MASTER

IS-T- 891

Inductively coupled plasma-atomic emission spectroscopy: A computer controlled, scanning monochromator system for the rapid determination of the elements

by

March Alan Floyd

M.S. Thesis submitted to Iowa State University

Ames Laboratory, DOE
Iowa State University
Ames, Iowa 50011

Date Transmitted: March 1980

PREPARED FOR THE U.S. DEPARTMENT OF ENERGY
UNDER CONTRACT NO. W-7405-eng-82

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

DISCLAIMER

Printed in the United States of America

Available from
National Technical Information Service
U.S. Department of Commerce
5265 Port Royal Road
Springfield, VA 22161

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Inductively coupled plasma-atomic emission spectroscopy: A computer controlled, scanning monochromator system for the rapid determination of the elements[1]

Mark Alan Floyd

Under the supervision of V. A. Fassel
From the Department of Chemistry
Iowa State University

A computer controlled, scanning monochromator system specifically designed for the rapid, sequential determination of the elements is described. The monochromator is combined with an inductively coupled plasma excitation source so that elements at major, minor, trace, and ultratrace levels may be determined, in sequence, without changing experimental parameters other than the spectral line observed. A number of distinctive features not found in previously described versions are incorporated into the system here described. Performance characteristics of the entire system and several analytical applications are discussed.

---

[1] USDOE Report IS-T-891. This work was performed under Contract W-7405-eng-82 with the Department of Energy.

CHAPTER I. INTRODUCTION

Radiofrequency excited, argon supported, inductively coupled plasmas (ICP) are being used to a rapidly increasing extent as excitation sources for the atomic emission determination of the elements at all concentration levels (1-5). Most existing ICP-AES facilities use polychromators (5) for the simultaneous determination of as many as 30 or more elements. These instruments are well-suited for the routine, simultaneous determination of the same set of elements in matrices of similar composition. However, their application for the determination of a broader range of elements in samples of widely varying composition is restricted by the fixed array of exit slits employed for isolation of the spectral lines.

In principle, linear-scan monochromators can be used for the sequential determination of most of the elements in the periodic table, and a number of lines may be measured for each element to enhance the reliability of the determinations. However, for multielement determinations, the operations involved in a sequential scan to the spectral lines of interest and measuring the intensities of the lines relative to the spectral background usually requires constant operator attention and lengthens the analysis time considerably, as compared to the time required for simultaneous multielement determinations with a polychromator.

Automation of the entire analytical cycle is particularly attractive with ICP-AES. As the monochromator sequences from one element to another, optimization of source characteristics or other experimental parameters is normally not required (1-5).

An attractive solution to this loss of versatility and automation would be an instrument that possesses the following characteristics and capabilities:

1. A computer-controlled scanning monochromator with a programmed, non-linear scanning capability between selected wavelengths.
2. Automated peak seeking routine for determination of the peak intensity of selected lines.
3. Storage in memory of a selected, ordered list of the most prominent spectral lines for each of the 70 or so elements determinable by ICP-AES.
4. Preselection of one or more of the stored analyte lines for each element to be determined.
5. Preselection of precise wavelengths at which the spectral background intensity measurement is to be made for each analysis line.
6. Computer software rearrangement of the selected analysis lines and corresponding background wavelengths in ascending wavelength to facilitate orderly measurements and to

minimize the analysis time.

7. Video terminal for the visualization of difficult background situations and the goodness of fit of analytical calibration curve data. The video terminal also aids the judicious selection of the most useful lines for a particular analytical problem.

8. Detection limit routine for monitoring instrument performance. Detection limit measurements that are based on signal to noise considerations provide a convenient and sensitive performance test of the entire system.

9. A signal measurement range of 10^6 for exploitation of the wide dynamic range characteristics of the ICP excitation source.

Computer controlled monochromator systems that have some of the features and characteristics discussed above have been described (6-9), suggested (10), or are commercially available, but none of the versions previously described or marketed have possessed all of the capabilities specified above. Moreover, documentation of the analytical performance of these systems for atomic emission spectrometry has been meager; only the papers by Spillman and Malmstadt (6) and by Kawaguchi, et al. (9), provide data on simple synthetic solutions prepared in the laboratory. In this thesis a monochromator system that provides all of the capabilities discussed above is described. The performance characteristics of the entire system as well as its application to several analytical problems are summarized.

CHAPTER II. INSTRUMENTATION AND SOFTWARE

Experimental Facilities

The experimental facilities and operating conditions utilized in this work are summarized in Table I and a block diagram of the analytical system is illustrated in Figure 1. A double monochromator was used because this instrument was immediately available for use. Any monochromator with a computer controlled stepper motor with appropriate specifications should be adaptable. The original stepper motor, as supplied by GCA-McPherson, was used for this work.

