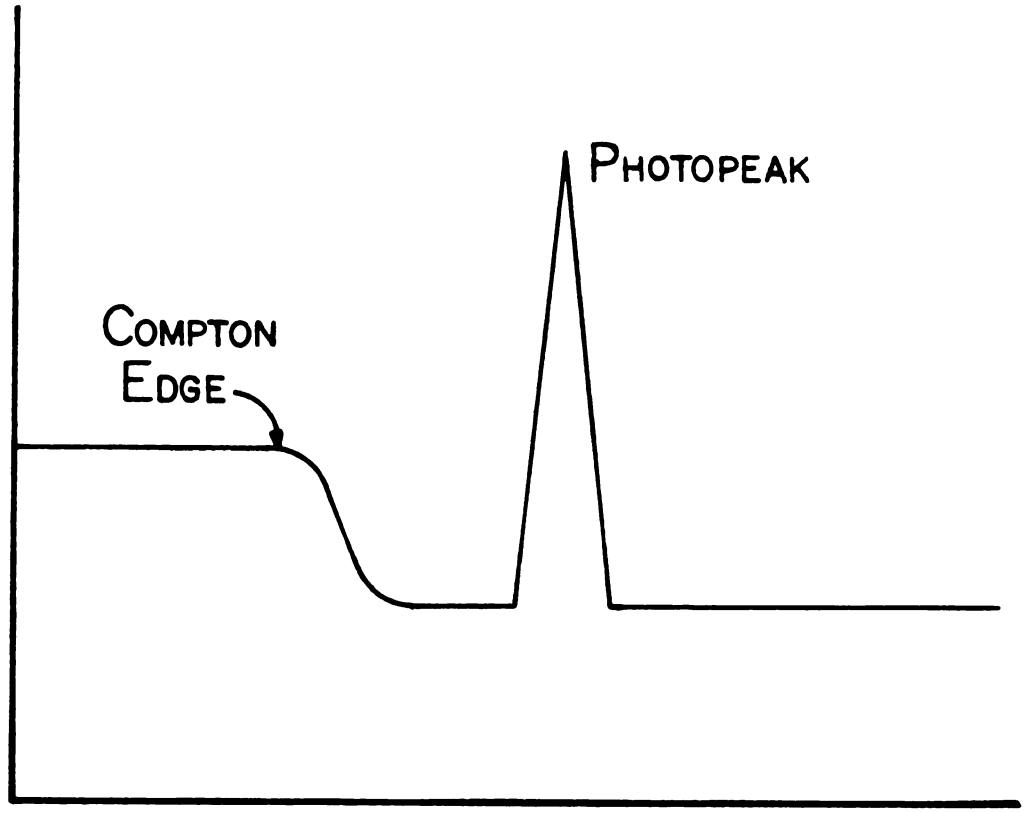


Figure 2.--Typical Photopeak and Compton Edge.

In order to calibrate the spectra (i.e.: assign energy values to each channel), reference and standard spectra must be provided to the program. First, a rough calibration is performed by using a ^{137}Cs spectrum. ^{137}Cs has a single peak located at 661.7 KeV. If photopeaks greater than 1836 KeV are to be located, a ^{88}Y spectrum is also necessary, to assist the program in compensating for possible nonlinearity in the spectrometer. The reference spectra addressed above may be deleted if the user supplies the program with the centermost channels for ^{137}Cs and ^{88}Y. Standard spectra are used next to give a precise calibration to the spectra. The user supplies the program with the energy levels of the gamma-rays in each standard spectrum. The program assigns the centroid of each peak located in the spectrum with the corresponding energy. Then the program determines energies for the photopeaks of the "unknown" sample spectra by interpolation assuming linearity between standard spectrum peaks. The program will not extrapolate energies above the highest standard peak supplied by the user. This completes the qualitative analysis portion of Spectra.

Spectra is also capable of performing activation analysis calculations (or quantitative analysis). A flux monitor (i.e.: sample containing a known quantity of the element of interest) must be irradiated in the reactor in the same batch as the "unknown" samples. The user must

provide the program with the gamma ray energy of the element of interest, the chemical symbol, the half life of the isotope and the weight or concentration of the flux monitor sample. The user must also supply name of sample, weight of sample, time of day sample was counted and live time duration of count for each spectrum used in the quantitative analysis, both "unknowns" and "flux monitors." The program then uses the flux monitor spectra to calibrate each peak of interest.

The "unknown" sample spectra are then analyzed for each peak of interest. Each peak is compared to the associated flux monitor peak and from sample weights the concentration of each element of interest is determined. In order to account for decay and live time the following formulae are used:

$$\text{Formula 1:} \quad SCF = \frac{(\text{peak area}) \times \lambda e^{-\lambda t_1}}{(\text{flux monitor wt.}) \times (1 - e^{-\lambda t_2})}$$

where SCF is the standard comparator factor or the calibration of the flux monitor peak,

λ is the decay constant for the isotope of interest,

t_1 is the time elapsed between some reference time (e.g.: time of irradiation or time of first count in this analysis) and the start time of the counting of this spectrum, and

t_2 is the duration of the count (live time).

$$\text{Formula 2:}\quad ACF = \frac{(\text{peak area}) \times \lambda e^{-\lambda t_1}}{(\text{sample wt.}) \times (1 - e^{-\lambda t_2})}$$

where ACF is the analytical comparator factor, or the weighted intensity of the "unknown" sample peak of interest, and λ, t_1 and t_2 are as described above, but evaluated for the "unknown."

For a more detailed discussion of Spectra, see Appendix C.

USER'S MANUAL

This chapter describes the procedures for transferring data for the ND analyzer to the IBM 1800 and for operating program Spectra. To transfer data to the IBM 1800, the ND analyzer, the "interface" and the IBM 1800 must be connected. The IBM 1800 must be cold started and program ND must be run. Once the data has been collected on data decks, the controls for Spectra must be prepared and finally Spectra is run. The procedure for these operations follows.

Data Transfer Hook-Up (See Figure 3)

(1) The following connections must be made:

(a) The interface must be in a nimbin;

(b) The cable from the back of the interface must be connected to the ND analyzer at "Printer A";

(c) The female single pin connection in the back of the interface must be connected to the ND analyzer at the point marked "complete";

(d) The cable from the IBM 1800 must be attached to the face of the interface;

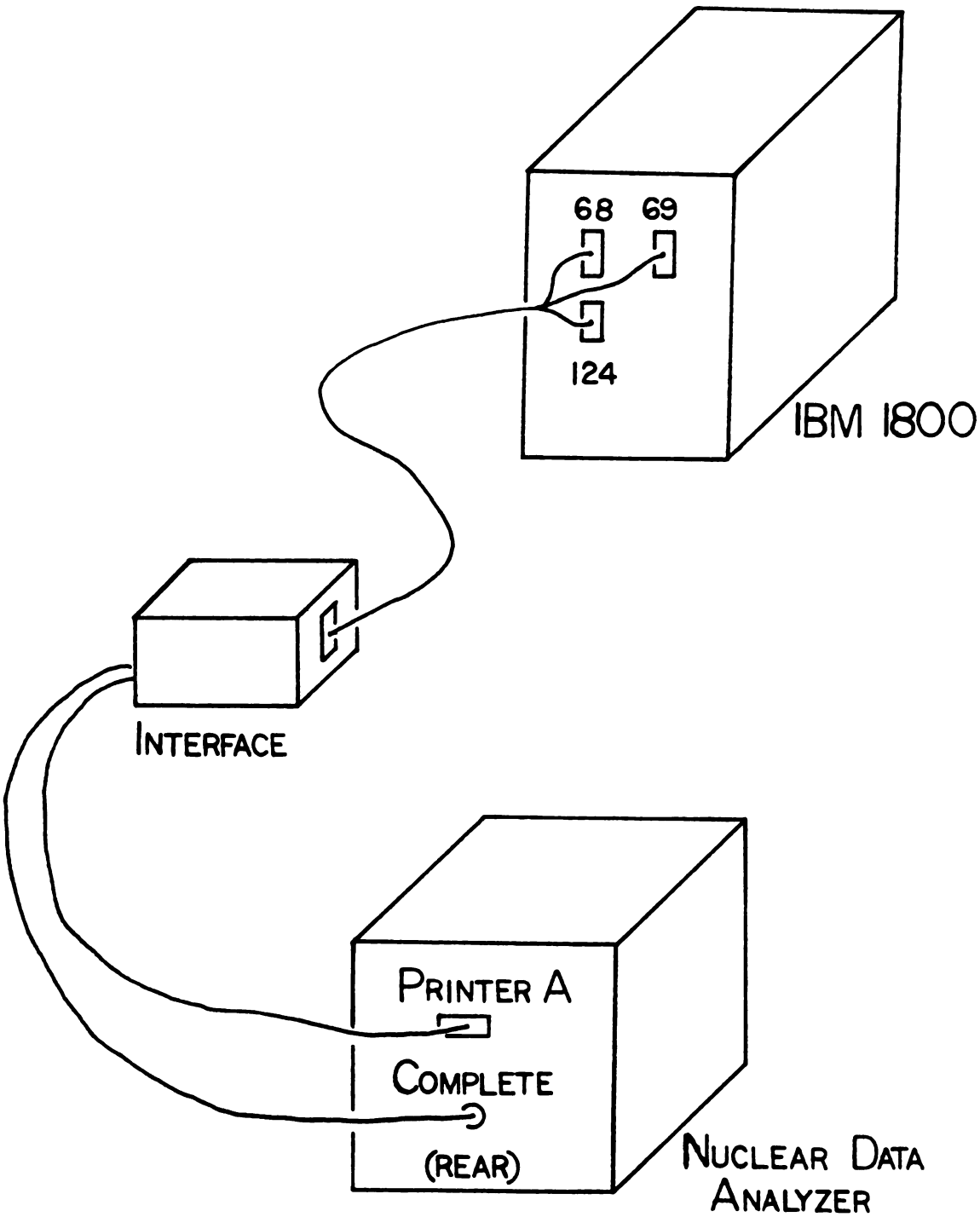


Figure 3.--ND Analyzer--Interface--IBM 1800 Cable Hook-Ups.

(Note: the following connections are in the IBM 1800 room.)

(a) Connect cable connectors marked "ND-68" and "ND-69" to the IBM 1800 at the digital inputs marked "68" and "69";

(b) Connect cable connector marked "ND-124" to the IBM 1800 at digital output "124".

Cold Start of the IBM 1800

(1) On console--push POWER ON.

(2) On printer--push START.

(3) Place disk marked "Reactor Lab" in disk drive (middle slot).

(4) On disk drive--push START.

(5) Wait for READY light on disk drive.

(6) Place cold start card in hopper.

(7) On console--push IMMEDIATE STOP.

(8) On console--push and hold CLEAR STOR.

(9) On console--push START.

(10) On console--release START and CLEAR STOR.

(11) On console--push IMMEDIATE STOP.

(12) On card reader--push START.

(13) On console--push PROGRAM LOAD.

(14) Wait for cold start pattern. The cold start pattern appears on the operating console. Most of the small white lights will be lit except for four lights forming a square near

the center of the light panel. These are the lights corresponding to columns 9 and 10 in the adress register and I register rows.

To Run ND

(1) Prepare the following deck: (all cards begin in col. 1).

```
//JOB
/// XEQ ND      FX  (Note:  FX begins in col. 16)
blank card
blank card
```

(2) Place deck in hopper.

(3) On card reader--push START.

(4) ND is self-explanatory. The program will request information from the user on the IBM selectric terminal in the IBM 1800 room.

Preparing to Run Spectra

Note: For each "card" a format is prescribed. There are four types of formats, "A", "I", "F", and "X". Formats "A" and "I" have the following structure: nAm or nIm. The value of "n" gives the repetition number (e.g.: 3A2 is the same as A2, A2, A2). The value of "m" gives the number of characters allowed in the input field. The "A" format allows the input of alphanumeric characters (e.g.: A4 allows the input of TEST). The "I" format is used to input numeric data in integer form (whole numbers). Note that data in an "I" field must be right justified. If the data to be inputted in an I6 field is three digits

long, the three digits must be preceded by three blanks (e.g.: 126 in an I6 field must be bbb126, where b represents blank). Format "X" has the following structure: nX. The "X" indicates a blanking format, where "n" is the number of blanks. The "F" format has a structure of: nFm.p where "n" is the repetition number, "m" is the total number of characters allowed in the input field, and "p" is the maximum number of digits to the right of the decimal point. The "F" format is used to input numeric data that is not whole number data. Note that the data need not be right justified in the input field, and that "m" includes the decimal point (e.g.: 125.89 requires a format of F6.2).

In order to run Spectra, the following deck must be prepared:

Card 1--Format 4I6

columns 1-6	(1)	Number of standardizing spectra (including Cs and Y standards, if provided)
columns 7-12	(2)	Maximum energy of standard peaks (in Kev).
columns 13-18	(3)	Cs peak channel number (if spectrum provided, input "0").
columns 19-24	(4)	Y peak channel number (if not used, input "0").

(Note: If max. energy of standards is greater than 1836.08 KeV, a Y spectrum or channel number is required.)

Card 2--Format 12I6 (Multiple cards using the same format may be required, depending on the number of standard spectra used.)

Cards with number of peaks in each standard.

(Sequence must correspond with order in which

standard spectrum are run. Not needed for Cs

or Y data).

Card 3--Format 8F9.3 (Multiple cards using the same format may be required, depending on the total number of peaks in the standard spectra used.)

Cards with energies of peaks in each standard.

(In the order in which they appear in the

decks, i.e.: min. to max. for each spectrum

in order.)

Card 4--Format 3I6 and then 13A4

column 6	(1) "0"
columns 7-12	(2) Number of channels in spectra
columns 13-18	(3) Number of spectra to be analyzed (including flux monitors, but not standards)
columns 19-70	(4) Labels (run name). A maximum of 52 alphanumeric characters are allowed.

Card 5--Format 4I1 and then 1I6

column 1	(1) 0--raw data printed out 1--raw data not printed out

column 2	(2)	0--smoothed data printed out 1--smoothed data not printed out 2--smoothing operation omitted
column 3	(3)	0--peak area by Wasson's method 1--peak area by total peak area method
column 4	(4)	0--activation analysis section used 1--activation analysis section not used
columns 5-10	(5)	Integer number of points on each side of the center channel, used for peak area.

(Note: If card 5 col. 4 = 1, omit card 6 and those following.)

Card 6--Format 3I6 and then A4

columns 1-6	(1)	Number of flux monitor spectra. (Note: multiple flux monitors have to have identical concentrations or quantities, only in different locations in reactor core during irradiation. The program does not accommodate different concentration flux monitors. Spectra merely averages the peak areas in multiple flux monitors.)
columns 7-12	(2)	Number of peaks to be analyzed.
columns 13-18	(3)	Amount of error in peak find--approximately three KeV (as recommended in the Baedecker write-up, see Appendix C.)

columns 19-22 (4) Four alphanumeric characters that identify the flux monitor spectra.

Card 7--One for each peak analyzed

columns 1-2 (1) Format A2--Chemical symbol being determined

columns 3-10 (2) Format 8X--Space 8

columns 11-20 (3) Format F10.3--Energy of peak in KeV.

columns 21-23 (4) Format 3X--Space 3

columns 24-30 (5) Format F7.3--Half-life

columns 31-34 (6) Format 4X--Space 4

column 35 (7) Format A1--Units of half-life (S,M,H,D,Y)

columns 36-40 (8) Format 5X--Space 5

columns 41-50 (9) Format F10.5--Flux monitor weight or concentration

columns 51-56 (10) Format 6X--Space 6

columns 57-60 (11) Format A4--Units of flux monitor weight or concentration (µg, mg, etc.)

Card 8--One for each spectrum in deck (to be analyzed for activation analysis)

columns 1-16 (1) Format 4A4--Sample name (The program requires that the first four characters must be the same as in col. 19-22 if spectrum is to be treated as a flux monitor.)

columns 17-20 (2) Format 4X--Space 4

columns 21-30 (3) Format F10.5--Sample weight

columns 31-36	(4) Format 6X--Space 6
columns 37-40	(5) Format A4--Units of weight (μg, mg, etc.)
columns 41-45	(6) Format 5X--Space 5
columns 46-48	(7) Format I3--hours, time of count, 2400 hour clock, if next day, add 2400
columns 49-50	(8) Format I2--Minutes
columns 51-53	(9) Format 3X--Space 3
columns 54-59	(10) Format I6--Live time duration of count
column 60	(11) Format A1--Units of time
columns 61-64	(12) Format 4X--Space 4
Column 70	(13) Format I6--"0"

Follow cards 1-8 with the data decks from the IBM 1800 as follows:

(1) Cs spectrum if used;

(2) Y spectrum if used;

(3) Standard spectra;

(4) Flux monitor spectra if doing activation analysis;

(5) Unknown spectra.

(Note: Place 789 multipunch card at the end of each spectrum and place 6789 multipunch card at end of deck.)

Running Spectra

(1) Prepare the following deck (each card begins in col. 1)

PNC

JOB CARD (Be sure to specify MT1 to mount tape)

PW=

HAL, BANNER, SPECTRA.

DISPOSE, **, V.

APLIB, TT1313, *SPECLAB.

MAP, OFF.

SPECLAB.

789

(2) Follow this deck with the Spectra cards as described above.

BIBLIOGRAPHY

BIBLIOGRAPHY

General References

1. Friedlander, G., Kennedy, J.W., and Miller, J.M. Nuclear and Radiochemistry. Second Edition, John Wiley and Sons, Inc. New York, N.Y., 1964.

2. Foster, A.R. and Wright, R.L., Jr. Basic Nuclear Engineering. Third Edition, Allyn and Bacon, Inc. Boston, Mass, 1977.

3. Gardner, R.P. and Ely, R.L., Jr. Radioisotope Measurement Applications in Engineering. Reinhold Publishing Corp. New York, N.Y., 1967.

4. Grouthamel, C.E., Adams, F. and Dams, R. Applied Gamma-Ray Spectrometry. Second Edition, Pergamon Press. Oxford, England, 1970.