Software

Control software

The software developed for use with the monochromator system is written in a high level language (FORTRAN) and segmented so that changes in analytical procedures, computations, or reporting of final results can be made conveniently. The software described in this chapter consists of one main program (SURVEY), which is broken down into several smaller routines. A block diagram of the software is shown in Figure 2. The operator can invoke each routine by typing the first four letters of the desired routine on the

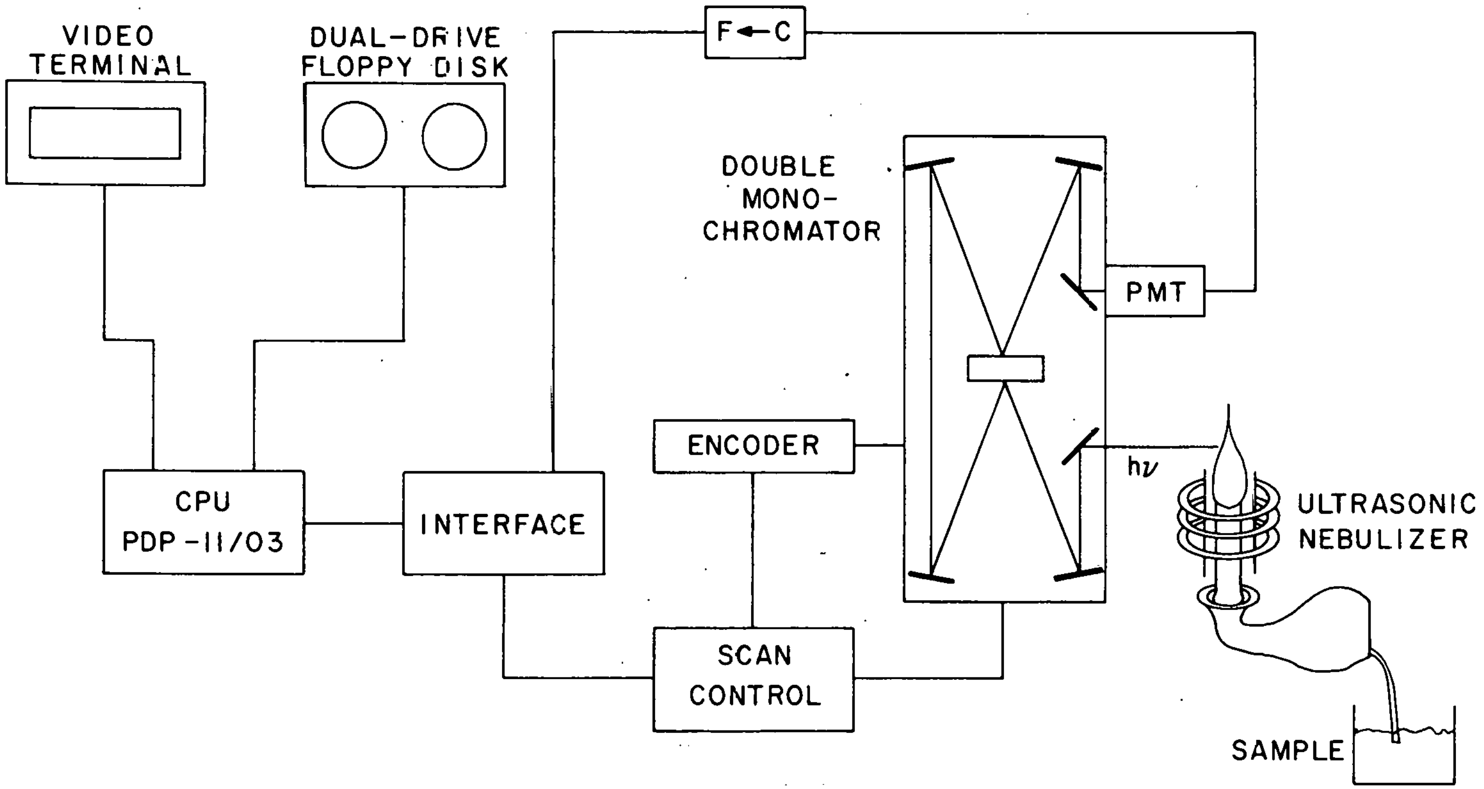

Figure 1. A block diagram of the computer controlled monochromator system

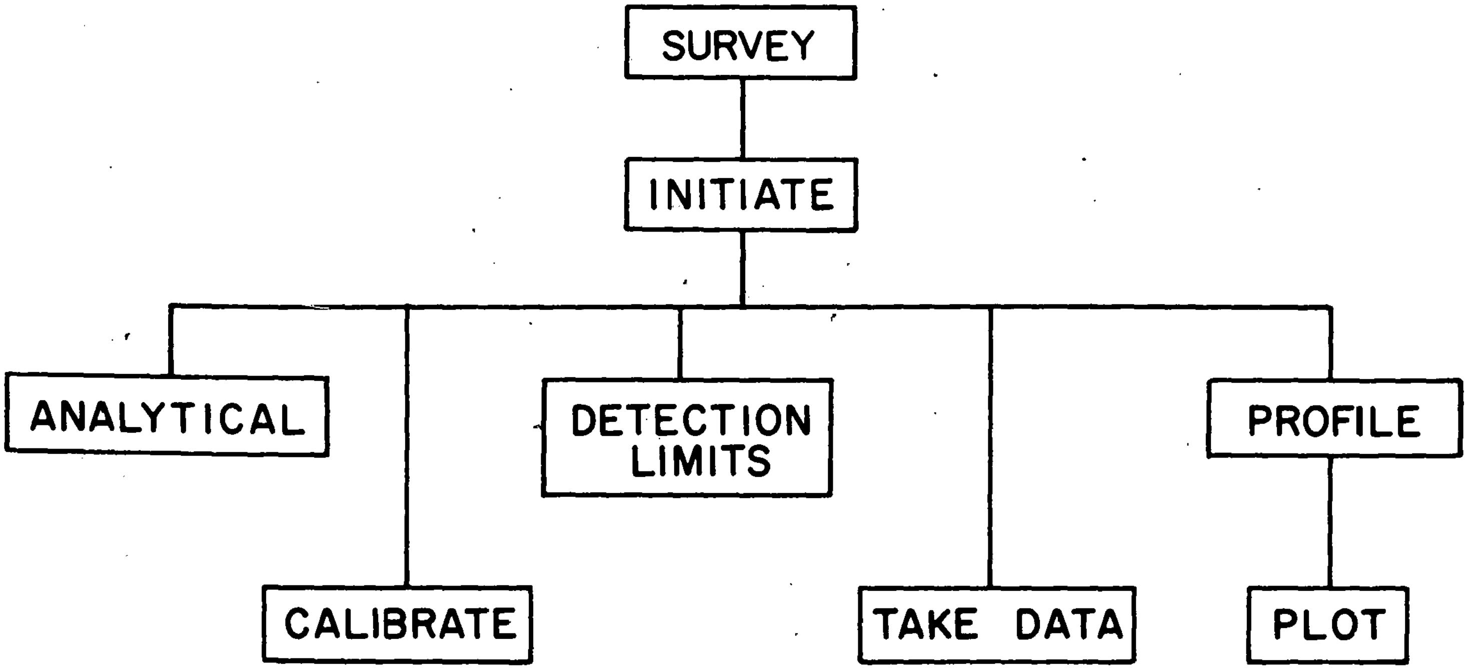

Figure 2. A block diagram of the software system

Table I. Experimental Facilities and Operating Conditions

Spectrometer	0.5 m GCA-McPherson (Acton, Ma) Model 285 double Czerny-Turner monochromator.
Scan Controller	Model 785 GCA-McPherson scan controller.
Encoder	Sequential Information Systems, Inc. (Elmsford, NY) Series 25G optical incremental encoder with 2 tracks phase shifted 90 degrees for direction sensing and count multiplication. The encoder has a resolution of 0.00125 nm/encoder pulse and a frequency response of 200,000 pulses/sec. Supplied as a standard item with the GCA monochromator
Computer	Digital Equipment Corp. (Maynard, Ma) PDP-11/03 minicomputer with 24K memory.
Interface	Ames Laboratory design.
Input/Output	Tektronix (Beaverton, Or) Model 4006-1 Computer Display Terminal.
Grating	1200 G/mm plane grating blazed for 300 nm.
A/D converter	Analog Technology Corp (Pasadena, Ca) Model 151 current to frequency converter.
Optical transfer	Plasma emission focused by a 16 cm focal length X 5 cm diameter planoconvex, fused quartz lens. Positioned at twice the focal length from the entrance slit and plasma center.
Photomultiplier	EMI 6256, spectral response S-13.

Table I (continued)

RF Generator	Plasma-Therm Inc. (Kresson, NJ) Model MN 2500 E generator, 27 MHz, 2500 W rating, operated at 1500 W forward power, <3 W reflected power.
Nebulizer	1.4 MHz ultrasonic nebulizer with desolvation as described by Olson, et al. (11).
Plasma Torch	All of fused quartz construction as described previously (12). Torch positioned in center of load coil with the top of intermediate tube approximately 3 mm below the bottom coil.
Ar flow rates	plasma gas 16 L/min auxiliary gas 1 L/min aerosol gas 1 L/min
Observation Height	Torch positioned so that optical axis of monochromator corresponded to a height of 15 mm above the top of the load coil. A 4 mm vertical aperture limited the observation zone to the region between approximately 13 and 17 mm above the load coil.
Slits	entrance 20 um intermediate 100 um exit 20 um
Solution uptake rate	2.0 mL/min
Integration period	1.0 sec/data point

video I/O terminal. A source listing of the software routines can be found in Appendix 1.

Versatility was an important objective in the development of the control software. The main features of the control logic are summarized as follows. For all modes of operation, the operator specifies the measurement period (0.1-999 s), and either manual or automatic ranging of the current to frequency (C/F) converter to give output measurements from 10^{-12} to 10^{-6} A photocurrent. In the autorange mode, the computer monitors the photocurrent just prior to the measurement period and switches the C/F converter to the appropriate range. Intensity measurements are read out directly as photocurrents (Amperes).

<u>Wavelength scanning system</u> In the monochromator described herein, an optical incremental encoder is connected to the drive mechanism and enables the computer, through the scan controller, to monitor the wavelength. A flowchart for this procedure is given in Figure 3. The encoder encodes the angular motion of the lead screw. To shift to a preselected wavelength, the computer loads a Binary Coded Decimal (BCD) value equivalent to the preselected wavelength into memory and the drive motor is actuated. As the motor drives the sine bar mechanism, which in turn rotates the grating toward the specified wavelength, the pulses generated by the encoder are fed back to the scan controller. By monitoring the BCD output