APPENDIX A

A LISTING OF PROGRAM ND

```
NOMPROCESS DISK NO.1
NO4 READY READER


// JOB
// FOR NO
*NOMPROCESS PROGRAM
*IOCS(1443 PRINTER,CARD,DISK,KEYBOARD,TYPEWRITER)
*ONE WORD INTEGERS
*LIST ALL
      INTEGER TITLE(20),DIG1,DIG2,DIG3,DIG4,Y
      INTEGER CR
      COMMON IDATA(2,160), MDATA(2,160)
      DATA Y,CR/'Y','CR'/
      DEFINE FILE1(26,320,U,K)
      DEFINE FILE2(26,320,U,L)
      WRITE(1,100)
  100 FORMAT(/,'PROGRAM NO TRANSFERS DATA FROM THE NO ANALYZER TO THE IB
     1M-1800',/,'THE PROGRAM WILL BE SELF EXPLANATORY.',//,'MAKE SURE THE
     2 NO IS IN THE STOP MODE',/,'AND THE INTERFACE IS LINKED UP AND ON.
     3-THEN TYPE-GO-')
      READ(6,312) DUMMY
  312 FORMAT(A2)
      CALL EGO
      WRITE(1,314)
  314 FORMAT(/,'TYPE IN THE NUMBER OF CHANNELS OF DATA IN I4 FORMAT')
      READ(6,315) NCHAN
  315 FORMAT(I4)
      N=(NCHAN+159)/160
      WRITE(1,313)
```

```
313 FORMAT(/,'TURN NO TO THE READ IN/OUT MODE AND MAKE SURE THE TRANSF
   1ER MODE SWITCH',/,' IS IN THE PRINTER MODE, THEN TYPE-GO-')
    READ(6,312) DUMMY
    K=1
    L=1
    DO 500 I=1,N
    CALL DI
    WRITE(1'K) IDATA
    WRITE(2'L) MDATA
500 CONTINUE
    WRITE(1,318)
318 FORMAT(/,'DATA TRANSFER IS COMPLETE AND THE NO CAN BE STOPPED',/,'
   1OR STARTED COUNTING AGAIN.',/,'WHEN READY TYPE-GO-')
    READ(6,312) DUMMY
    WRITE(1,316)
316 FORMAT(/,'LIST THE RUN NUMBER IN I4 FORMAT')
    READ(6,317) DIG1,DIG2,DIG3,DIG4
317 FORMAT(4I1)
    WRITE(1,330)
330 FORMAT(/,'INPUT TITLE IN 20A2 FORMAT--IE. 40 SPACES')
    READ(6,331)(TITLE(J),J=1,20)
331 FORMAT(20A2)
    WRITE(1,320)
320 FORMAT(/,'PRINT OUT ON THE 1443 PRINTER  TYPE-Y- FOR YES OR -N- FO
   1R NO.')
    READ(6,321)IPRIN
321 FORMAT(A1)
    IF(IPRIN-Y)101,102,101
102 WRITE(3,700)DIG1,DIG2,DIG3,DIG4,(TITLE(J),J=1,20)
700 FORMAT(/,' RUN NUMBER',2X,4I1,6X,20A2)
    L=1
    DO 501 I=1,N
    M=MINO(NCHAN-(I-1)*160,160)
    READ(2'L)MDATA
    CALL PRINT(M)
501 CONTINUE
101 WRITE(1,322)
322 FORMAT(/,'DATA PUNCHED ON CARDS  TYPE-Y- FOR YES OR -N- FOR NO.')
    READ(6,321)IPUNC
    IF(IPUNC-Y)103,104,103
103 IPRU=1
    GO TO 900
104 WRITE(1,324)
324 FORMAT(/,'DATA PUNCHED IN BCD OR CRUNCH  TYPE-BC- FOR BCD OR -CR-
   1FOR CRUNCH.')
    READ(6,312)IPUNC
    IF(IPUNC-CR)105,106,105
106 IPRU=2
    GO TO 107
105 IPRU=1
107 GO TO(108,109),IPRU
108 NC10=NCHAN/10+1
    WRITE(1,326)NC10
326 FORMAT(/,'LOAD THE HOPPER-1442 CARD READER PUNCH- WITH APPROXIMATE
   1LY',/,I4,' BLANK CARDS, THEN PUSH START AND TYPE-GO-')
    READ(6,312) DUMMY
    READ(2,312) DUMMY
    CALL BLKCK
    WRITE(2,701)DIG1,DIG2,DIG3,DIG4,(TITLE(J),J=1,20)
701 FORMAT('*C',26X,'RUN ',4I1,4X,20A2)
    L=1
    DO 502 I=1,N
    M=MINO(NCHAN-(I-1)*160,160)
    READ(2'L) MDATA
    CALL FPUNC(M)
502 CONTINUE
    GO TO 900
109 NC40=NCHAN/40+1
    WRITE(1,328)NC40
328 FORMAT(/,'LOAD THE HOPPER-1442 CARD READER PUNCH- WITH APPROXIMATE
   1LY',/,I4,' BLANK CARDS, THEN PUSH START AND TYPE-GO-')
    READ(6,312) DUMMY
    READ(2,312) DUMMY
    CALL BLKCK
    WRITE(2,701)DIG1,DIG2,DIG3,DIG4,(TITLE(J),J=1,20)
    K=1
    DO 503 I=1,N
    READ(1'K)IDATA
    CALL CRUNC(NCHAN,DIG1,DIG2,DIG3,DIG4)
    M=I*160
    IF(M-NCHAN)503,900,900
```

```
  503 CONTINUE
  900 WRITE(1,329)
  329 FORMAT(/,'EXECUTION HAS BEEN COMPLETED, THANK YOU.')
      CALL EXIT
      END
VARIABLE ALLOCATIONS
 IDATA(IC)=FFFF-FFC0 MDATA(IC)=FFBF-FF80 DUMMY(R )=000C       TITLE(I )=0023-0010   DIG16( )=0024        DIG2(I )=0025
  DIG3(I )=0026        DIG4(I )=0027          Y(I )=002B         CR(I )=0029          K(I )=002A          L(I )=002B
 NCHAN(I )=002C         M(I )=0020           I(I )=002E         J(I )=002F     IPRIN(I )=0030          N(I )=0031
 IPUNC(I )=0032       IPRO(I )=0033       NC10(I )=0034      NC40(I )=0035

STATEMENT ALLOCATIONS
  100  =0041  312  =00AB  314  =00AA  315  =00C7  313  =00C9  318  =0104  316  =013C  317  =014F  330  =0152  331  =016A
  320  =0160  321  =018F  700  =0191  322  =01A0  324  =01BF  326  =01E3  701  =021B  328  =0227  329  =025F  500  =0280
  102  =0309  501  =0340  101  =0356  103  =0366  104  =036C  106  =037C  105  =0382  107  =0386  108  =038C  502  =03ED
  109  =03FB  503  =0457  900  =0460

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 FCD      DI       MINO     PRINT    BLKCK    FPUNC    CRUNC    COMGO    STEAC    SHEAC    MRED     MWRT     MCOMP    MIDIX    MIOF
 MIOI     SUBSC    MORED    MOWRT    MOCOM    MDAT     TYPEN    HOLER    PRNIN    FRRRT    CARDN

INTEGER CONSTANTS
      1=003A       6=0039     159=003A     160=003B       2=003C      20=003D       3=003E      10=003F      40=0040

CORE REQUIREMENTS FOR ND
 COMMON     640  INSKEL COMMON       0  VARIABLES     56  PROGRAM   1070

 END OF COMPILATION
```

```
NO
DUP FUNCTION COMPLETED
// FOR PRINT
*NONPROCESS PROGRAM
*ONE WORD INTEGERS
*LIST ALL
      SUBROUTINE PRINT (M)
      INTEGER PRIN(4,2,10), CHAR(10), BLNK, BLANK(8,10)
      COMMON IDATA(2,160), MDATA(2,160)
      EQUIVALENCE (PRIN(1,1,1), BLANK(1,1))
      DATA CHAR /'0','1','2','3','4','5','6','7','8','9'/
      DATA BLNK/' '/
      DATA I/0/
      DATA LINE /0/
      DO 200 I4=1,M
      I=I+1
      IF(I-11) 1,2,2
    2 WRITE(3,4) LINE,PRIN
      LINE=LINE+10
      I=1
    1 DO 100 I3=1,2
      DO 100 I1=1,4
      I2= IAND(15,LSHFT(MDATA(I3,I4),4*I1))+1
  100 PRIN  (I1,I3,I)= CHAR(I2)
      DO 101 I1=1,7
      IF(BLANK(I1,I)-CHAR(1)) 500,501,500
  501 BLANK(I1,I)= BLNK
  101 CONTINUE
  500 CONTINUE
  200 CONTINUE
      IF(10-I)300,300,301
  301 WRITE (3,4)LINE,((BLANK(I1,I2),I1=1,8),I2=1,I)
    4 FORMAT (1X,I4,2X,80A1)
  300 RETURN
      END
VARIABLE ALLOCATIONS
 IDATA(IC)=FFFF-FFC0 MDATA(IC)=FFBF-FF80  PRIN(I )=004F-0000 BLANK(I )=004F-0000  CHAR(I )=0050-0054  BLNK(I )=005F
    I4(I )=005F           I(I )=0060        LINE(I )=0061          I2(I )=0062          I1(I )=0063          I3(I )=0064

STATEMENT ALLOCATIONS
 4    =0071  2    =008C  1    =009F  100  =00C8  501  =0101  101  =010C  500  =0115  200  =0115  301  =0124  300  =014F


                                                                    PAGE 02

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 IAND    LSHFT   ISTOX   MWRT    MCOMP   MIOAT   MIOIX   MIOI    SUBSC   SUBIN

INTEGER CONSTANTS
     1=0068     11=0069      3=006A     10=006B      2=006C      4=006D     15=006E      7=006F      8=0070

CORE REQUIREMENTS FOR PRINT
 COMMON     640  INSKEL COMMON       0  VARIABLES    104  PROGRAM    230

 END OF COMPILATION
```

```
PRINT
DUP FUNCTION COMPLETED
// FOR FPUNC
*NONPROCESS PROGRAM
*ONE WORD INTEGERS
*LIST ALL
      SUBROUTINE FPUNC(M)
      INTEGER PRIN(4,2,10), CHAR(10), BLNK, BLANK(8,10)
      COMMON IDATA(2,160), MDATA(2,160)
      EQUIVALENCE (PRIN(1,1,1), BLANK(1,1))
      DATA CHAR /'0','1','2','3','4','5','6','7','8','9'/
      DATA BLNK/' '/
      DATA I/0/
      DO 200 I4=1,M
      I=I+1
      IF(I-11) 1,2,2
    2 WRITE(2,4) PRIN
      I=1
    1 DO 100 I3=1,2
      DO 100 I1=1,4
      I2= IAND(15,LSHFT(MDATA(I3,I4),4*I1))+1
  100 PRIN  (I1,I3,I)= CHAR(I2)
      DO 101 I1=1,7
      IF(BLANK(I1,I)-CHAR(1)) 500,501,500
  501 BLANK(I1,I)= BLNK
  101 CONTINUE
  500 CONTINUE
  200 CONTINUE
      IF(10-I)300,300,301
  301 WRITE (2,4)((BLANK(I,I2),I1=1,8),I2=1,I)
    4 FORMAT(80A1)
  300 RETURN
      END
VARIABLE ALLOCATIONS
 IDATA(IC)=FFFF-FFC0 MDATA(IC)=FFBF-FF80  PRIN(I )=004F-0000 BLANK(I )=004F-0000  CHAR(I )=0050-0054  BLNK(I )=005F
    I4(I )=005F          I1(I )=0060          I3(I )=0061          I1(I )=0062          I2(I )=0063

STATEMENT ALLOCATIONS
 4     =0070  2     =0088  1     =0093  100   =00BC  501   =00E5  101   =0100  500   =0109  200   =0109  401   =011A  300   =0140

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 IAND    LSHFT   ISTOX   MWRT    MCOMP   MIOAT   MIOIX   SUBSC   SUBIN

INTEGER CONSTANTS
     1=0068     11=0069      2=006A      4=006B     15=006C      7=006D     10=006E      8=006F

CORE REQUIREMENTS FOR FPUNC
 COMMON     640  INSKEL COMMON       0  VARIABLES    104  PROGRAM     216


 END OF COMPILATION
```

```
FPUNC
DUP FUNCTION COMPLETED
// FOR MINO
*NONPROCESS PROGRAM
*ONE WORD INTEGERS
*LIST ALL
      FUNCTION MINO(N,M)
      MINO=M
      IF(M-N)2,2,1
    1 MINO=N
    2 RETURN
      END
VARIABLE ALLOCATIONS
  MINO(I )=0000

STATEMENT ALLOCATIONS
  1     =0015   2     =0019

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 SUBIN

CORE REQUIREMENTS FOR MINO
 COMMON       0  INSKEL COMMON       0  VARIABLES       2  PROGRAM      2A

 END OF COMPILATION
```

```
MIND
DUP FUNCTION COMPLETED
// ASM DI
  *LIST ALL
```



```
0000     04240000          ENT     DI
0000 0   0000        DI    DC      0
0001 01  DC000074          STD   L  TEMP    SAVE A-Q REGISTERS
0003 0   6956              STX     1 XR1[1]   STORE XR1
0004 0   6A57              STX     2 XR2[1]   STORE XR2
0005 01  74000000          MDX   L  DI,0     SETS UP RETURN
0007 00  66000140          LDX   L2 320
0009 0   1000        LOOP2 NOP               COUNTER IS IN XR2
000A 01  C4000076          LD    L  ZERO
000C 01  D4000068          STD   L  VAL1
000E 0   0857              XIO     [1]
000F 0   F060              EOR     NEG    CONVERT LOGIC
0010 0   1800              RTE     1A
0011 0   0856              XIO     IN2
0012 0   F050              EOR     NEG    CONVERT LOGIC
0013 0   1800              RTE     1A
0014 00  DE00FD7F          STD   L2 -642    STORE IN COMMON
0016 0   1800              RTE     1A
0017 01  DC000078          STD   L  HOLD    STORE FOR CONVERSION
0019 0   0846              XIO     UP
001A 0   0847              XIO     DN
001B 01  C4000078          LDD   L  HOLD
0010 01  D4000078          STD   L  SAVE    SAVE THE TOP TWO DIGITS
```

```
001F 0  1800              RTE     1A
0020 01 D4000078          STO   L HOLD   SAVE BOTTOM FOUR DIGITS
0022 0  6104              LDX   1 4    SET UP CONVERSION LOOP
0023 0  1800              RTE     1A
0024 0  180C        LOOP1 RTE     2A
0025 0  E044              AND     MASK   PICK UP LOWER 4 BITS
0026 01 DC000078          STD   L HOLD   STORE A-B REGISTERS
0028 01 C400006A          LD    L VAL1   LOAD PREVIOUS VALUE
002A 0  A047              M       TEN    MULTIPLY BY TEN
002B 0  1800              RTE     1A   CONVERT TO ONE WORD
002C 01 D400006A          STO   L VAL1   STORE IT
002E 01 C4000078          LD    L HOLD   LOAD MASKED PORTION
0030 0  A03A              A       VAL1   ADD THE OLD VAL1
0031 01 D400006A          STO   L VAL1   STORE THE NEW VALUE
0033 01 CC000078          LDD   L HOLD   LOAD THE A-B REGISTERS
0035 0  71FF              MDX     -1     LOOP FOUR TIMES
0036 01 4C000024          BSC   L LOOP1
0038 01 C4000076          LD    L ZERO
003A 01 D400006C          STO   L VAL2
003C 01 C400006A          LD    L VAL1   LOAD BOTTOM 4 CONVERT DIGITS
003E 0  1800              RTE     1A
003F 01 C400006C          LD    L VAL2     ARRANGE THEM IN THE
0041 01 DC00006E          STD   L VAL3     B-REGISTER
0043 01 C4000074          LD    L SAVE   LOAD TOP TWO DIGITS
0045 0  1804              SRA     4      GET DIGIT SIX
0046 0  E023              AND     MASK   PICK UP DIGIT
0047 0  A02A              M       TEN    MULTIPLY BY TEN
0048 0  1800              RTE     1A   GET INTO A-REGISTER
0049 01 D4000078          STO   L HOLD   STORE IN HOLD
004B 01 C4000074          LD    L SAVE     LOAD AGAIN
004D 0  E01C              AND     MASK   PICK OFF DIGIT FIVE
004F 01 A4000078          A     L HOLD   ADD DIGIT SIX
0050 0  A020              M       TENK     MULTIPLY BY 10,000
0051 0  A01C              AD      VAL3   ADD LOWER 4 CONVERTED DIGITS
0052 00 0F00FF8E          STO   L2 -322  STORE IN COMMON
0054 0  72FF              MDX     -2
0055 01 4C000009          BSC   L LOOP2
0057 01 CC000074          LDD   L TEMP   RESTORE
0059 00 65000000    XR1   LDX   L1 *-*
005B 00 66000000    XR2   LDX   L2 *-*
005D 01 4C800000          BSC   I 0T   RETURN
0060    0000              BSS   E 0
0060 1  0076        DP    DC      ZERO
0061 0  617C              DC      /617C
0062    0000              BSS   E 0
0062 1  0064        DN    DC      NN
0063 0  617C              DC      /617C
0064 0  FFFF        NN    DC      /FFFF
0066    0000              BSS   E 0
0066 0  0000        IN1   DC      *-*
0067 0  5F45              DC      /5F45
0068    0000              BSS   E 0
0068 0  0000        IN2   DC      *-*
0069 0  5F44              DC      /5F44
006A 0  000F        MASK  DC      /000F
006B    0001        VAL1  BSS     1
006C    0001        VAL2  BSS     1
006E    0002        VAL3  BSS   E 2
0070 0  FFFF        NEG   DC      /FFFF
0071 0  2710        TENK  DC      10000
0072 0  000A        TEN   DC      10
0074    0002        TEMP  BSS   E 2
0076 0  0000        ZERO  DC      /0000
0078    0002        HOLD  BSS   E 2
007A    0001        SAVE  BSS     1
007C                      END

     NO ERRORS IN ABOVE ASSEMBLY.
DI
DUP FUNCTION COMPLETED
// ASM FCO
  *LIST ALL
```

```
0000    050006000          ENT     ECO
0000 0  0000        ECO    DC      0
0001 01 DC000012           STD   L TEMP   SAVE A-0 REGISTERS
0003 0  6906               STX   1 XR1,1    SAVE XR1
0004 01 74000000           MDX   L ECO,0     SETS UP RETURN FOR ZERO PARMS
0006 0  0807               XIO     ON
0007 01 CC000012           LDD   L TEMP   RESET  A-0 REGISTERS
0009 00 A5000000    XR1    LDX   L1 *-*   RESET XR1
000B 01 4C000000           MSC   I ECO     RETURN
000E    0000               BSS   E 0
000F 1  0010        ON     DC      NN
000F 0  A17C               DC      /A17C
0010 0  FFFF        NN     DC      /FFFF
0012    0002        TEMP   BSS   E 2
0014                       END

      NO ERRORS IN ABOVE ASSEMBLY.
ECO
DUP FUNCTION COMPLETED
// ASM CRUNC
  *LIST ALL
```

```
0000    03664543          ENT     CRUNC
0000 0  0000      CRUNC   DC      0
0001 01 DC0000D4          STD   L TEMP1   SAVE A-Q REGISTERS
0003 01 60000003          STX   L1 XR1&1  SAVE  XR1
0005 01 6E0000C5          STX   L2 XR2&1  SAVE XR2
0007 01 65800000          LDX   I1 CRUNC  LOADS ADDR OF PARM VECTOR
0009 00 C5800000          LD    I1 0
000B 01 D4000110          STD   L MCHAN
000D 00 C5800001          LD    I1 1
000F 01 D4000116          STD   L DIG1
0011 00 C5800002          LD    I1 2
0013 01 D4000117          STD   L DIG2
0015 00 C5800003          LD    I1 3
0017 01 D4000118          STD   L DIG3
0019 00 C5800004          LD    I1 4
001B 01 D4000119          STD   L DIG4
001D 01 74050000          MDX   L CRUNC,5     SETS UP RETURN
001F 01 C400011A          LD    L MCHAN
0021 01 4C300060          ASC   L LOAD,7-
0023 01 C4000008          LD    L MARK
0025 01 65800116          LDX   I1 DIG1    LOAD VALUE OF DIG1 IN XR1
0027 0  1900              SRA   1             SHIFT  THE PUNCH
0028 01 D4000116          STD   L DIG1
002A 01 C4000008          LD    L MARK  THIS IS DONE FOR ALL THE DIGITS
002C 01 65800117          LDX   I1 DIG2
002E 0  1900              SRA   1
002F 01 D4000117          STD   L DIG2
0031 01 C4000008          LD    L MARK
0033 01 65800118          LDX   I1 DIG3
0035 0  1900              SRA   1
0036 01 D4000118          STD   L DIG3
0038 01 C4000008          LD    L MARK
003A 01 65800119          LDX   I1 DIG4
003C 0  1900              SRA   1
003D 01 D4000119          STD   L DIG4
003F 01 C4000007          LD    L ZERO   START PACKING ROUTINE TO
0041 0  1800              RTE     16     PACK-UP
0042 01 C4000117          LD    L DIG2
0044 0  180C              SRA     12
0045 01 84000116          A     L DIG1   THE RUN NO.
0047 01 D400000C          STO   L RUN1    AND
0049 0  1800              RTE     16      STORE IT
004A 01 D4000008          STO   L RUN2   IN RUN 1-3
004C 01 C4000007          LD    L ZERO
004E 0  1800              RTE     16
004F 01 C4000118          LD    L DIG3
0051 0  1808              SRA     8
0052 01 84000008          A     L RUN2
0054 01 D4000008          STO   L RUN2
0056 0  1800              RTE     16
0057 01 D400000A          STO   L RUN3
0059 01 C4000119          LD    L DIG4
```

```
005B 0  1804              SRA    4
005C 01 840000DA          A      L  RUN3
005F 01 040000DA          STO    L  RUN3
0060 01 C4000009    LOAD  LD     L  FOUR
0062 01 0400000I          STO    L  COUNT    PREPARE THE CARD COUNTER
0064 0  02FE              LDX    2 -2   COMMON WORD LOCATION
0065 0  0132        LOOP2 LDX    1 50   ZERO OUT THE
0066 01 C4000007          LD     L  ZERO     CARD ARRAY
0068 01 050000DF    LOOP  STO    L1 PUNCH-50
006A 0  71FF              MDX    1 -1
006B 0  70FC              MDX      LOOP
006C 0  612E              LDX    1 46
006D 01 C400011D          LD     L  MCHAN    CHECK FOR LAST CARD
006F 01 940001IA          S      L  MCHAN
0071 01 040001IB          STO    L  REM
0073 01 94000006          S      L  FORTY
0075 01 4C08000A          BSC    L  HIT,&
0077 01 C400011C          LD     L  ONE
0079 01 84000006          A      L  FORTY    PREPARE FIRST COLUMN ON CARD
007B 01 04000115          STO    L  CARD
007D 00 CE000000    LOOP1 LOD    L2 0   HERE IS WHERE THE DATA IS PACKED
007F 0  108C              SLT      12
0080 01 050000F3          STO    L1 PUNCH-46
0082 0  1800              RTE      16
0083 01 04000003          STO    L  TEMP
0085 00 CE00FFFF          LOD    L2 -2
0087 0  1088              SLT      8
0088 01 84000003          A      L  TEMP
008A 01 050000E2          STO    L1 PUNCH-47
008C 0  1800              RTE      16
008D 01 04000003          STO    L  TEMP
008F 00 CE00FFFC          LOD    L2 -4
0091 0  1084              SLT      4
0092 01 84000003          A      L  TEMP
0094 01 050000E1          STO    L1 PUNCH-48
0096 0  1800              RTE      16
0097 01 04000003          STO    L  TEMP
0099 00 CE00FFFA          LOD    L2 -6
009B 01 84000003          A      L  TEMP
009D 01 050000E0          STO    L1 PUNCH-49
009F 0  1800              RTE      16
00A0 01 050000DF          STO    L1 PUNCH-50
00A2 01 C400011A          LD     L  MCHAN
00A4 01 84000009          A      L  FOUR
00A6 01 0400011A          STO    L  MCHAN
00A8 01 9400011D          S      L  NCHAN    CHECK FOR END OF DATA
00AA 01 4C100080          BSC    L  STOP,-
00AC 0  72F8              MDX    2 -8
00AD 0  1000              NOP
00AE 0  71F8              MDX    1 -5
00AF 0  70D0              MDX      LOOP1
00B0 30 022559A3    STOP  CALL     RTNWT
00B2 1  0115              DC       CARD
00B3 01 C4000112          LD     L  FCHAN
00B5 01 84000006          A      L  FORTY    CHANNEL COUNTER
00B7 01 04000112          STO    L  FCHAN
00B9 01 C400011D          LD     L  NCHAN
00BB 01 9400011A          S      L  MCHAN
00BD 01 4C08000C          BSC    L  FINAL,&  LAST CARD JUMP
00BF 01 74FF0001          MDX    L  COUNT,-1  CARD COUNTER
00C1 0  70A3              MDX      LOOP2
00C2    0000        FINAL BSS      0
00C2 00 65000000    XR1   LDX    L1 *-*
00C4 00 66000000    XR2   LDX    L2 *-*
00C6 01 CC000004          LDD    L  TEMP1
00C8 01 4C800000          BSC    I  GRUNC    RETURN
00CA 01 C4000002    HIT   LD     L  LAST     SETS UP LAST CARD PUNCH
00CC 0  804F              A        REM
00CD 01 04000115          STO    L  CARD
00CF 01 4C000070          BSC    L  LOOP1
0001 0  0000        COUNT DC       /0000
0002 0  1800        LAST  DC       /1800
0003 0  0000        TEMP  DC       /0000
0004    0002        TEMP1 BSS    E 2
0006 0  0028        FORTY DC       40
0007 0  0000        ZERO  DC       /0000
0008 0  2000        MARK  DC       /2000
0009 0  0004        FOUR  DC       4
                    *
000A    0001        RUN3  BSS      1
```

```
000B     0001          RUN2  BSS     1
000C     0001          RUN1  BSS     1
000D 0   0000                DC      /0000
000E 0   2041                DC      /2041
000F 0   4012                DC      /4012
0010     0031                BSS     49
0111     0001          PUNCH BSS     1
0112 0   0000          PCHAN DC      /0000
0113 0   0000                DC      /0000
0114 0   0000                DC      /0000
0115 0   0000          CARD  DC      /0000
                       *
0116     0001          DIG1  BSS     1
0117     0001          DIG2  BSS     1
0118     0001          DIG3  BSS     1
0119     0001          DIG4  BSS     1
011A 0   0000          MCHAN DC      /0000
011B     0001          REM   BSS     1
011C 0   3H00          ONE   DC      /3H00
011D     0001          MCHAN BSS     1
011E                   END

     NO ERRORS IN ABOVE ASSEMBLY.
CRUNC
DUP FUNCTION COMPLETED
*STORECT              ND    ND
*CCEND
```

```
CLR, BUILD ND

CLR, ND    LD XQ

DUP FUNCTION COMPLETED
NONPROCESS DISK NO.1
```

APPENDIX B

A LISTING OF PROGRAM SPECTRA

```
      1                 OVERLAY(SPECTRA,0,0)                                          SP        2
                        PROGRAM SPECTRA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)        SP        3
                  C                                                                   SP        4
                  C     DECLARATIONS, DIMENSIONS, AND FORMAT STATEMENTS               SP        5
      5           C                                                                   SP        6
                        DIMENSION SPECTR(4100),MDATA(4100),LDATA(4100),LABEL(16),X(7),AP(7  SP        7
                     1),CENTRD(200),WCM(200),WCM(200),LCM(200),AAREA(200),SIGMA(200),UNE  SP        8
                     2PGY(200),WSTART(200),SCNTRD(50),ENERGY(50),AMENRG(50),AMCNTR(50),MU  SP        9
                     1F(256),TAGWRD(16)                                               SP       10
     10              DIMENSION ELEMNT(50),AMKEV(50),EMWT(50),EMDIM(50),CFARAY(50,11),M  SP       11
                     1ELE(50),MELETW(50),LAMBDA(50),SCFARY(50,11),EMWGT(1?)           SP       12
                        DIMENSION SAMPNM(100,9),SAMWT(100),AMCT(100),AINCT(100),TIMECT(1  SP       13
                     10),SUBTIV(100),DLIVTV(100),SMPDIM(100),NSKIP(100)               SP       14
                        DIMENSION ELNAM(200),EGAM(200),SCF(200),DATA(200),SIGSCF(200),UNIT  SP       15
     15              1S(200),UNITS1(200),STGDAT(200)                                  SP       16
                     EQUIVALENCE (APE?RD(1),FWA),(APCNTRD(1),FWAMCNTR(1)),(SPECTR(1),MD  SP       17
                     1ATA(1)),(AP(1),X(1))                                            SP       18
                        COMMON MDATA,AP                                               SP       19
                        INTEGER SPECTR,SIZE,EMAX,TAGWRD                               SP       20
     20                 INTEGER TIMECT,AMCT,AINCT,START                               SP       21
                        REAL MDATA,LDATA                                              SP       22
                        REAL INTEG,M                                                  SP       23
                        DOUBLE PRECISION DCONST,LAMBDA,HELF,ATIME,CTIME               SP       24
                        DOUBLE PRECISION TEMP                                         SP       25
     25                 LOGICAL ELNAM                                                 SP       26
                        DATA BLK,MD,Y,BLANK/1HS,1HM,1HP,1HD,1HY,1H /                  SP       27
                      2 FORMAT(12I6)                                                  SP       28
                      3 FORMAT(1H1,T2,*12F=*,I4,* CHANNELS*)                          SP       29
                      4 FORMAT(5I6,15A4,A2)                                           SP       30
     30               7 FORMAT(5X,15,10I9)                                           SP       31
                      9 FORMAT(1H1////5X,15,10I9)                                     SP       32
                     10 FORMAT(1H1////5X,*SPECTRUM NUMBER*,I4,10X,*TAGWORD=*,I4/)     SP       33
                     12 FORMAT(//10X,*SPECTRUM NUMBER*,I4,10X,*TAGWORD=*,I4)          SP       34
                     13 FORMAT(6I12,I6)                                               SP       35
     35              14 FORMAT(//T20,15A4,A2)                                         SP       36
                     81 FORMAT(1HF9.3)                                                SP       37
                     82 FORMAT(//T20,* FCS CENTROID=* ,F9.3,1?X,* FGAIN=* ,F9.7,2X,   SP       38
                        1 *KEV/CHANNEL* )                                             SP       39
                     83 FORMAT(//T20,*CENTROID OF STANDARD*,5X,*ENERGY OF STANDARD*)  SP       40
     40              84 FORMAT (T22,*R=*,1PE9.3)                                      SP       41
                    109 FORMAT(1H1/T20,*AREAS AND CENTROIDS OF PEAKS  -- PEAK AREA EVALUAT  SP       42
                        1ED BY GAUSSIAN METHOD USING*,1X,I6,1X,*CHANNELS*)            SP       43
                    110 FORMAT(//T10,*#CENTROID*,6X,*#CHANNELS*,6X,*#PEAK*,9X,*#AREA*,  SP       44
                        1 11X,*#SIGMA*)                                               SP       45
     45             111 FORMAT(1HX,F10.3,6X,I4,4X,I4,4X,I4,5Y,F12.1,4X,F11.2)        SP       46
                    112 FORMAT(1H1/T20,*AREAS AND CENTROIDS OF PEAKS -- PEAK AREA DETERMIN  SP       47
                        1ED AS TOTAL AREA BETWEEN LISTED PEAK LIMITS*)                SP       48
                    114 FORMAT(1H1/T10,*#CENTROID*,6X,*#CHANNELS*,6X,*#PEAK*,9X,*#AREA*,  SP       49
                        111X,5HSIGMA)                                                 SP       50
     50             330 FORMAT (/1CX,*#CENTROID*,6X,*#CHANNELS*,6X,*#PEAK*,14X,*#ENERGY*,8X,  SP       51
                        1*AREA*,11X,*SIGMA*)                                          SP       52
                    331 FORMAT(1HX,F10.3,4X,I4,4X,I4,4X,I4,10X,F10.3,4X,F12.1,4X,F11.2)  SP       53
                    332 FORMAT (1H1/1CX,*CENTROID*,6X,*CHANNELS*,6X,*PEAK*,14X,*ENERGY*,  SP       54
     55                 1HX,*AREA*,11X,*SIGMA*)                                       SP       55
                    350 FORMAT(1H1///10X,*SMOOTHED DATA POINTS*,10X,*SPECTRUM NUMBER*,I4,1  SP       56
                        10X,*TAGWORD=*,I4)                                            SP       57
                    360 FORMAT(5X,I10,1PE9.2)                                         SP       58
                    370 FORMAT(1H1////5X,I10,1PE9.2)                                  SP       59
                    414 FORMAT(/2X,*CENTROID*,6X,*CHANNELS*,7X,*ENERGY*,10X,*AREA*,4X,*ELE  SP       60
     60                 1MENT*,4X,*PEAK*,11X,*SCF*,7X,*SIGMA*)                        SP       61
                    418 FORMAT(1X,F9.3,4X,I4,2X,I4,4X,F8.3,4Y,F10.1,4X,A4,5X,F8.3,4X,1PF10  SP       62
                        1.4,4Y,1PE8.2)                                                SP       63
                    419 FORMAT(1X,F8.3,4X,I4,2X,I4,4X,F8.3,4Y,F10.1)                  SP       64
                    420 FORMAT(1H1////5X,*SPECTRUM NUMBER*,I4,*CHANNELS*,7X,*ENERGY*,10X,*AREA*,4X  SP       65
     65                 1,*ELEMENT*,4X,*PEAK*,11X,*SCF*,7X,*SIGMA*)                   SP       66
                    423 FORMAT(/2X,*CENTROID*,6X,*CHANNELS*,7X,*ENERGY*,10X,*AREA*,4X,*ELE  SP       67
                        1MENT*,4X,*PEAK*,11X,*SCF*,7X,*CONC.*,17X,*SIGMA*)            SP       68
                    427 FORMAT(1X,F9.3,4X,I4,2X,I4,4X,F8.3,4Y,F10.1,4X,A4,5X,F8.3,4X,1PF10  SP       69
     70                 1.4,4X,1PE10.4,1X,A4,///,A4,4X,1PE9.2)                        SP       70
                    429 FORMAT(1H1///2X,*CENTROID*,6X,*CHANNELS*,7X,*ENERGY*,10X,*AREA*,4X  SP       71
                        1,*ELEMENT*,4X,*PEAK*,11X,*ACF*,9X,*CONC.*,17X,*SIGMA*)       SP       72
                    434 FORMAT(1H1///T20,*STANDARD COMPARISON FACTORS -- LAST COLUMN LISTS  SP       73
                        1 AVERAGE SCF///)                                             SP       74
                    439 FORMAT(1X,*ELEMENT*,5X,*PE*,///)                              SP       75
     75             441 FORMAT(1X,A4,2X,F8.3,11(2X,1PE9.2))                          SP       76
                    455 FORMAT (//T20,* CENTROID=*, F9.3)                             SP       77
                    470 FORMAT(A2,8X,F10.3,3X,F7.3,4Y,A1,5Y,F10.5,6X,A4)              SP       78
```

```
                471 FORMAT(4A4,4X,F10.5,4X,A4,5X,I3,I2,3X,I6,A1,4X,I6)                  SP        79
                481 FORMAT(5X,#SAMPLE: #,4A4,/)                                        SP        80
     80         485 FORMAT(I6,A4)                                                      SP        81
                494 FORMAT(/10X,#SAMPLE: #,4A4)                                        SP        82
                514 FORMAT(//2X,#CENTROID#,5X,#CHANNELS#,7X,#ENERGY#,10X,#AREA#,4X,#ELE SP        83
                   1MENT#,6X,#PEAK#,11X,#ACF#,6X,#SIGMA#)                              SP        84
                520 FORMAT(1H1///2X,#CENTROID#,5X,#CHANNELS#,7X,#ENERGY#,10X,#AREA#,4X SP        85
                   1,#ELEMENT#,6X,#PEAK#,11X,#ACF#,6X,#SIGMA#)                         SP        86
                562 FORMAT(///T20,#DATA FOR ACTIVATION ANALYSIS RUN#)                  SP        87
                563 FORMAT(1H1//T20,#CENTROIDS OF GAMMA-RAY STANDARDS#)                SP        88
                582 FORMAT(//T20,#ELEMENT#,7X,#PHOTOPEAK ENERGY#,11X,#HALF LIFE#,6X,#FL SP        89
                   1UX MONITOR WEIGHT#/)                                               SP        90
     90         584 FORMAT(T20,A2,18X,F10.5,10X,F8.3,1X,A1,10X,F10.5,1X,A4)            SP        91
                599 FORMAT(/T20,#WEIGHT=#,F9.5,1X,A4,10X,#COUNT STARTED#,1X,I3,I2,9X,# SP        92
                   1LIVE TIME DURATION=#,I4,A1)                                        SP        93
                630 FORMAT(#1#,T20,#SPECTRAL DATA SMOOTHED#)                           SP        94
                C                                                                      SP        95
     95         C     INITIALIZE VARIABLES AND ARRAYS                                  SP        96
                C                                                                      SP        97
                   10 DO 66 I=1, 4100                                                  SP        98
                      SPECTR(I)=0                                                      SP        99
                      MDATA(I)=0.0                                                     SP       100
    100            66 LDATA(I)=0.0                                                     SP       101
                      NRDERR=0                                                         SP       102
                      NSTSPC=0                                                         SP       103
                      NCHECK=0                                                         SP       104
                      NPASS=0                                                          SP       105
    105               NSMPRO=0                                                         SP       106
                      NRUN=200                                                         SP       107
                      ARAT=1.0                                                         SP       108
                C                                                                      SP       109
                C     READ STATEMENTS                                                  SP       110
    110         C                                                                      SP       111
                      READ(5,2)NSTSPC,EMAX,ISSPOT,NYSPOT                               SP       112
                      CSSPOT=ISSPOT                                                    SP       113
                      YSPOT=NYSPOT                                                     SP       114
    115               IF (NSTSPC .EQ. 0) GO TO 71                                      SP       115
                      IF(CSSPOT.EQ.0.0) GO TO 70                                       SP       116
                      RAT=661.595/CSSPOT                                               SP       117
                      ARAT=1.0                                                         SP       118
                      IF(YSPOT.NE.0.0)ARAT=1836.08/YSPOT                               SP       119
                585 MSTSPC=NSTSPC                                                      SP       120
    120               GO TO 72                                                         SP       121
                   70 MSTSPC=NSTSPC+1                                                  SP       122
                      IF(EMAX.LE.1836) GO TO 72                                        SP       123
                      MSTSPC=MSTSPC+1                                                  SP       124
                   72 READ (5,2) (NSTAND(I), I=1, MSTSPC)                              SP       125
    125               NPEAKS=0                                                         SP       126
                      DO 73 I=1, MSTSPC                                                SP       127
                   73 NPEAKS=NPEAKS+NSTAND(I)                                          SP       128
                      READ (5,81) (ENERGY(I), I=1, NPEAKS)                             SP       129
                   71 READ(5,4)      SKP,SIZE,(NSPEC,(LABEL(I), I=1,16)               SP       130
    130               IF (EOF(5LINPUT).NE.0) STOP                                      SP       131
                      IF (SKP .LT. 0) GO TO 15                                         SP       132
                      IFLAG=0                                                          SP       133
                      IF(NPEAK.GT.0) GO TO 631                                         SP       134
                      INSKP=SKP                                                        SP       135
    135               NUMSPC=NSTSPC+INSKP                                              SP       136
                631 NSPEC=NSPEC                                                        SP       137
                      READ(5,13)NPRINT,MPRINT,IWT,JANAL,NDEL                           SP       138
                      NDTPNT=NDEL*2+1                                                  SP       139
                475 WRITE (6,3) SIZE                                                   SP       140
    140               WRITE (6,14) (LABEL(I), I=1,16)                                  SP       141
                      IF(NSPEC.EQ.0) GO TO 561                                         SP       142
                      IF(JANAL.GT.0) GO TO 561                                         SP       143
                      IFM=0                                                            SP       144
                      IWIND=0                                                          SP       145
    145               DO 484 I=1,50                                                    SP       146
                      DO 484 J=1,11                                                    SP       147
                484 CFARAY(I,J)=0.0                                                    SP       148
                      READ(5,485)NFM,NPKAN,IFILT,FLUX                                  SP       149
                      IF(NFM.LE.0) IWIND=1                                             SP       150
    150               DO 490 I=1,NPKAN                                                 SP       151
                490 READ(5,470)ELEMNT(I),GAMKEV(I),HFLF(I),HFLFTU(I),FMWT(I),FMDIM(I)  SP       152
                      WRITE(6,562)                                                     SP       153
                      WRITE(6,582)                                                     SP       154
                      DO 583 I=1,NPKAN                                                 SP       155
```

```
                  583 WRITE(6,584)ELEMNT(I),WAVKEV(I),HFLF(I),HFLFTU(I),FWHM(I),FMDIM(I)
                      DO 480 I=1,NCHAN
                      IF(HFLFTU(I).EQ.S) HFLF(I)=HFLF(I)/60.0
                      IF(HFLFTU(I).EQ.M) HFLF(I)=HFLF(I)*60.0
                      IF(HFLFTU(I).EQ.D) HFLF(I)=HFLF(I)*60.0*24.0
                      IF(HFLFTU(I).EQ.Y) HFLF(I)=HFLF(I)*60.0*24.0*365.0
                  480 LAMBDA(I)=0.693/HFLF(I)
                      DO 471 I=1,NSPEC
                  481 READ(5,471)(SAMPWT(I,J),J=1,4),SAMPWT(I),SIPDIM(I),HRCT(I),MINCT(I
                     1),WORLIV(I),DLIVTU(I),NSKIP(I)
                      ISKPPY=1
                      DO 474 I=1,NSPEC
                      ISKPPY=ISKPPY*NSKIP(I)
                  474 TIMECT(I)=60*HRCT(I)+MINCT(I)
                      NSPEC=ISKPPY*NSPEC
                      START=TIMECT(1)
                      IF(NSPEC.EQ.1) GO TO 561
                      DO 482 I=2,NSPEC
                  482 START=MIN0(START,TIMECT(I))
                C
                C     SECTION TO READ DATA
                C
                  561 NBLOCK=SIZE/256
                      NUMSPC=0
                  537 IFLAG=1
                  100 NPASS=NPASS+1
                      IPASS=NPASS-INSKP
                      DO 67 I=1,NRUN
                   67 UNERGY(I)=0.0
                      IF(JANAL.EQ.0.AND.NPASS.GT.NUMSPC) GO TO 502
                  476 CALL CREAD(SPECTR,NCHAN,NRUN,NERR)
                      IF(NCHAN.EQ.-SIZE.AND.NERR.EQ.0) GO TO 537
                C     NOTE--IF NERR .NE. 0 NERR = CARD NUMBER OF ERROR
                      IF(NCHAN .NE. -SIZE .AND. NERR .EQ. 0) WRITE (6,801)NRUN,NCHAN,SIZE
                  801 FORMAT (/2HNR#I5# CONTAINS#I5# CHANNELS#I5# WERE EXPECTED#)
                      GO TO (809,802,804,806), NERR+1
                  802 WRITE (6,803) NCHAN
                  803 FORMAT (# CARD#I4# ILLEGAL#)
                      GO TO 809
                  804 WRITE (6,805) NCHAN
                  805 FORMAT (# CHECKSUM ERROR--CARD#I4)
                      GO TO 809
                  806 WRITE (6,808)
                  808 FORMAT (# LAST CARD(S) MISSING#)
                      GO TO 809
                  809 WRITE (6,810)
                  810 FORMAT (# CREAD ERROR--ITS ALL OVER#/# STOP#)
                      STOP
                  807 TAGWRD(I)=NRUN
                      GO TO 150
                C
                C     SECTION TO CONTROL INPUT OF SPECTRAL DATA IN ACTIVATION ANALYSIS
                C
                  502 NUMSPC=NUMSPC+1
                      IF(NUMSPC.GT.NSPEC) GO TO 503
                      FLXMON=SAMPWT(NUMSPC,1).EQ.FLUX
                  503 IF(NUMSPC.EQ.1) GO TO 402
                      PRESPC=NUMSPC-1
                      IF(NSKIP(PRESPC).GT.0) GO TO 510
                  402 IF(NFM.LT.1) GO TO 574
                      IF(NFM.LE.IFM) GO TO 430
                      IF(FLXMON) GO TO 476
                      GO TO 632
                  430 IF(IWIND.EQ.0) GO TO 431
                      IF(.NOT.FLXMON) GO TO 476
                      GO TO 632
                  431 IRACK=NUMRUN*NBLOCK
                      NPASS=NPASS-(NUMRUN-1)
                      NUMRUN=0
                      NUMSPC=1
                      IWIND=1
                      DO 437 I=1,NPEAK
                      CFARAY(I,11)=0.0
                      SCFARY(I,11)=0.0
                      NRCF=0
                      DO 436 J=1,NFM
                      IF(CFARAY(I,J).EQ.0.0) GO TO 436
```

```
                    CFARAY(I,11)=CFARAY(I,11)+CFARAY(I,J)                                  SP      233
                    SCFARY(I,11)=SCFARY(I,11)+(SCFARY(I,J)**2)                             SP      234
                    NRCF=NRCF+1                                                            SP      235
  235           438 CONTINUE                                                               SP      236
                    IF(NRCF.EQ.0) GO TO 437                                                SP      237
                    CFARAY(I,11)=CFARAY(I,11)/NRCF                                         SP      238
                    SCFARY(I,11)=SQRT(SCFARY(I,11)/(NRCF**2))                              SP      239
                437 CONTINUE                                                               SP      240
  240           442 WRITE(6,438)                                                           SP      241
                    WRITE(6,439)                                                           SP      242
                    NFM1=NFM+1                                                             SP      243
                    DO 440 I=1,NPKAN                                                       SP      244
                    CFARAY(I,NFM1)=CFARAY(I,11)                                            SP      245
  245           440 WRITE(6,441)ELEMNT(I),GAMKEV(I),(CFARAY(I,J),J=1,NFM1)               SP      246
                    GO TO 100                                                              SP      247
                910 IF(NFM.EQ.IFM.AND.IWINO.EQ.0) GO TO 431                                SP      248
                    SKP=NSKIP(PRESPC)                                                      SP      249
                    GO TO 632                                                              SP      250
  250       C                                                                            SP      251
            C       SECTION TO PRINT OUT RAW SPECTRAL DATA                                 SP      252
            C                                                                            SP      253
              150 IF(NPRINT.GT.0.OR.IPASS.LE.NSTSPC) GO TO 152                           SP      254
                    WRITE (6,10) NPASS,TAGWRD(1)                                           SP      255
  255               IF(JANS.EQ.0)WRITE(6,441)(SAMPNM(NUMSPC,J),J=1,4)                    SP      256
                    LNCNT=0                                                                SP      257
                    DO 65 J=1,SIZE,10                                                      SP      258
                    LNCNT=LNCNT+1                                                          SP      259
                    NC=J-1                                                                 SP      260
  260               K=J+9                                                                SP      261
                    IF(LNCNT.GT.51) GO TO 63                                               SP      262
                    WRITE(6,7)NC,(SPECTR(I),I=J,K)                                         SP      263
                    GO TO 64                                                               SP      264
               63   WRITE(6,9)NC,(SPECTR(I),I=J,K)                                       SP      265
  265               LNCNT=0                                                              SP      266
               65   CONTINUE                                                             SP      267
            C                                                                            SP      268
            C       SECTION TO LOCATE PEAKS                                              SP      269
            C                                                                            SP      270
  270       152 TIMECH=SPECTR(1)                                                         SP      271
                    DO 147 I=1,4100                                                        SP      272
              147   MDATA(I)=SPECTR(I)                                                   SP      273
                    IF(NPRINT.EQ.1) GO TO 148                                              SP      274
                    CALL  SMOOTH(SIZE,M1)                                                  SP      275
  275               GO TO 151                                                            SP      276
              148   M1=SIZE                                                              SP      277
              151   M2=SIZE-M1+3                                                         SP      278
                    L=M1-2                                                               SP      279
                    N=1                                                                    SP      280
  280               I=M2-2                                                               SP      281
                    J=I+3                                                                  SP      282
                    DO 153 K=I,J                                                           SP      283
                    N=N+1                                                                  SP      284
              153   AP(N)=MDATA(K)                                                       SP      285
  285               DO 154 I=M2,L                                                        SP      286
                    J=I+2                                                                  SP      287
                    DO 155 K=1,4                                                           SP      288
                    KA=K+1                                                               SP      289
              155   AP(K)=AP(KA)                                                         SP      290
  290               AP(5)=MDATA(J)                                                       SP      291
                    A21=+AP(4)-AP(2)+2.0*(AP(5)-AP(1))                                     SP      292
                    LDATA(I)=A21/10.0                                                    SP      293
              154   CONTINUE                                                             SP      294
                    DO 149 I=1,200                                                         SP      295
  295               CENTRD(I)=0.0                                                        SP      296
                    NRUN=0                                                               SP      297
                    MP=SIZE-L+1                                                          SP      298
                    MR=L-2                                                               SP      299
                    DO 200 I=MP,MR                                                         SP      300
  300               IF(LDATA(I).LT.0.0) GO TO 200                                        SP      301
                    I1 = I+1                                                             SP      302
                    IF(LDATA(I1).GT.0.0) GO TO 200                                         SP      303
                156 I1=I1-2                                                              SP      304
                157 IF(LDATA(I1).LE.0.0) GO TO 200                                       SP      305
  305               J = I1-1                                                             SP      306
                159 IF(LDATA(J)) 162,162,160                                             SP      307
                160 J = J-1                                                              SP      308
                    GO TO 159
```

```
                  162 KL=J                                                  SP        310
      310             KL2=J                                                 SP        311
                      ALMIN=MDATA(KL)                                       SP        312
                      KL1=KL+1                                              SP        313
                      KL3=KL+3                                              SP        314
                      DO 163 J=KL1,KL3                                      SP        315
      315             IF(MDATA(J).GT.ALMIN) GO TO 163                       SP        316
                      ALMIN=MDATA(J)                                        SP        317
                      KL=J                                                  SP        318
                  163 CONTINUE                                              SP        319
                      J=I+2                                                 SP        320
      320             IF(LDATA(J)) 166,166,164                              SP        321
                  164 J=J+1                                                 SP        322
                      GO TO 165                                             SP        323
                  168 N1=J                                                  SP        324
                      J=I+2                                                 SP        325
      325         171 IF(LDATA(KL0).LE.LDATA(J)) GO TO 174                  SP        326
                      J=J+1                                                 SP        327
                      GO TO 171                                             SP        328
                  174 N2=J                                                  SP        329
                      IF(N2.LE.N1) GO TO 178                                SP        330
      330             KR=N1                                                 SP        331
                      GO TO 179                                             SP        332
                  178 KR=N2                                                 SP        333
                  179 ARMIN=MDATA(KR)                                       SP        334
                      KR1=KR-1                                              SP        335
      335             KR3=KR-3                                              SP        336
                      DO 180 J=KR3,KR1                                      SP        337
                      IF(MDATA(J).GT.ARMIN) GO TO 180                       SP        338
                      ARMIN=MDATA(J)                                        SP        339
                      KR=J                                                  SP        340
      340         180 CONTINUE                                              SP        341
                  210 K=0                                                   SP        342
                      KR1=KR-1                                              SP        343
                      AL=KL                                                 SP        344
                      AR=KR                                                 SP        345
      345             IF(KR.LE.KL) GO TO 200                                SP        346
                      DYDX=(MDATA(KL)-MDATA(KR))/(AL-AR)                    SP        347
                      YINT=MDATA(KL)-(AL*DYDX)                              SP        348
                      DO 211 J=1,KR1                                        SP        349
                      A=J                                                   SP        350
      350             CHECK=(DYDX*A)+YINT                                   SP        351
                      IF(CHECK.LT.MDATA(J)) GO TO 211                       SP        352
                      K=K+1                                                 SP        353
                  211 CONTINUE                                              SP        354
                      IF(K.LE.0) GO TO 212                                  SP        355
      355             KR=KR-1                                               SP        356
                      GO TO 210                                             SP        357
                  212 K=0                                                   SP        358
                      KL1=KL+1                                              SP        359
                      AL=KL                                                 SP        360
      360             AR=KR                                                 SP        361
                      IF(KR.LE.KL) GO TO 200                                SP        362
                      DYDX=(MDATA(KL)-MDATA(KR))/(AL-AR)                    SP        363
                      YINT=MDATA(KL)-(AL*DYDX)                              SP        364
                      DO 213 J=KL1,I                                        SP        365
      365             A=J                                                   SP        366
                      CHECK=(DYDX*A)+YINT                                   SP        367
                      IF(CHECK.LT.MDATA(J)) GO TO 213                       SP        368
                      K=K+1                                                 SP        369
                  213 CONTINUE                                              SP        370
      370             IF(K.LE.0) GO TO 214                                  SP        371
                      KL=KL+1                                               SP        372
                      GO TO 212                                             SP        373
                  214 CONTINUE                                              SP        374
                      AL=KL                                                 SP        375
      375             AR=KR                                                 SP        376
                  C                                                         SP        377
                  C     SECTION TO DETERMINE BASELINE                       SP        378
                  C                                                         SP        379
                      IF(MDATA(KL).LT.0.0) MDATA(KL)=0.0                    SP        380
      380             IF(MDATA(KR).LT.0.0) MDATA(KR)=0.0                    SP        381
                      YMIN=2.0*(SQRT(MDATA(KL)))                            SP        382
                      ZMIN=2.0*(SQRT(MDATA(KR)))                            SP        383
                      AVBKGL=MDATA(KL)                                      SP        384
                      AVBKGR=MDATA(KR)                                      SP        385
      385             NL=1                                                  SP        386
```

```
                      NR=1                                                      SP       387
                      DO 188 J=1,4                                              SP       388
                      KRAV=KR+J                                                 SP       389
                      KLAV=KL-J                                                 SP       390
    390               AKR=ABS(MDATA(KR)-MDATA(KRAV))                            SP       391
                      AKL=ABS(MDATA(KL)-MDATA(KLAV))                            SP       392
                      IF(AKL.GT.YMIN) GO TO 187                                 SP       393
                      AVBKGL=AVBKGL+MDATA(KLAV)                                 SP       394
                      NL=NL+1                                                   SP       395
    395           187 IF(AKR.GT.ZMIN) GO TO 188                                 SP       396
                      AVBKGR=AVBKGR+MDATA(KRAV)                                 SP       397
                      NR=NR+1                                                   SP       398
                  188 CONTINUE                                                  SP       399
                      AVBKGL=AVBKGL/NL                                          SP       400
    400               AVBKGR=AVBKGR/NR                                          SP       401
                      DYDX=(AVBKGL-AVBKGR)/(AL-AR)                              SP       402
                      YINT=AVBKGL-(AL*DYDX)                                     SP       403
                C                                                               SP       404
                C     SECTION TO APPLY STATISTICAL TEST                         SP       405
                C                                                               SP       406
    405               PEAK=0                                                    SP       407
                      DO 182 J=KL,KR                                            SP       408
                  182 PEAK=PEAK+MDATA(J)                                        SP       409
                      BKG=((AVBKGL+AVBKGR)/2.0)*(AR-AL+1.0)                     SP       410
    410               AREA=PEAK-BKG                                             SP       411
                      VAR=PEAK+(AVBKGL+AVBKGR)*((((AR-AL-1.0)/2.0)**2)-1)       SP       412
                      IF(VAR.LE.0.0) VAR=1.0                                    SP       413
                      STAT=AREA/(SQRT(VAR))                                     SP       414
                      IF(STAT.LT.2.0) GO TO 200                                 SP       415
    415           C                                                             SP       416
                C     SECTION TO FILTER OUT COMPTON EDGES AND TEST FIRST DERIVATIVE SP       417
                C                                                               SP       418
                      K=KL+1                                                    SP       419
                      AKL=LDATA(KL)                                             SP       420
    420               DO 215 J=K,I                                              SP       421
                      AK=AMAX1(AKL,LDATA(J))                                    SP       422
                  215 AKL=AK                                                    SP       423
                      K=I+1                                                     SP       424
                      AKR=LDATA(I)                                              SP       425
    425               DO 216 J=K,KR                                             SP       426
                      AK=AMIN1(AKR,LDATA(J))                                    SP       427
                  216 AKR=AK                                                    SP       428
                      BKR=ABS(AKR)                                              SP       429
                      A=1                                                       SP       430
    430               A=DYDX*A+YINT                                             SP       431
                      IF(A.LT.1.0) A=1.0                                        SP       432
                      A=SQRT(A)                                                 SP       433
                      DRVTST=(AKR-AKL)/A                                        SP       434
                      SLPTST=BKR/AKL                                            SP       435
    435               IF(DRVTST.LT.2.0) GO TO 200                               SP       436
                      IF(SLPTST.GT.2.0.OR.SLPTST.LT.0.20) GO TO 200             SP       437
                C                                                               SP       438
                C     SECTION TO DETERMINE CENTROID AND TEST SECOND DERIVATIVE  SP       439
                C                                                               SP       440
    440               J=I-3                                                     SP       441
                      A=0.0                                                     SP       442
                      LP=0                                                      SP       443
                      DO 195 N=J,K                                              SP       444
                      LP=LP+1                                                   SP       445
    445               X(LP)=MDATA(N)-((DYDX*A)+YINT)                            SP       446
                  195 A21 = (3.0*(X(7)-X(1))) + (2.0*(X(6)-X(2))) + (X(5)-X(3)) SP       447
                      A21 = A21/26.0                                            SP       448
                      A22 = (4.0*(X(7)+X(1))) - (3.0*(X(6)+X(5))) - (4.0*X(4))  SP       449
                      A22=A22/42.0                                              SP       450
    450               IF(A22.GT.0.0) GO TO 200                                  SP       451
                      CENT=-A21/A22                                             SP       452
                      A=1                                                       SP       453
                      CENT=CENT+A                                               SP       454
    455               MID=CENT+0.5                                              SP       455
                C                                                               SP       456
                C     SECTION TO DETERMINE PEAK AREA                            SP       457
                C                                                               SP       458
                      IF(INT.GT.0) GO TO 240                                    SP       459
    460               TOP=MDATA(MID)                                            SP       460
                      NHAFL=MID-MDEL                                            SP       461
                                                                                SP       462
                                                                                SP       463
```

```
                NHAFR=MID+NDEL                                          SP      464
                IF(NHAFL.LT.KL) GO TO 183                               SP      465
465             IF(NHAFR.GT.KR) GO TO 183                               SP      466
                ADEL=NDEL                                               SP      467
                APEAK=0                                                 SP      468
                DO 184 I4=NHAFL,NHAFR                                   SP      469
184             APEAK=APEAK+MDATA(I4)                                   SP      470
470             AHAFL=NHAFL                                             SP      471
                AHAFR=NHAFR                                             SP      472
                ALPKG=DYDX*AHAFL+YINT                                   SP      473
                ARKG=DYDX*AHAFR+YINT                                    SP      474
                ARKG=(ALPKG+ARKG)*(ADEL+0.5)                            SP      475
475             AREA=APEAK-ARKG                                         SP      476
                VAR=APEAK+(ADEL+0.5)*ARKG                               SP      477
                IF(VAR.LT.0.0) VAR=0.0                                  SP      478
                GO TO 797                                               SP      479
183             AREA=0.0                                                SP      480
480             VAR=0.0                                                 SP      481
C                                                                       SP      482
C       SECTION TO DEFINE PEAK ATTRIBUTES                               SP      483
C                                                                       SP      484
790             NRUN=NRUN+1                                             SP      485
485             LCH(NRUN)=KL                                            SP      486
                MCH(NRUN)=KR                                            SP      487
                MCH(NRUN)=MID                                           SP      488
                CENTRD(NRUN) = CENT                                     SP      489
                AAREA(NRUN)=AREA                                        SP      490
490             SIGMA(NRUN)=SQRT(VAR)                                   SP      491
205             CONTINUE                                                SP      492
C                                                                       SP      493
C       SECTION TO PASS CONTROL TO OTHER PARTS OF THE PROGRAM           SP      494
C                                                                       SP      495
495             IF(CSSPOT.EQ.0.0) GO TO 560                             SP      496
                IF (NSTSPC .EQ. 0) GO TO 91                             SP      497
                IF(IPASS.EQ.1) GO TO 586                                SP      498
                IF(IPASS-NSTSPC)456,456,311                             SP      499
560             IF(NSTSPC.EQ.0) GO TO 91                                SP      500
500             IF (IPASS .EQ. 1) GO TO 301                             SP      501
                IF (EMAX .GT. 1836) GO TO 401                           SP      502
                IF (IPASS .EQ. 2) GO TO 302                             SP      503
                IF(IPASS-NSTSPC)456,456,311                             SP      504
401             IF (IPASS .EQ. 2) GO TO 451                             SP      505
505             IF (IPASS .EQ. 3) GO TO 305                             SP      506
                IF(IPASS-NSTSPC)456,456,311                             SP      507
C                                                                       SP      508
C       SECTION TO FIND CS PEAK IN SPECTRUM 1 AND APPROXIMATE KEV/CHANNEL SP      509
C                                                                       SP      510
510             301 A=CENTRD(1)                                         SP      511
                B=AAREA(1)                                              SP      512
                IF(NRUN.EQ.1) GO TO 220                                 SP      513
                DO 302 I=2,NRUN                                         SP      514
                C=AAREA(I)/AAREA(1)                                     SP      515
515             IF(C.EQ.B) GO TO 302                                    SP      516
                B=C                                                     SP      517
                A=CENTRD(I)                                             SP      518
302             CONTINUE                                                SP      519
220             RAT=661.595/A                                           SP      520
520             WRITE(6,92) A,RAT                                       SP      521
                GO TO 190                                               SP      522
C                                                                       SP      523
C       SECTION TO FIND 1836 PEAK OF Y(88) IN SPECTRUM TWO AND APPROX.  SP      524
C       KEV/CHANNEL                                                     SP      525
525     C                                                               SP      526
451             J=0                                                     SP      527
452             J=J+1                                                   SP      528
                A=CENTRD(J)                                             SP      529
                B=(1836.0A/RAT)+0.75                                    SP      530
530             IF(A.GT.B) GO TO 453                                    SP      531
                GO TO 452                                               SP      532
453             C=AAREA(J)                                              SP      533
                IF(J.EQ.NRUN) GO TO 221                                 SP      534
                K=J+1                                                   SP      535
535             DO 454 I=K,NRUN                                         SP      536
                IF(AAREA(I).LE.C) GO TO 454                             SP      537
                C=AAREA(I)                                              SP      538
                A=CENTRD(I)                                             SP      539
454             CONTINUE                                                SP      540
```

```
                  221 ARAT=1836.08/A                                                SP       541
                      WRITE(6,455) A                                                SP       542
                      GO TO 100                                                     SP       543
                C                                                                   SP       544
                C     SECTION TO ASSIGN CORRECT GAMMA ENERGIES TO PEAKS IN STANDARDS SP       545
                C     SPECTRUM                                                      SP       546
                C                                                                   SP       547
                  986 WRITE(6,963)                                                  SP       548
                      WRITE (6,82) CSSPOT, ARAT                                     SP       549
                      IF(YSPOT.EQ.0.0) GO TO 309                                    SP       550
                      WRITE(6,455) YSPOT                                            SP       551
                  309 MEMA=0                                                        SP       552
                      MEMB=0                                                        SP       553
                  456 MEMA = MEMA + 1                                               SP       554
                      MEMC = MEMB + 1                                               SP       555
                      MEMD= MEMB + NSTAND(MEMA)                                     SP       556
                      DO 307 I=MEMC, MEMD                                           SP       557
                      MEMB=MEMB + 1                                                 SP       558
                      IF (ENERGY(I) .GT. 1836.08) GO TO 457                         SP       559
                      A=ENERGY(I)/ARAT                                              SP       560
                      GO TO 458                                                     SP       561
                  457 A=ENERGY(I)/ARAT                                              SP       562
                  458 B=APS(CENTRD(1)-A)                                            SP       563
                      J=1                                                           SP       564
                      DO 306 K=2,NRUN                                               SP       565
                      C=ABS(CENTRD(K)-A)                                            SP       566
                      IF(B.LT.C) GO TO 306                                          SP       567
                      B=C                                                           SP       568
                      J=K                                                           SP       569
                  306 CONTINUE                                                      SP       570
                      SENTRD(MEMB)=CENTRD(J)                                        SP       571
                  307 CONTINUE                                                      SP       572
                      IF (MEMB .EQ. NPEAKS) GO TO 459                               SP       573
                      GO TO 456                                                     SP       574
                C                                                                   SP       575
                C     SECTION TO ARRANGE PEAKS IN STANDARDS SPECTRA IN ORDER OF     SP       576
                C     INCREASING GAMMA-RAY ENERGY                                   SP       577
                C                                                                   SP       578
                  459 DO 462 I=1, NPEAKS                                            SP       579
                      K=0                                                           SP       580
                  460 K=K+1                                                         SP       581
                      IF(ENERGY(K).EQ.0.0) GO TO 460                                SP       582
                      A=ENERGY(K)                                                   SP       583
                      L=K                                                           SP       584
                      IF(L.EQ.NPEAKS) GO TO 222                                     SP       585
                      K=K+1                                                         SP       586
                      DO 461 J=K,NPEAKS                                             SP       587
                      IF(ENERGY(J).EQ.0.0) GO TO 461                                SP       588
                      IF(ENERGY(J).GE.A) GO TO 461                                  SP       589
                      A=ENERGY(J)                                                   SP       590
                      L=J                                                           SP       591
                  461 CONTINUE                                                      SP       592
                  222 ABENRG(I)=ENERGY(L)                                           SP       593
                      APCNTR(I)=SENTRD(L)                                           SP       594
                      ENERGY(L)=0.0                                                 SP       595
                      SCNTRD(L)=0.0                                                 SP       596
                  462 CONTINUE                                                      SP       597
                      WRITE (6,83)                                                  SP       598
                  304 WRITE(6,84) (APCNTR(I), ABENRG(I), I=1, NPEAKS)               SP       599
                      DO 310 I=1, NPEAKS                                            SP       600
                      SENTRD(I)= APCNTR(I)/1000.0                                   SP       601
                  310 ENERGY(I) = ABENRG(I)/1000.0                                  SP       602
                      GO TO 92                                                      SP       603
                C                                                                   SP       604
                C     SECTION TO DETERMINE GAMMA RAY ENERGIES                       SP       605
                C                                                                   SP       606
                  311 DO 312 I=1,NRUN                                               SP       607
                  312 CENTRD(I) = CENTRD(I)/1000.0                                  SP       608
                      DO 314 I=1,NRUN                                               SP       609
                      K=0                                                           SP       610
                  313 K=K+1                                                         SP       611
                      L=K+1                                                         SP       612
                      IF (CENTRD(I) .GT. SENTRD(NPEAKS)) GO TO 314                  SP       613
                      IF (CENTRD(I) .GE. SENTRD(L)) GO TO 313                       SP       614
                      A=(SENTRD(L)-SENTRD(K))/(ENERGY(L)-ENERGY(K))                 SP       615
                      B=SENTRD(K)-A*ENERGY(K)                                       SP       616
                      UNERGY(I)=(CENTRD(I)-B)/A                                     SP       617
```

```
      314 CONTINUE                                                      SP        618
          DO 315 I=1,NRUN                                               SP        619
          CENTRD(I) = CENTRD(I)*1000.0                                  SP        620
  320     ENERGY(I) = ENERGY(I)*1000.0                                  SP        621
C                                                                       SP        622
C     SECTION FOR PROCESSING ACTIVATION ANALYSIS DATA                   SP        623
C                                                                       SP        624
          IF(JANAL.GT.0) GO TO 41                                       SP        625
  625     DCYTM=TIMECT(NUMSPC)-START                                    SP        626
          IF(DURLIV(NUMSPC).EQ.0.0) GO TO 640                           SP        627
          TIME=DURLIV(NUMSPC)                                           SP        628
          IF(DLIVTU(NUMSPC).EQ.S) TIME=TIME/60.0                        SP        629
  630     IF(DLIVTU(NUMSPC).EQ.H) TIME=TIME*60.0                        SP        630
          IF(DLIVTU(NUMSPC).EQ.D) TIME=TIME*60.0*24.0                   SP        631
          IF(DLIVTU(NUMSPC).EQ.Y) TIME=TIME*60.0*24.0*365.0             SP        632
          GO TO 645                                                     SP        633
      640 DURLIV(NUMSPC)=TIMECH                                         SP        634
          DLIVTU(NUMSPC)=S                                              SP        635
  635     TIME=TIMECH/60.0                                              SP        636
      645 DO 443 I=1,210                                                SP        637
      443 SCF(I)=0.0                                                    SP        638
          DO 410 I=1,NPKAN                                              SP        639
          SPOT=ABS(ENERGY(1)-GAMKEV(I))                                 SP        640
  650     INDEX=1                                                       SP        641
          DO 411 J=2,NRUN                                               SP        642
          ASPOT=ABS(ENERGY(J)-GAMKEV(I))                                SP        643
          IF(ASPOT.GT.SPOT) GO TO 411                                   SP        644
          SPOT=ASPOT                                                    SP        645
  655     INDEX=J                                                       SP        646
      411 CONTINUE                                                      SP        647
          ECHKA=ENERGY(INDEX)-IFILT                                     SP        648
          ECHKB=ENERGY(INDEX)+IFILT                                     SP        649
          IF(GAMKEV(I).LT.ECHKA.OR.GAMKEV(I).GT.ECHKB) GO TO 410        SP        650
  660     ELNAME(INDEX)=ELEMNT(I)                                       SP        651
          EGAM(INDEX)=GAMKEV(I)                                         SP        652
          UNITS(INDEX)=FNDIM(I)                                         SP        653
          INTEG=AAREA(INDEX)                                            SP        654
          DCONST=LAMBDA(I)                                              SP        655
  665     SIGMA=SIGMA(INDEX)                                            SP        656
          ATIME=1.000-(DEXP(-1.0*DCONST*TIME))                          SP        657
          CTIME=DEXP(DCONST*DCYTM)                                      SP        658
          AF=(CTIME*DCONST)/ATIME                                       SP        659
          CF=INTEG*AF                                                   SP        660
  670     SIGCF=SIGMA*AF                                                SP        661
          IF(NFM.LT.0) GO TO 477                                        SP        662
          IF(NFM.EQ.0) GO TO 500                                        SP        663
          IF(NFM.LE.IFM) GO TO 412                                      SP        664
          SCF(INDEX)=CF/FMWT(I)                                         SP        665
  675     SIGSCF(INDEX)=SIGCF/FMWT(I)                                   SP        666
          JFM=IFM+1                                                     SP        667
          IF(SAMPWT(NUMSPC).EQ.0.0) GO TO 494                           SP        668
          SCF(INDEX)=SCF(INDEX)/SAMPWT(NUMSPC)                          SP        669
          SIGSCF(INDEX)=SIGSCF(INDEX)/SAMPWT(NUMSPC)                    SP        670
  680     CFARAY(I,JFM)=SCF(INDEX)                                      SP        671
          SCFARY(I,JFM)=SIGSCF(INDEX)                                   SP        672
          GO TO 412                                                     SP        673
      412 IF(CFARAY(I,1).EQ.0.0) GO TO 410                              SP        674
          SCF(INDEX)=CF/SAMPWT(NUMSPC)                                  SP        675
  685     SIGSCF(INDEX)=SIGCF/SAMPWT(NUMSPC)                            SP        676
          DATA(INDEX)=SCF(INDEX)/CFARAY(I,1)                            SP        677
          VARACF=SIGSCF(INDEX)**2                                       SP        678
          VARSCF=SCFARY(I,1)**2                                         SP        679
          ACFSQ=SCF(INDEX)**2                                           SP        680
  690     SCFSQ=CFARAY(I,1)**2                                          SP        681
          QUOVAR=SQRT((VARACF/ACFSQ)+(VARSCF/SCFSQ))                    SP        682
          SIGDAT(INDEX)=DATA(INDEX)*QUOVAR                              SP        683
          UNITS1(INDEX)=SMPDIM(NUMSPC)                                  SP        684
          GO TO 410                                                     SP        685
  695     477 IF(.NOT.FLXMON) GO TO 412                                 SP        686
          SCF(INDEX)=CF/FMWT(I)                                         SP        687
          CFARAY(I,1)=SCF(INDEX)                                        SP        688
          SIGSCF(INDEX)=SIGCF/FMWT(I)                                   SP        689
          SCFARY(I,1)=SIGSCF(INDEX)                                     SP        690
  700     GO TO 410                                                     SP        691
      500 SCF(INDEX)=CF/SAMPWT(NUMSPC)                                  SP        692
          SIGSCF(INDEX)=SIGCF/SAMPWT(NUMSPC)                            SP        693
      410 CONTINUE                                                      SP        694
```

```
      C                                                                 SP        695
      C     SECTION FOR PRINTING OUT DATA AND RESULTS                   SP        696
      C                                                                 SP        697
   41 NURRUN=NURRUN+1                                                   SP        698
   68 IF (NSTSPC .EQ. 0) GO TO 75                                       SP        699
      IF(INT.EQ.0) WRITE(6,109) NDTPNT                                  SP        700
      IF(INT.GT.0) WRITE(6,112)                                         SP        701
      WRITE(6,12) NPASS,TAGWRD(1)                                       SP        702
      IF(MPRINT.LT.2) WRITE(6,630)                                      SP        703
      IF(JANAL.GT.0) GO TO 413                                          SP        704
      WRITE(6,499)(SAMPNM(NUMSPC,J),J=1,4)                              SP        705
      WRITE(6,999)SAMPWT(NUMSPC),SMPUI*(NUMSPC),WRCT(NUMSPC),WINCT(NUMSP SP        706
     1C),NURLIV(NUMSPC),DLIVTU(NUMSPC)                                  SP        707
      IF(NFM.LT.0.AND.FLXMON) GO TO 580                                 SP        708
      IF(NFM.LT.0.AND..NOT.FLXMON) GO TO 422                            SP        709
      IF(NFM.EQ.0) GO TO 478                                            SP        710
      IF(NFM.LE.IFM) GO TO 422                                          SP        711
      GO TO 580                                                         SP        712
  478 WRITE(6,414)                                                      SP        713
      GO TO 581                                                         SP        714
  580 WRITE(6,414)                                                      SP        715
  581 LNCNT=0                                                           SP        716
      DO 415 I=1,NPUN                                                   SP        717
      LNCNT=LNCNT+1                                                     SP        718
      IF(LNCNT.GT.50) GO TO 416                                         SP        719
  421 IF(SCF(I).EQ.0.0) GO TO 417                                       SP        720
      WRITE(6,419)CENTRD(I),LCH(I),NCH(I),UNERGY(I),AAREA(I),FLNAM(I),EG SP        721
     1AM(I),SCF(I),SIGSC*(I)                                            SP        722
      GO TO 415                                                         SP        723
  417 WRITE(6,419)CENTRD(I),LCH(I),NCH(I),UNERGY(I),AAREA(I)            SP        724
      GO TO 415                                                         SP        725
  416 IF(NFM.EQ.0) GO TO 585                                            SP        726
      WRITE(6,420)                                                      SP        727
      LNCNT=0                                                           SP        728
      GO TO 421                                                         SP        729
  585 WRITE(6,520)                                                      SP        730
      LNCNT=0                                                           SP        731
      GO TO 421                                                         SP        732
  415 CONTINUE                                                          SP        733
      IFM=IFM+1                                                         SP        734
      GO TO 260                                                         SP        735
  422 WRITE(6,423)                                                      SP        736
      LNCNT=0                                                           SP        737
      DO 424 I=1,NRUN                                                   SP        738
      LNCNT=LNCNT+1                                                     SP        739
      IF(LNCNT.GT.50) GO TO 428                                         SP        740
  426 IF(SCF(I).EQ.0.0) GO TO 425                                       SP        741
      WRITE(6,427)CENTRD(I),LCH(I),NCH(I),UNERGY(I),AAREA(I),ELNAM(I),EG SP        742
     1AM(I),SCF(I),DATA(I),UNITS(I),UNITS(I),SIGDAT(I)                  SP        743
      GO TO 424                                                         SP        744
  425 WRITE(6,419)CENTRD(I),LCH(I),NCH(I),UNERGY(I),AAREA(I)            SP        745
      GO TO 424                                                         SP        746
  428 WRITE(6,429)                                                      SP        747
      LNCNT=0                                                           SP        748
      GO TO 426                                                         SP        749
  424 CONTINUE                                                          SP        750
      GO TO 260                                                         SP        751
  413 WRITE(6,939)                                                      SP        752
      LNCNT=0                                                           SP        753
      DO 250 I=1, NRUN                                                  SP        754
      LNCNT=LNCNT + 1                                                   SP        755
      IF (LNCNT .GT.50) GO TO 333                                       SP        756
  334 WRITE (6,331) CENTRD(I), LCH(I), NCH(I), MCH(I), UNERGY(I),       SP        757
     1AAREA(I), SIGMA(I)                                                SP        758
      GO TO 250                                                         SP        759
  333 WRITE (6,332)                                                     SP        760
      LNCNT=0                                                           SP        761
      GO TO 334                                                         SP        762
  250 CONTINUE                                                          SP        763
      GO TO 260                                                         SP        764
   75 IF(INT.EQ.0) WRITE(6,109) NDTPNT                                  SP        765
      IF(INT.GT.0) WRITE(6,112)                                         SP        766
      WRITE(6,12) NPASS,TAGWRD(1)                                       SP        767
      IF(MPRINT.LT.2) WRITE(6,630)                                      SP        768
      WRITE (6,110)                                                     SP        769
      LNCNT = 0                                                         SP        770
      DO 77 I=1,NRUN                                                    SP        771
```

```
                      LNCNT = LNCNT+1                                                  SP       772
                      IF(LNCNT.GT.60) GO TO 76                                         SP       773
                   79 WRITE(6,111)(CLATRD(I),LCH(I),VCH(I),MCH(I),AAREA(I),SIGMA(I)    SP       774
                      GO TO 77                                                         SP       775
     775              76 WRITE(6,113)                                                  SP       776
                      LNCNT = 0                                                        SP       777
                      GO TO 74                                                         SP       778
                   77 CONTINUE                                                         SP       779
                  260 IF(MENT.GT.0) GO TO 780                                          SP       780
     780              WRITE(6,350) NPASS,TAUMND(I)                                     SP       781
                      LNCNT = 0                                                        SP       782
                      DO 400 J = 1,SIZE,10                                             SP       783
                      LNCNT = LNCNT + 1                                                SP       784
                      NC = J - 1                                                       SP       785
     785              K = J + 9                                                        SP       786
                      IF (LNCNT.GT.61) GO TO 790                                       SP       787
                      WRITE(6,360) (C,(MDATA(I),I=J,K)                                 SP       788
                      GO TO 400                                                        SP       789
                  790 WRITE(6,370) (C,(MDATA(I),I=J,K)                                 SP       790
     790              LNCNT = 0                                                        SP       791
                  400 CONTINUE                                                         SP       792
                      C                                                                SP       793
                      C     ERROR MESSAGES AND JOB TERMINATION                         SP       794
                      C                                                                SP       795
     795              92 IF(JANAL.EQ.0.AND.IWIND.EQ.0) GO TO 100                       SP       796
                      IF(NUMRUN.LT.NSPEC) GO TO 100                                    SP       797
                      GO TO 71                                                         SP       798
                   99 STOP                                                             SP       799
                      END                                                              SP       800
```

SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS DEF LINE REFERENCES

```
 4114  SPECTRA      2
```

VARIABLES SN TYPE RELOCATION

```
10312  A          REAL                    REFS     350     366     430     431     432     433     446
                                           456     470     520     530     540     541     545     565
                                           598     615     616  DEFINED    109     565     573     570
                                           431     437     495     453     510     517     523     538
                                           598     561     582     589     614
 [illegible]  AAREA      REAL      ARRAY   REFS       2     511     514     532     536     547     643
                                           720     723     741     744     756     773
                                        DEFINED    489
10467  ARCNTR     REAL      ARRAY          REFS       6      16     598     600  DEFINED    533
 [illegible]  ARENRO     REAL      ARRAY   REFS       4      14     530     631  DEFINED    544
10315  ALRKG      REAL                    REFS     475     476  DEFINED    474
10412  ACESQ      REAL                    REFS     681  DEFINED     670
10346  ADEL       REAL                    REFS     474     475  DEFINED     466
 [illegible]  AF         REAL              REFS     659     660  DEFINED     648
10351  AHAFL      REAL                    REFS     472  DEFINED     470
10352  AHAFR      REAL                    REFS     473  DEFINED     471
10333  AK         REAL                    REFS     422     427  DEFINED     421     426
 [illegible]  AKL        REAL              REFS     [illegible]     421     435     436  DEFINED     591     619
                                           422
10324  AKP        REAL                    REFS     395     426     428  DEFINED     390     429     427
10345  AL         REAL                    REFS     386     387     362     353     401     402     439
                                        DEFINED     [illegible]     [illegible]     [illegible]     [illegible]
10353  ALRKG      REAL                    REFS     474  DEFINED     472
10275  ALMIN      REAL                    REFS     315  DEFINED     311     316
10064  AP         REAL      ARRAY         REFS       6      16      18     289    4*291
                                        DEFINED    284     284     290
10347  APEAK      REAL                    REFS     459     475     476  DEFINED     467     469
10317  AR         REAL                    REFS     386     362     401     409     411
                                        DEFINED     344     360     375
 [illegible]  ARAT       REAL              REFS     [illegible]     591  DEFINED     107     117     119     540
10354  ARRKG      REAL                    REFS     474  DEFINED     473
10330  AREA       REAL                    REFS     413     409  DEFINED     410     475     479
10363  ARMIN      REAL                    REFS     337  DEFINED     333     338
 [illegible]  ASQRT      REAL              REFS     643     644  DEFINED     640
10174  ATIME      DOUBLE                  REFS      23     658  DEFINED     656
10316  AVRKGL     REAL                    REFS     393     399     401     402     409     411
                                        DEFINED     383     393     399
```

VARIABLES	SN	TYPE	RELOCATION								
10317	AVBKGR	REAL		REFS	396	400	401	409	411		
				DEFINED	384	396	400				
10267	A21	REAL		REFS	292	448	452	DEFINED	291	447	448
10340	A22	REAL		REFS	430	431	432	DEFINED	449	455	
10356	B	REAL		REFS	514	515	530	546	616		
				DEFINED	511	516	529	562	567	615	
10327	BKG	REAL		REFS	410	DEFINED	409				
10324	BKR	REAL		REFS	433	434	DEFINED	428			
6701	BLANK	* REAL		DEFINED	26						
23721	BUF	REAL	*UNDEF	REFS	6						
10357	C	REAL		REFS	515	516	536	566	567		
				DEFINED	514	532	547	553			
10341	CENT	REAL		REFS	454	455	488	DEFINED	452	456	
20741	CENTRD	REAL	ARRAY	REFS	6	510	517	528	538	552	555
				570	607	612	613	616	619	720	723
				741	744	758	775	DEFINED	795	449	607
				619							
10375	CF	REAL		REFS	664	674	686	691	DEFINED	659	
24651	CFARAY	REAL	ARRAY	REFS	10	231	2*232	237	244	245	673
				678	680	DEFINED	147	227	232	237	244
				670	687						
10313	CHECK	REAL		REFS	351	367	DEFINED	350	366		
10213	CSSPOT	REAL		REFS	115	116	495	548	DEFINED	112	
10175	CTIME	DOUBLE		REFS	25	699	DEFINED	697			
6677	D	REAL		REFS	159	630	DEFINED	26			
32471	DATA	REAL	ARRAY	REFS	10	642	701	DEFINED	676		
10172	DCONST	DOUBLE		REFS	23	656	657	658	DEFINED	654	
10369	DCTTM	REAL		REFS	677	DEFINED	825				
31175	DLIVTU	REAL	ARRAY	REFS	12	628	629	630	631	705	
				DEFINED	163	634					
10335	DRVTST	REAL		REFS	435	DEFINED	433				
10310	DYDX	REAL		REFS	347	350	363	348	402	443	448
				472	473	DEFINED	346	362	401		
10371	ECHKA	REAL		REFS	649	DEFINED	647				
10372	ECHKB	REAL		REFS	649	DEFINED	648				
10365	FGAM	REAL	ARRAY	REFS	10	2*14	720	741	DEFINED	691	
24741	ELEMNT	REAL	ARRAY	REFS	10	155	245	650	DEFINED	151	
31651	ELMAX	REAL	ARRAY	REFS	14	720	741	DEFINED	650		
10167	EMAX	INTEGER		REFS	19	122	501	DEFINED	111		
23837	ENERGY	REAL	ARRAY	REFS	6	558	559	561	561	562	567
				588	589	592	2*614	615	DEFINED	129	534
				601							
10242	FLUX	REAL		REFS	210	DEFINED	148				
10200	FLPMON	LOGICAL		REFS	25	210	219	445	707	709	
				DEFINED	210						
20567	FMDIM	REAL	ARRAY	REFS	10	155	652	DEFINED	151		
27357	FMWGT	REAL	*UNDEF	REFS	10						
24565	FMWT	REAL	ARRAY	REFS	10	145	644	645	686	699	
				DEFINED	151						
24423	GAMKEV	REAL	ARRAY	REFS	10	155	245	639	642	2*649	651
				DEFINED	151						
4676	H	REAL		REFS	150	629	DEFINED	24			
25717	HFLF	DOUBLE	ARRAY	REFS	10	23	155	157	158	159	160
				161	DEFINED	151	157	158	159	160	
26063	HFLFTU	REAL	ARRAY	REFS	10	155	157	158	159	160	
				DEFINED	151						
30355	HRCT	INTEGER	ARRAY	REFS	12	20	168	705	DEFINED	153	
10211	I	INTEGER		REFS	98	99	100	124	127	128	129
				140	147	6*151	6*155	3*157	3*158	3*159	3*160
				2*161	8*165	167	3*168	173	183	217	220
				231	3*232	3*233	2*237	2*238	2*244	3*245	262
				264	2*272	281	282	296	292	235	308
				301	319	324	348	364	420	423	424
				428	432	443	516	517	535	537	538
				558	559	561	592	593	2*598	2*600	2*601
				2*607	612	615	2*616	2*619	2*620	637	639
				642	2*649	650	651	652	654	658	665
				670	671	672	676	679	680	686	687
				688	689	719	9*720	5*723	740	12*741	5*744
				7*756	6*773	787	789	DEFINED	97	124	126
				128	129	140	145	150	154	155	162
				166	170	182	226	243	248	258	271
				280	285	294	299	513	535	555	578
				598	599	606	608	618	635	659	716
				777	753	770	787	789			
10212	IBACK	* INTEGER		DEFINED	221						
10241	IFILT	INTEGER		REFS	647	648	DEFINED	148			

VARIABLES	SN	TYPE	RELOCATION							
10262 IFLAG		INTEGER		DEFINED	132	179						
10264 IFF		INTEGER		REFS	215	347	663	666	710	733		
				DEFINED	143	713						
10267 INDEX		INTEGER		REFS	647	648	650	671	672	673	674	
					664	665	2*668	2*669	670	671	674	675
					2*676	677	679	2*682	683	686	687	688
					689	691	692	DEFINED	690	695		
10229 INSKP		INTEGER		REFS	140	141	DEFINED	139				
10220 INT		INTEGER		REFS	662	699	700	764	765			
				DEFINED	137							
10171 INTEG		REAL		REFS	32	659	DEFINED	653				
10265 IPASS		INTEGER		REFS	294	297	298	300	302	303	304	
					505	506	DEFINED	181				
10243 ISKPPY		INTEGER		REFS	147	169	DEFINED	165	167			
10211 ISSPOT		INTEGER		REFS	112	DEFINED	111					
10254 ISTWO		INTEGER		REFS	218	247	444	DEFINED	144	149	229	
10272 I1		INTEGER		REFS	302	303	304	305	DEFINED	301	303	
10260 I4		INTEGER		REFS	468	DEFINED	468					
10226 J		INTEGER		REFS	147	163	231	232	233	245	255	
					259	289	292	294	296	329	307	
					309	310	315	316	317	320	321	323
					325	329	328	337	338	339	349	351
					365	367	388	389	408	421	426	441
					443	527	528	535	570	597	598	
					580	590	642	645	704	744	745	787
					744	DEFINED	146	163	230	245	253	257
					281	286	305	307	314	319	321	324
					326	336	348	364	387	407	420	425
					442	526	527	543	558	586	601	704
					783							
10231 JANAL		INTEGER		REFS	142	184	255	624	703	735		
				DEFINED	139							
10277 JFM		INTEGER		REFS	570	671	DEFINED	666				
10260 K		INTEGER		REFS	262	264	284	288	299	352	354	
					368	370	420	425	443	534	535	555
					568	580	581	582	593	595	598	610
					611	2*614	2*615	787	789	DEFINED	261	282
					287	341	352	357	368	418	423	441
					534	564	579	580	585	609	610	785
10266 KA		INTEGER		REFS	259	259	DEFINED	244				
10273 KL		INTEGER		REFS	311	312	313	343	345	346	347	
					359	361	362	363	371	379	2*379	
					381	383	389	391	407	418	419	464
					485	DEFINED	309	317	371			
10323 KLAV		INTEGER		REFS	391	393	DEFINED	389				
10274 KL0		INTEGER		REFS	325	DEFINED	310					
10276 KL1		INTEGER		REFS	314	364	DEFINED	312	358			
10277 KL2		INTEGER		REFS	316	DEFINED	313					
10312 KR		INTEGER		REFS	333	334	335	342	344	345	346	
					355	360	361	362	378	2*380	382	394
					788	390	407	425	465	486		
				DEFINED	330	332	339	355				
10322 KRAV		INTEGER		REFS	390	396	DEFINED	388				
10304 KR1		INTEGER		REFS	336	348	DEFINED	334	342			
10305 KR3		INTEGER		REFS	336	DEFINED	335					
10269 L		INTEGER		REFS	289	291	298	744	592	595	596	
					595	613	2*614	DEFINED	278	583	590	611
20721 LABEL		INTEGER	ARRAY	REFS	5	140	DEFINED	129				
26145 LAMBDA		DOUBLE	ARRAY	REFS	19	23	654	DEFINED	161			
22071 LCP		INTEGER	ARRAY	REFS	8	720	723	741	744	756	775	
				DEFINED	485							
10715 LDATA		REAL	ARRAY	REFS	6	21	300	302	304	306	320	
				2*325	419	421	424	426	DEFINED	108	292	
10256 LNCNT		INTEGER		REFS	250	251	717	718	735	739	750	
					755	771	772	783	786	DEFINED	255	258
					265	715	717	727	730	736	738	747
					752	754	760	769	771	776	781	783
					790							
10337 LP		INTEGER		REFS	444	446	DEFINED	442	444			
4675 M		REAL		REFS	22	DEFINED	26					
21251 MCH		INTEGER	ARRAY	REFS	6	756	773	DEFINED	487			
0 MDATA		REAL	ARRAY	REFS	6	15	16	21	299	290	311	
					315	316	333	337	338	2*346	347	351
					2*362	363	367	379	380	381	382	383
					384	2*390	2*391	393	396	408	445	461
					469	787	789	DEFINED	22	276	379	380
10360 MEMA		INTEGER		REFS	553	555	DEFINED	551	553			

VARIABLES	SN	TYPE	RELOCATION							
10361 MEMR		INTEGER		REFS	554	555	557	570	572	
				DEFINED	553	557				
10362 MEMC		INTEGER		REFS	556	DEFINED	559			
10363 MEMD		INTEGER		REFS	558	DEFINED	559			
10342 MID		INTEGER		REFS	461	462	463	487	DEFINED	455
30561 MIACT		INTEGER	ARRAY	REFS	12	20	168	705	DEFINED	153
10270 MP		INTEGER		REFS	299	DEFINED	297			
10267 MPRINT		INTEGER		REFS	274	705	767	779	DEFINED	137
10271 MR		INTEGER		REFS	299	DEFINED	298			
10225 MSPEC		INTEGER		REFS	796	DEFINED	136	169		
10216 MSTSPC		INTEGER		REFS	123	124	126	DEFINED	119	121	123
10224 NUMSPC		INTEGER		REFS	180	DEFINED	139			
10262 N1		INTEGER		REFS	274	277	278	DEFINED	276	
10263 N2		INTEGER		REFS	280	285	DEFINED	277		
10265 N		INTEGER		REFS	283	284	445	446	DEFINED	273	283
					443					
10244 NBLOCK		INTEGER		REFS	221	DEFINED	177			
10217 NC		INTEGER		REFS	262	264	787	749	DEFINED	253	784
21561 NCH		INTEGER	ARRAY	REFS	5	720	723	741	744	735	773
				DEFINED	684					
10247 NCHAN		INTEGER		REFS	185	186	2+188	191	194	
10260 NCHECK	*	INTEGER		DEFINED	103					
10232 NDEL		INTEGER		REFS	138	462	463	466	DEFINED	137	
10235 NOTPNT		INTEGER		REFS	698	704	DEFINED	139		
10250 NPAR		INTEGER		REFS	185	186	188	190		
10227 NPM		INTEGER		REFS	149	214	215	230	702	707	661
					662	663	707	708	709	710	725
				DEFINED	148					
10255 NPM1		INTEGER		REFS	244	245	DEFINED	242		
10344 NPAFL		INTEGER		REFS	466	468	470	DEFINED	462	
10345 NPAFR		INTEGER		REFS	465	468	471	DEFINED	463	
10240 NP		INTEGER		REFS	248	249	DEFINED	184	194	
10215 NPASS		INTEGER		REFS	133	180	181	194	222	230	701
					746	748	DEFINED	104	190	222	
10217 NPEAKS		INTEGER		REFS	127	128	572	578	584	585	598
					599	612	DEFINED	125	127		
10240 NPKAN		INTEGER		REFS	150	154	156	226	243	633	
				DEFINED	148						
10246 NPRINT		INTEGER		REFS	253	DEFINED	137				
10221 NR		INTEGER		REFS	237	238	DEFINED	208	397		
10224 NREF		INTEGER		REFS	234	236	237	238	DEFINED	233	234
10212 NREFRR	*	INTEGER		DEFINED	101						
10267 NRUN		INTEGER		REFS	182	185	188	203	484	485	486
					487	488	489	490	512	513	535
					598	606	608	618	641	716	737	753
					770	DEFINED	166	296	480			
31565 NSKIP		INTEGER	ARRAY	REFS	12	167	213	248	DEFINED	153	
10214 NSKP	*	INTEGER		DEFINED	100						
10241 NSPEC		INTEGER		REFS	136	141	142	166	169	171	172
					204	DEFINED	129				
23521 NSTAND		INTEGER	ARRAY	REFS	5	127	555	DEFINED	124		
10243 NSTSPC		INTEGER		REFS	119	119	121	136	455	485	498
					499	505	506	598	DEFINED	102	111	
10253 NUMRUN		INTEGER		REFS	221	222	697	796	DEFINED	223	637
10245 NUMSPC		INTEGER		REFS	208	209	210	211	212	255	525
					526	527	528	538	541	543	545
					557	668	669	674	675	683	691	692
					774	6+705	DEFINED	178	208	224		
31061 NUMLIV		INTEGER	ARRAY	REFS	12	525	627	705	DEFINED	149	633
10212 NTSPOT		INTEGER		REFS	113	DEFINED	111				
10300 N1		INTEGER		REFS	329	330	DEFINED	323			
10301 N2		INTEGER		REFS	329	332	DEFINED	328			
10326 PEAK		REAL		REFS	408	410	411	DEFINED	405	408	
10211 PKSPC		REAL		REFS	213	248	DEFINED	212			
10464 QUOVAR		REAL		REFS	687	DEFINED	681				
10215 RAT		REAL		REFS	520	529	559	DEFINED	116	519	
56740 S		REAL		REFS	197	628	634	DEFINED	96		
27371 SAMPNM		REAL	ARRAY	REFS	12	210	245	726	DEFINED	143	
30211 SAMPWT		REAL	ARRAY	REFS	12	667	668	669	674	675	691
					692	705	DEFINED	153			
32171 SCF		REAL	ARRAY	REFS	14	668	670	676	679	687	719
					726	744	761	DEFINED	657	664	694	696
					686	691					
26311 SCFARY		REAL	ARRAY	REFS	12	2+233	238	678	DEFINED	228	233
					230	671	699				
10413 SCFSQ		REAL		REFS	691	691	DEFINED	690			
23565 SFMTRD		REAL	ARRAY	REFS	6	593	612	613	2+614	615	

VARIABLES SN TYPE RELOCATION

10376	SIGCF	REAL		DEFINED	577	595	600					
34131	SIGDAT	REAL	ARRAY	REFS	665	675	688	692	DEFINED	650		
10375	SIGINT	REAL		REFS	14	741	DEFINED	682				
22711	SIGMA	REAL	ARRAY	REFS	680	DEFINED	693					
33061	SIGSCF	REAL	ARRAY	REFS	6	655	756	773	DEFINED	430		
				REFS	14	669	671	677	689	720		
				DEFINED	665	669	675	688	692			
10369	SIZE	INTEGER		REFS	14	139	177	198	2+199	247	274	
					276	277	297	782	DEFINED	129		
10220	SKP	REAL		REFS	131	134	DEFINED	129	248			
10345	SLPTST	REAL		REFS	2+436	DEFINED	434					
31341	SPOTM	REAL	ARRAY	REFS	12	643	705	DEFINED	183			
0	SPECTR	INTEGER	ARRAY	REFS	6	16	19	185	262	264	270	
					272	DEFINED	98					
10346	SPOT	REAL		REFS	643	DEFINED	639	644				
10179	START	INTEGER		REFS	22	171	523	DEFINED	170	173		
10332	STAT	REAL		REFS	414	DEFINED	413					
24371	TAGWRD	INTEGER	ARRAY	REFS	6	19	254	701	766	793		
				DEFINED	203							
10369	TIME	REAL		REFS	528	629	630	631	638			
				DEFINED	627	628	629	630	631	635		
10261	TIMECH	REAL		REFS	633	635	DEFINED	270				
30665	TIMECT	INTEGER	ARRAY	REFS	12	20	170	173	625			
				DEFINED	448							
10343	TOP	REAL		DEFINED	441							
23221	UNERGY	REAL	ARRAY	REFS	6	620	639	642	647	649	720	
					723	741	744	756	DEFINED	183	616	620
33311	UNITS	REAL	ARRAY	REFS	14	741	DEFINED	632				
33621	UNITS1	REAL	ARRAY	REFS	14	741	DEFINED	693				
10331	VAR	REAL		REFS	412	413	477	490	DEFINED	411	412	
					476	477	480					
10410	VARSCF	REAL		REFS	441	DEFINED	477					
10411	VARSCF	REAL		REFS	681	DEFINED	678					
10004	X	REAL	ARRAY	REFS	6	16	6+447	5+449	DEFINED	445		
6760	Y	REAL		REFS	169	631	DEFINED	26				
10311	YINT	REAL		REFS	354	388	448	448	472	473		
				DEFINED	387	353	402					
10314	YMIN	REAL		REFS	392	DEFINED	381					
10214	YSPOT	REAL		REFS	2+118	549	550	DEFINED	113			
10315	ZMIN	REAL		REFS	444	DEFINED	542					

FILE NAMES MODE

0	INPUT										
2043	OUTPUT										
0	TAPE5	FMT	READS	111	124	128	129	137	148	151	163
2043	TAPE6	FMT	WRITES	179	180	152	153	155	188	191	194
				197	200	249	241	245	254	255	262	264
				429	441	543	548	550	547	548	675	680
				701	702	704	705	712	714	720	723	726
				729	735	741	744	746	751	756	759	764
				765	766	767	768	773	775	780	787	789

EXTERNALS	TYPE	ARGS		REFERENCES						
CREAD		4		185						
DEXP	DOUBLE	1	LIBRARY	20	656	657				
FSD	REAL	1		139						
SMOOTH		2		274						
SORT	REAL	1	LIBRARY	238	381	382	413	432	498	681

INLINE FUNCTIONS	TYPE	ARGS		DEF LINE	REFERENCES						
ABS	REAL	1	INTRIN		390	391	428	562	565	639	642
AMAX1	REAL	0	INTRIN		421	514					
AMIN1	REAL	0	INTRIN		436						
MINO	INTEGER	0	INTRIN		195						

STATEMENT LABELS			DEF LINE	REFERENCES						
6702	2	FMT	27	111	124					
6704	5	FMT	28	139						
6711	4	FMT	29	129						
6714	7	FMT	30	262						
6717	9	FMT	31	264						
6722	10	FMT	32	270						
6730	12	FMT	33	701	766					
6735	13	FMT	34	137						
6740	14	FMT	35	140						
6745	15		37	131						
6252	41		697	496	499	624				

STATEMENT	LABELS		DEF LINE	REFERENCES							
4714	63		264	261							
4724	65		256	257	263						
0	66		100	97							
0	67		183	182							
0	68	INACTIVE	698								
4146	70		121	115							
4175	71		129	119	797						
4153	72		124	120	172						
0	73		127	126							
6534	75		764	698							
6570	76		775	772							
6550	77		778	777	779						
6557	78		773	777							
6743	81	FMT	36	128							
6745	82	FMT	37	520	548						
6755	83	FMT	38	547							
6764	84	FMT	40	578							
6637	92		795	602	779						
0	99	INACTIVE	798								
6650	100		100	240	321	342	373	798	798		
6767	109	FMT	41	699	764						
7003	110	FMT	43	768							
7012	111	FMT	45	773							
7019	112	FMT	46	780	795						
7033	113	FMT	48	775							
0	147		272	271							
4744	149		276	273							
0	150		279	278							
4654	150		253	209							
4746	151		277	275							
4727	152		270	253							
0	153		284	282							
0	154		293	285							
0	155		289	287							
0	156	INACTIVE	303								
0	157	INACTIVE	304								
5035	159		306	308							
0	160	INACTIVE	307	306							
5041	162		309	2*306							
5054	164		318	314	514						
5061	165		322	322							
0	166	INACTIVE	321	320							
5065	168		323	2*320							
5070	171		325	327							
5075	174		328	335							
5102	178		332	329							
5106	179		333	331							
5113	180		334	332	337						
0	192		408	407							
5501	193		479	464	465						
0	194		469	468							
5224	197		395	390							
5242	198		398	387	395						
0	199		446	443							
0	199	INACTIVE	460								
5317	200		491	299	300	302	304	345	341	414	435	438	
				451									
5121	210		341	356									
5146	211		353	348	351								
5153	212		357	354	372								
5177	213		368	354	367								
5204	214		373	370									
0	215		422	420									
0	216		483	482									
5503	220		514	512									
5629	221		540	535									
5723	222		592	588									
6530	230		762	757	758								
6619	260		778	734	759	763							
5546	301		510	500									
5553	302		518	513	515								
5624	305		551	508	525	549							
5670	306		564	564	566								
0	307		571	556									
0	309	INACTIVE	598										
0	310		601	599									
5760	311		606	498	503	506							

STATE	FMT LABELS		DEF LINE	REFERENCES					
0	312		607	606					
5771	313		611	611					
6015	314		617	608	612				
0	315		620	619					
7042	330	FMT	50	751					
7053	331	FMT	52	756					
7011	332	FMT	53	753					
6524	333		758	755					
6505	334		766	761					
7072	350	FMT	55	780					
7113	360	FMT	57	787					
7115	370	FMT	58	789					
6624	390		789	786					
6674	400		791	787	788				
5540	401		504	501					
6536	402		216	211					
6247	410		693	638	649	672	673	684	690
5117	411		646	541	543				
6211	412		677	663	685				
6418	413		753	703					
7112	414	FMT	59	714					
6410	415		732	716	722	724			
6377	416		727	718					
6443	417		744	739					
7125	418	FMT	61	720					
7135	419	FMT	63	725	744				
7142	420	FMT	64	726					
6341	421		717	724	731				
6415	422		735	708	710				
7156	423	FMT	66	735					
6472	424		749	737	743	745			
6427	425		744	740					
6414	426		740	708					
7137	427	FMT	68	741					
6466	428		746	739					
7224	429	FMT	70	746					
4543	430		218	215					
4546	431		221	218	247				
4572	436		235	230	231				
4614	437		239	234	236				
7221	438	FMT	72	240					
7232	439	FMT	74	241					
0	440		245	243					
7226	441	FMT	75	244					
0	442	INACTIVE	248						
6861	443		636	632					
5572	451		526	504					
5573	452		527	531					
5613	453		532	530					
5616	454		539	535	536				
7242	455	FMT	76	541	550				
5636	456		555	2*538	2*543	2*546			
5651	457		561	558					
5654	458		562	560					
5712	459		578	572					
5720	460		580	581					
5721	461		581	586	587	588			
0	462		596	578					
7246	470	FMT	77	151					
7253	471	FMT	78	163					
0	474		168	166					
0	475	INACTIVE	139						
4464	476		185	214	216	219			
6254	477		685	681					
6327	478		712	729					
0	480		161	156					
7260	481	FMT	79	255					
0	482		173	172					
0	483		637	636					
0	484		147	145	146				
7264	485	FMT	80	148					
0	490		151	152					
0	491		163	152					
0	492	INACTIVE	150						
6203	494		670	667					
7244	499	FMT	81	704					
6243	500		691	662					

STATEMENT LABELS			DEF LINE	REFERENCES			
4523	502		208	184			
4531	503		211	209			
4644	510		247	213			
7272	514	FMT	82	712			
7305	520	FMT	84	729			
5530	560		499	495			
4444	561		177	141	142	171	
7311	562	FMT	86	732			
7317	563	FMT	87	597			
6404	565		729	725	•		
6332	580		714	707	711		
6334	581		715	713			
7335	582	FMT	88	153			
0	583		155	154			
7347	584	FMT	90	155			
0	585	INACTIVE	119				
5655	586		547	497			
7355	599	FMT	91	705			
7367	630	FMT	93	767	767		
4211	631		138	133			
4447	632		179	217	220	240	
6054	640		653	626			
5503	790		484	460	478		
7374	801	FMT	149	184			
4576	A02		191	190			
7547	A03	FMT	192	191			
4511	A04		194	190			
7553	A05	FMT	195	194			
4514	A06		197	190			
4521	A07		203	186			
7566	A08	FMT	198	197			
4515	A09		240	140	193	194	199
7575	A10	FMT	201	200			

LOOPS	LABEL		INDEX	FROM-TO	LENGTH	PROPERTIES	
4120	66		I	97 100	3B	OPT	
4164	73		I	126 127	2B	OPT	
4227	484	•	I	145 147	13B		NOT INNER
4234	484		J	146 147	2B	OPT	
4247	490	•	I	150 151	2QB		EXT REFS
4274	583	•	I	154 155	2QB		EXT REFS
4322	480		I	156 161	41B	OPT	
4345	491	•	I	162 163	32B		EXT REFS NOT INNER
4370		•	J	163 163	1QB		EXT REFS
4422	474		I	166 168	5B	OPT	
4437	482		I	172 173	4B	OPT	
4455	67		I	182 183	2B	OPT	
4562	437	•	I	234 238	53B		EXT REFS NOT INNER
4565	436		J	239 235	7B	OPT	
4616	440	•	I	243 245	25B		EXT REFS NOT INNER
4627		•	J	245 245	1QB		EXT REFS
4645		•	J	255 255	1QB		EXT REFS
4700	65	•	J	257 266	27B		EXT REFS
4734	147		I	271 272	2B	OPT	
4760	153		K	282 284	3B	OPT	
4766	154	•	I	285 293	23B		NOT INNER
4773	155		K	287 289	4B	OPT	
5014	149		I	294 295	2B	OPT	
5024	200	•	I	299 491	476B		EXT REFS NOT INNER
5041	163		J	314 318	5B	OPT	
5113	180		J	336 340	5B	OPT	
5137	211		J	348 353	10B	OPT	
5170	213		J	364 369	13B	OPT	
5264	100		J	397 398	2QB	OPT	
5311	182		J	417 418	3B	OPT	
5353	215		J	420 422	4B	OPT	
5385	216		J	425 427	4B	OPT	
5424	145		K	443 446	6B	OPT	
5455	184		I4	468 469	3B	OPT	
5555	302		I	513 518	7B	OPT	
5612	454		I	529 534	5B	OPT	
5643	307	•	I	564 571	32B		NOT INNER
5603	306		K	569 569	1QB	OPT	
5715	462	•	I	576 596	27B		NOT INNER
5715	461		J	585 591	5B	OPT	
5736		•	I	598 598	1QB		EXT REFS
5752	310		I	599 601	4B	OPT	

LOOPS	LABEL		INDEX	FROM-TO	LENGTH	PROPERTIES
5763	312		I	607	36	OPT
5770	314	•	I	617	21B	
6015	315		I	420	30	OPT
6069	413		I	657	25	OPT
6079	410	•	I	693	162B	EXT REFS NOT INNER
6102	411		J	695	6B	OPT
6234		•	J	704	10B	EXT REFS
6329	412	•	I	732	22B	EXT REFS
6411	414	•	I	749	54B	EXT REFS
6512	250	•	I	762	31B	EXT REFS
6554	77	•	I	778	27B	EXT REFS
6618	400	•	J	791	27B	EXT REFS

COMMON BLOCKS	LENGTH
/ /	4107

STATISTICS

PROGRAM LENGTH	30335B	12509
BUFFER LENGTH	4106B	2118
CP BLANK COMMON LENGTH	10013B	4107

```
      SUBROUTINE SMOOTH     73/73   OPT=1 TRACE                FTN 4.6+433       08/15/77 .16.19.53       PAGE     1

        1           C
                    C     SUBROUTINE TO SMOOTH SPECTRAL DATA                                                  SP       A01
                    C     FIVE POINT CONVOLUTION TECHNIQUE                                                    SP       A02
                    C                                                                                         SP       A03
                                                                                                              SP       A04
        5           C     SUBROUTINE SMOOTH (N,M1)                                                            SP       A05
                    C     DIMENSION MDATA(*100),AP(7)                                                         SP       A06
                    C     COMMON MDATA,AP                                                                     SP       A07
                    C     REAL MDATA                                                                          SP       A08
                          M1 = N - 2                                                                          SP       A09
       10                 DO 1 I = 2,5                                                                        SP       A10
                          J = I - 1                                                                           SP       A11
                        1 AP(I)=MDATA(J)                                                                      SP       A12
                          DO 3 I = 3,M1                                                                       SP       A13
                          J = I + 2                                                                           SP       A14
       15                 DO 2 K = 1,4                                                                        SP       A15
                          KA = K + 1                                                                          SP       A16
                        2 AP(K) = AP(KA)                                                                      SP       A17
                          AP(5)=MDATA(J)                                                                      SP       A18
                          A2=17.0*AP(3)+12.0*(AP(2)+AP(4))-3.0*(AP(1)+AP(5))                                  SP       A19
       20                 MDATA(I)=A2/35.0                                                                    SP       A20
                        3 CONTINUE                                                                            SP       A21
                          RETURN                                                                              SP       A22
                          END                                                                                 SP       A23


      SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS      DEF LINE     REFERENCES
     3  SMOOTH       5          24

VARIABLES     SN  TYPE          RELOCATION
IDENT    AP       REAL      ARRAY   / /          REFS                 7         17        5*19    DEFINED      12        17
                                                                     1*M
    21   AP0      REAL                           REFS          20    DEFINED     19
    14   I        INTEGER                        REFS          11       12       20      DEFINED      10        15
    16   J        INTEGER                        REFS          12       18    DEFINED     11         14
    17   K        INTEGER                        REFS          15       17    DEFINED     15
    20   KA       INTEGER                        REFS          17    DEFINED     16
     6   MDATA    REAL      *ARRAY  / /          ****          7         8        10        18
                                                DEFINED       20
     8   M1       INTEGER           S,P,       REFS          13    DEFINED      5         9
     9   N        INTEGER           S,P,       REFS          1     DEFINED      5

STATEMENT LABELS              DEF LINE    REFERENCES
     6  1                        12          10
     8  2                        17          15
     9  3                        21          13

LOOPS  LABEL    INDEX      FROM-TO     LENGTH     PROPERTIES
    12  1        I          11 13        4*        OPT
    33  3      * I          14 21       24*        NOT INNER
    25  2        K          15 17        4*        OPT

COMMON BLOCKS    LENGTH
     / /           *100

STATISTICS
  PROGRAM LENGTH                 72*        62
  CM BLANK COMMON LENGTH       10015*      *100
```

```
      SUBROUTINE CREAD      77/77   OPT=1 TRACE                    FTN 4.6+457        08/15/77 .16.19.55        PAGE     1

        1                   SUBROUTINE CREAD(IDATA,NCHAN,NRUN,NERR)                                    CREAD      2
                            INTEGER IDATA(4116)                                                        CREAD      3
                      C     READ NRUN FROM TITLE CARD                                                  CREAD      4
                            READ(*,5)NRUN                                                              CREAD      5
        6                   IF (EOF(5LINPUT).NE.0) STOP                                                CREAD      6
                        6   FORMAT(32X,I4)                                                             CREAD      7
                            NERR=0                                                                     CREAD      8
                            DO 14 I=1,4200,10                                                          CREAD      9
                            N=I-1                                                                      CREAD     10
       10                   READ(5,9)(IDATA(N+J),J=1,10)                                               CREAD     11
                        9   FORMAT(10I6)                                                               CREAD     12
                            IF(EOF(5LINPUT).NE.0)GO TO 20                                              CREAD     13
                            K=N+10                                                                     CREAD     14
                       14   CONTINUE                                                                   CREAD     15
       15                   NCHAN=-1                                                                   CREAD     16
                      C     TOO MANY CARDS, SHOULD NEVER SEE THIS                                      CREAD     17
                            NERR=3                                                                     CREAD     18
                            RETURN                                                                     CREAD     19
                       20   NCHAN=K                                                                    CREAD     20
       20                   DO 21 L=1,10                                                               CREAD     21
                            IF((IDATA(NCHAN).AND.MASK(2)).EQ.0)GO TO 22                                CREAD     22
                            NCHAN=NCHAN-1                                                              CREAD     23
                       21   CONTINUE                                                                   CREAD     24
                       22   RETURN                                                                     CREAD     25
       25                   END                                                                        CREAD     26


      SYMBOLIC REFERENCE MAP (R=2)

ENTRY POINTS      DEF LINE      REFERENCES
     6 CREAD             1      15

VARIABLES      SN  TYPE              RELOCATION
   115  I          INTEGER                        REFS          6    DEFINED       8
     0  IDATA      INTEGER   ARRAY     F.P.       REFS          9      21    DEFINED       1      10
   117  J          INTEGER                        REFS         10    DEFINED      10
   120  K          INTEGER                        REFS         19    DEFINED      13
   121  L        * INTEGER                      DEFINED        20
   116  N          INTEGER                        REFS         12      15    DEFINED       0
     0  NCHAN      INTEGER             F.P.       REFS         21      22    DEFINED       1      15      19      22
     0  NERR       INTEGER             F.P.     DEFINED         1       7          17
     0  NRUN       INTEGER             F.P.     DEFINED         1       4

FILE NAMES      MODE
     TAPE5      FMT                    READS          4       10

EXTERNALS       TYPE    ARGS      REFERENCES
     EOF        REAL       1                        12

INLINE FUNCTIONS    TYPE    ARGS      DEF LINE   REFERENCES
     MASK      NO TYPE       1  IN 14              21

STATEMENT LABELS              DEF LINE   REFERENCES
   101  5        FMT              6         4
   112  9        FMT             11        10
     1  14                       14         9
    55  20                       19        12
     0  21                       23        20
    66  22                       24        21

LOOPS  LABEL    INDEX      FROM-TO     LENGTH     PROPERTIES
    25  14      * I          8 19        25R                    EXT REFS     EXITS   NOT INNER
    38          * J         11 19        10R                    EXT REFS
    60  21      * L         20 23         6R          OPT        EXITS

STATISTICS
     PROGRAM LENGTH              1176       9K
```

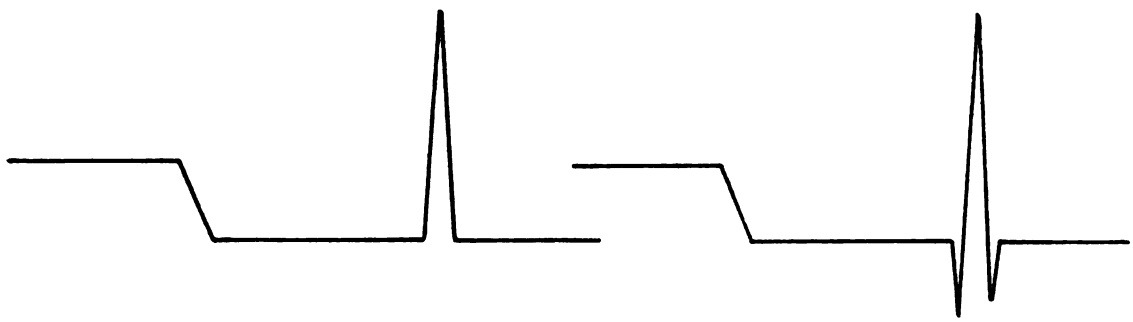
APPENDIX C

A DESCRIPTION AND LISTING OF THE PROGRAM SPECTRA

by Philip A. Baedecker

The program SPECTRA is a program for the analysis of gamma-ray spectra. The program searches out photopeaks, calculates the area under the photopeak, determines the centroid of the photopeak, and, when spectra of gamma-ray standards are included on the input tape, determines the energies of the photopeaks. In an activation analysis experiment, the program will calculate concentrations of the elements to be determined.

The basic program analyzes a spectrum in the following way. The spectral data are read into the computer from magnetic tape, and the data can then be smoothed by a five point convolution technique, using the method of Savitsky and Golay (Anal. Chem. 36, 1627 (1964)). This smoothing operation may be bypassed however, since, for very narrow peaks, the smoothing operation may cause the spectrum to undershoot the baseline on either side of the peak.



In general, for 1024 channel Ge(Li) spectra, it is best to avoid smoothing; for 4096 spectra smoothing is often desirable to help eliminate fluctuations that might be recognized as peaks by the program.

After smoothing the program uses a five-point convolution technique to determine the first derivative at each channel. The program then locates possible peaks by determining where the first derivative changes sign from positive to negative. The program, having located a possible peak, determines the minimum on each side of the peak as the peak limits. Since there is some danger that the right boundary may be too far out, the routine tests the first derivative of each channel on the right hand side of the peak against the first tangent with a negative slope on the left. If the left hand tangent has a slope X, the program locates the first channel on the right with a tangent which has slope -X. If this channel number is less than the previously fixed right boundary, the boundary is considered to be the newer lower channel number.

The program then draws a linear baseline between the boundary channels and checks the right side of the peak to see if any channel falls below the baseline. If this occurs, the boundary channel is decreased by one and the operation repeated until all channels of the right hand side are above the baseline. A similar process is then carried out on the left hand side of the peak.

The next section of the program tests the peak to decide whether or not it is a valid peak. The first test is a statistical test. The peak area and base area are calculated as follows:

$$\text{Base Area} = 0.5 \times \left(D_l + D_r\right) \times (R - L + 1)$$

where

D_l = counts in left hand boundary channel (L)

D_r = counts in right hand boundary channel (R)

$$\text{Peak Area} = \sum_{l} D_i - \text{base area}$$

A standard deviation of the peak is then calculated.

$$\text{SIGMA} = \sqrt{\sum_{L}^{R} D_i + \left(D_l + D_r\right) \times \left(\frac{R - L - 1}{2}\right)^2 - 1}$$

A statistical test is then applied to the peak. To be valid a peak must meet the following criterion:

$$\text{Peak Area} > (5.0 \times \text{SIGMA})$$

The primary purpose of the next test is to eliminate Compton edges, but it is also useful in eliminating spurious peaks which get by the first test. The maximum first derivative is located on the left side of the peak, the minimum derivative on the right. This gives a measure of the slope at the point of inflection on each side of

the peak. The absolute values of the slopes are then compared to see that they are the same within a factor of 1.5.

After a peak has passed the previous two tests, a linear baseline is defined for the photopeak. Several channels on each side of the peak are used to define the background. Four channels to the left of the left peak limit are examined. Those channels which have the same total accumulated counts as the left hand boundary channel (within two standard deviations) are averaged and this average is taken as a new value for the counts accumulated at the left hand peak limit and used to define the background. The process is repeated on the right hand side of the photopeak. The baseline is then a straight line drawn between the two boundary channels.

The centroid of the peak is determined next using seven channels (three on each side of the channel with the greatest number of counts) using the method of Savitzky and Golay.

Two methods for measuring the intensity of the peak are built into the program, to be selected by the user. These methods are discussed in Appendix A, where they are referred to as the "Total Peak Area Method" and the "Wasson Method." In the total peak area method, the area is calculated as for the statistical test above, but using the redefined values for the boundary channels. In

the Wasson method a number of channels, specified by the user, are taken as the limits of integration. The baseline is determined as above, and using that baseline, background is subtracted from the channels within the selected limits for integration. The peak area is calculated as:

$$\text{Base Area} = (B_l + B_r) \times (X + 0.5)$$

$$\text{Peak Area} = \sum_{l-X}^{l+X} D_i - \text{base area}$$

where

l = centermost channel

X = number of channels on each side of centermost channel to be included in peak area determination

D_i = number of counts in ith channel

B_l and B_r = calculated values for the background in channels l-X and l+X respectively, computed from a straight line drawn between the peak limits

$$\text{SIGMA} = \sqrt{\sum_{l-X}^{l+X} D_i + (B_l + B_r) \times (X + 0.5)}^{\,2}$$

The program will determine the energies of the photopeaks from their centroids, provided spectra of gamma-ray standards are included on the input tape. If the energies are to be determined, the first spectrum on the tape should be a ^{137}Cs spectrum. This spectrum serves as

a reference spectrum to provide an approximate energy calibration for locating peaks of known energy in the standard spectra. Due to possible nonlinearity in the spectrometer, ^{88}Y must be provided as a second reference spectrum if any of the photopeaks used as standards have energies greater than 1836 Kev (the Cs and Y reference spectra may be deleted providing the centermost channels for the Cs (and Y) peaks are included as input data on the first data card).

Following the reference spectra on the tape, up to 20 spectra of standards may be used to obtain the energy calibration of spectrometer. The energies of the gamma-rays in each standards spectrum are read in on data cards and the correspondong centroids are determined. Up to 50 lines may be used in the standards spectra.

The program determines the gamma-ray energies in all subsequent spectra by interpolation, assuming that the energy calibration is linear between standard lines. The program does not extrapolate beyond the highest energy of the gamma-ray standards.

When the program is used to process spectral data from an activation analysis experiment, the energies of the gamma-ray lines to be used in the analyses are read in on one set of data cards. Along with the gamma-ray energy, the chemical symbol of the element to be determined, the half-life of the radioisotope used and the weight (or

concentration) of the element in the flux monitor are included on the data card. A second set of data cards (one card for each spectrum on the tape) provides the following information: (1) the name of the sample; (2) the weight of the sample; (3) the units in which the sample weight is expressed; (4) the time of day the sample was counted; (5) the live-time duration of the count. If the weights of the elements in the flux monitors are constant from flux monitor to flux monitor, they may be entered in the appropriate field on the first set of data cards, and no sample weights entered for the flux monitors in the second set of data cards. If the flux monitors used are standard powders or solutions, the concentrations of the elements in the standard powder or solution may be placed on the first set of data cards, and the weights of the standard powders or solutions entered in the sample weight field on the second set of data cards.

The program first makes a pass through the spectra on the tape and analyzes the flux monitor spectra. It picks out the peaks of interest in each flux monitor spectrum and calculates the "standard comparator factor" for each peak. After all flux monitor spectra have been analyzed (the program can process up to ten), the standard comparator factors for each peak are averaged to yield an "average standard comparator factor" for each peak, which

is used to calculate concentrations when the sample spectra are analyzed.

The tape is now backspaced to the first spectrum for the activation analysis run under consideration (there may be more than one run on a given reel of tape). Each sample spectrum is then analyzed in turn, an "analytical comparator factor" calculated for each peak of interest, and, from the average standard comparator factors previously calculated, concentrations are calculated for the elements of interest in the sample. The standard deviation for the concentration is calculated based on counting statistics alone.

In an activation analysis experiment involving the counting of short-lived nuclides, one flux monitor may be irradiated and counted for each sample analyzed. The program will handle this as a special case, as follows: in an activation analysis run a card is provided which tells the computer how many flux monitor spectra are in a given run. If this number is put to less than zero, the program expects that every other spectrum will be a flux monitor spectrum, and that each sample spectrum will follow its corresponding flux monitor spectrum on the tape.

If, for some reason, no flux monitors were counting during an activation analysis run, the number of flux monitor spectra can be put equal to zero on the data card

and the program will process the spectra and calculate analytical comparator factors for each peak in each sample spectrum.

In calculating standard comparator factors and analytical comparator factors, the program corrects the counting data for decay during the count and corrects each count for decay back to the time of the start of the first count in the activation analysis run. That is:

$$SCF = \frac{(\text{peak area}) \times \lambda e^{\lambda t_1}}{(\text{flux monitor weight}) \times (1 - e^{-\lambda t_a})}$$

$$ACF = \frac{(\text{peak area}) \times \lambda e^{\lambda t_1}}{(\text{sample weight}) \times (1 - e^{-\lambda t_a})}$$

where

t_1 = the elapsed time between the start of the first count in the experiment and the start of the count being processed

t_a = duration of the count (live time)

The program operates on various different levels of complexity, determined by the input data cards. It can simply read spectra and analyze the spectral data. On another level the program will determine the energies of the photopeaks in the spectra, provided appropriate standard spectra are provided. Thirdly, the program can process spectral data from an activation analysis experiment and calculate element concentrations. Some output

Table 1.--Gamma Rays Used as Primary Energy Standards.

Source	Energy (keV)	Source	Energy (keV)
^{241}AM	59.536 ± 0.001	^{192}Ir	468.060 ± 0.010
^{109}Cd	88.034 ± 0.010	Annihilation	511.003 ± 0.002
^{182}Ta	100.106 ± 0.001	^{207}Bi	569.690 ± 0.030
^{57}Co	122.046 ± 0.020	^{208}Tl	583.139 ± 0.023
^{144}Ce	133.503 ± 0.020	^{192}Ir	604.378 ± 0.020
^{57}Co	136.465 ± 0.020	^{192}Ir	612.430 ± 0.020
^{141}Ce	145.442 ± 0.010	^{137}Cs	661.615 ± 0.030
^{182}Ta	152.435 ± 0.004	^{54}Mn	834.840 ± 0.050
^{139}Ce	165.852 ± 0.010	^{88}Y	898.023 ± 0.065
^{182}Ta	179.393 ± 0.003	^{207}Bi	1063.635 ± 0.040
^{182}Ta	222.110 ± 0.003	^{60}Co	1173.231 ± 0.030
^{212}Pb	238.624 ± 0.008	^{22}Na	1274.550 ± 0.040
^{203}Hg	279.179 ± 0.010	^{60}Co	1332.508 ± 0.015
^{192}Ir	295.938 ± 0.010	^{140}La	1596.200 ± 0.040
^{192}Ir	308.440 ± 0.010	^{124}Sb	1691.022 ± 0.040
^{192}Ir	316.490 ± 0.010	^{88}Y	1836.127 ± 0.050
^{131}I	364.491 ± 0.015	^{208}Tl	2614.708 ± 0.050
^{198}Au	411.792 ± 0.008	^{24}Na	2754.142 ± 0.060

options are included in the program in that the user can tell the program whether or not to print out the raw spectral data and whether or not to print out the smoothed data.

The following data cards must be provided by the user.

Card		Format
1	Card with number of reference and standards spectra on tape, and the maximum gamma-ray energy of standard lines. The centermost channels of the Cs (and Y) peaks may be included on this card if the gamma-ray energies are to be determined and the Cs (and Y) reference spectra are left off the input tape. IF THE NUMBER OF STANDARDS SPECTRA IS ZERO, OMIT CARDS 2 AND 3.	4I6
2	Cards with the number of photopeaks in each standards spectrum.	12I6
3	Cards with the energies of the photopeaks in each standards spectrum in the order that the spectra appear on the tape.	8Fg.3
4	Card with number of spectra to be skipped, size of spectra, number of spectra (not including reference and	3I6, 13A4

Card		Format
	standard spectra or any spectra to be skipped following the first spectrum in an activation analysis run), and labels.	
5	Card which controls various options built into program. The first four columns control the operation of the program. Column 1--zero-raw data printed out one-raw data not printed out Column 2--zero-smoothed data printed out one-smoothed data not printed out two-smoothing operation deleted Column 3--zero-peak area integrated by Wasson's method one-peak integrated to give total peak area Column 4--zero-program will calculate elemental concentrations from activation analysis data one-activation analysis program bypassed Column 5--10--integer number of points on each side of center channel to be included in peak area determined by Wasson's method	4I1,I6

Card		Format
	Cards 4 and 5 may be repeated together any number of times. If there is more than one run per tape, place a card with a negative number (I6 format) between each set of data cards.	
6	If column 4 of card 5 is 1, omit this and the following data cards:	
	Card with number of flux monitor spectra on tape and number of peaks in spectra which are to be analyzed. Columns 13-17 must have the letters SMHDY in that order. Columns 18-21 must have the first four characters to be used in the sample name field on cards described under 8 below, which identify the flux monitor spectra. The card for each flux monitor spectrum must have the first four characters in the sample name field, identical to the listed characters.	2I6,5A1, A4
7	Set of cards, one for each peak to be analyzed. Each card must have:	A2,BX, F10, 3, 3X, F7, 3, 4X, A1, 5X F10, 5, 6X, A4
	a. chemical symbol for element being determined using given photopeak	
	b. energy of photopeak in KeV	
	c. radioisotope half-life	
	d. units in which half-life is expressed (S = seconds, M = minutes, H = hours, D = days, Y = years)	

Card		Format
	e. flux monitor weight or concentration	
	f. units in which flux monitor weight is expressed (µg, mg, etc.)	µg, mg
8	Set of cards, one for each spectrum on tape in order of spectra appearance on tape. Each card must have:	4A4, 4X, F10.5, 6X, A4, 5X, I3, I2, 3X, I6, A1, 4X, I6
	a. sample name	
	b. sample weight	
	c. sample weight units (GRAM, Mg, etc.)	
	d. time of day count started--8:30 PM would be represented as 2030. If an activation analysis experiment runs into another day, time of second day starts at 2400 (e.g., 1:00 AM would be 2500)	
	e. live time duration of count	
	f. live time units	
	g. spectra to be skipped after processing spectrum described by card.	