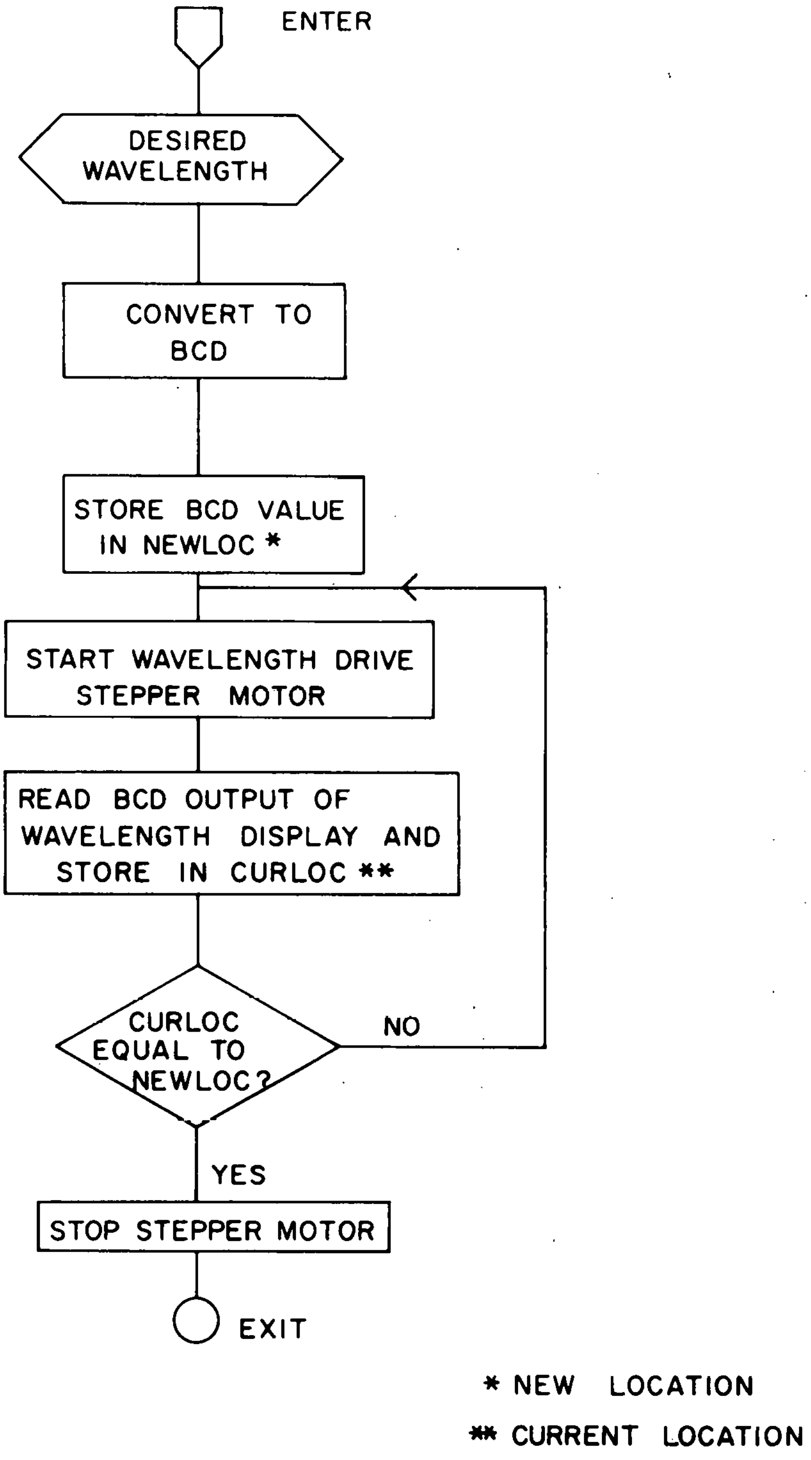

Figure 3. Program flowchart for computer monitoring of the monochromator wavelength

of the scan controller and comparing it with the BCD value stored in memory, the computer is able to determine when the desired wavelength has been reached. Two scanning speeds, fast (500 nm/min) and slow (10 nm/min), are employed so that the wavelength selection is both rapid and precise (without overshoot). For the fast scanning speed, one step of the stepper motor is equivalent to a wavelength change of 0.005 nm. One step of the stepper motor in the slow speed is equivalent to a wavelength change of 0.0001 nm.

Analytical wavelengths The selection of analytical wavelengths is simplified by the availability from a "masterfile" on disk of the four most prominent spectral lines (13) of each of 70 elements. These "true" wavelengths are listed to 4 decimal places as shown in the available wavelength tables (14-18). The wavelengths for each element are assigned "priority" levels based on detection limits and potential spectral line interferences. For each analytical wavelength contained in the "masterfile", a second wavelength, close to the analytical wavelength, is selected for measurement of the spectral background and is also stored in the "masterfile". To begin a series of sequential multielement analyses, the operator specifies the number of lines of each element whose net relative intensity is to be measured. Chemical symbols for the desired elements are entered into the computer through the video terminal. The computer searches the "masterfile" and locates the appropriate analytical wavelengths on a "priority" basis. If an analytical problem requires a special set of spectral lines not available from the "masterfile" ranking, these special wavelengths (with appropriate corresponding background wavelengths) can be entered through the video terminal. The computer then rearranges the analysis wavelengths and the associated "background" wavelengths in order of increasing wavelength. The element symbols, analytical wavelengths, and "background" wavelengths are then stored in a user-named file on disk. Because the information is stored on disk and not in memory, the number of lines that can be measured sequentially is limited only by the number of lines in the "masterfile".

Wavelength calibration Small errors may be introduced into the grating wavelength drive by such factors as thermal drift and through changes in gears for the rapid and slow scan modes. For exchange of the latter, precise stepper motor step sizes may not lead to precisely the same wavelength step sizes. These and other factors may cause a wavelength shift that is not reflected by the encoder, hence the "true" and "instrumental" wavelengths may not agree. This situation is rectified by a "peak search" routine that determines the "instrumental" wavelength of each spectral line. After the appropriate wavelengths are chosen from the "masterfile", the operator runs a reference solution that contains the elements of interest. The "peak search" routine steps across a 0.3 nm wavelength region of the spectrum centered at the analytical wavelength. Intensity measurements are taken every 0.01 nm for a total of 31 intensity measurements. The flowchart for this routine is shown in Figure 4. The criterion used for determining if a peak has been located requires at least five or more consecutive points with increasing intensity followed by five consecutive points with decreasing intensity. If a peak is sensed, the computer then calculates the "instrumental" wavelength of the analysis line and updates the operator's disk file. The "instrumental" wavelengths are used for all subsequent operations.

Maximum intensity routine The "instrumental" wavelength calibration procedure just described is not sufficient to guarantee that subsequent intensity measurements at the "instrumental" wavelengths correspond exactly to the peak intensity of the analysis line. This is because residual thermal and mechanical drift, spurious electronic signals taken as encoder pulses, and other unidentified aberrations may shift the "instrumental" wavelength off peak. A "maximum intensity" routine solves this problem. The computer flowchart for this routine is shown in Figure 5. The major assumptions made with this routine are that the shape of the spectral line is Gaussian in form and that the maximum error in the "instrumental" wavelengths of the analysis lines is

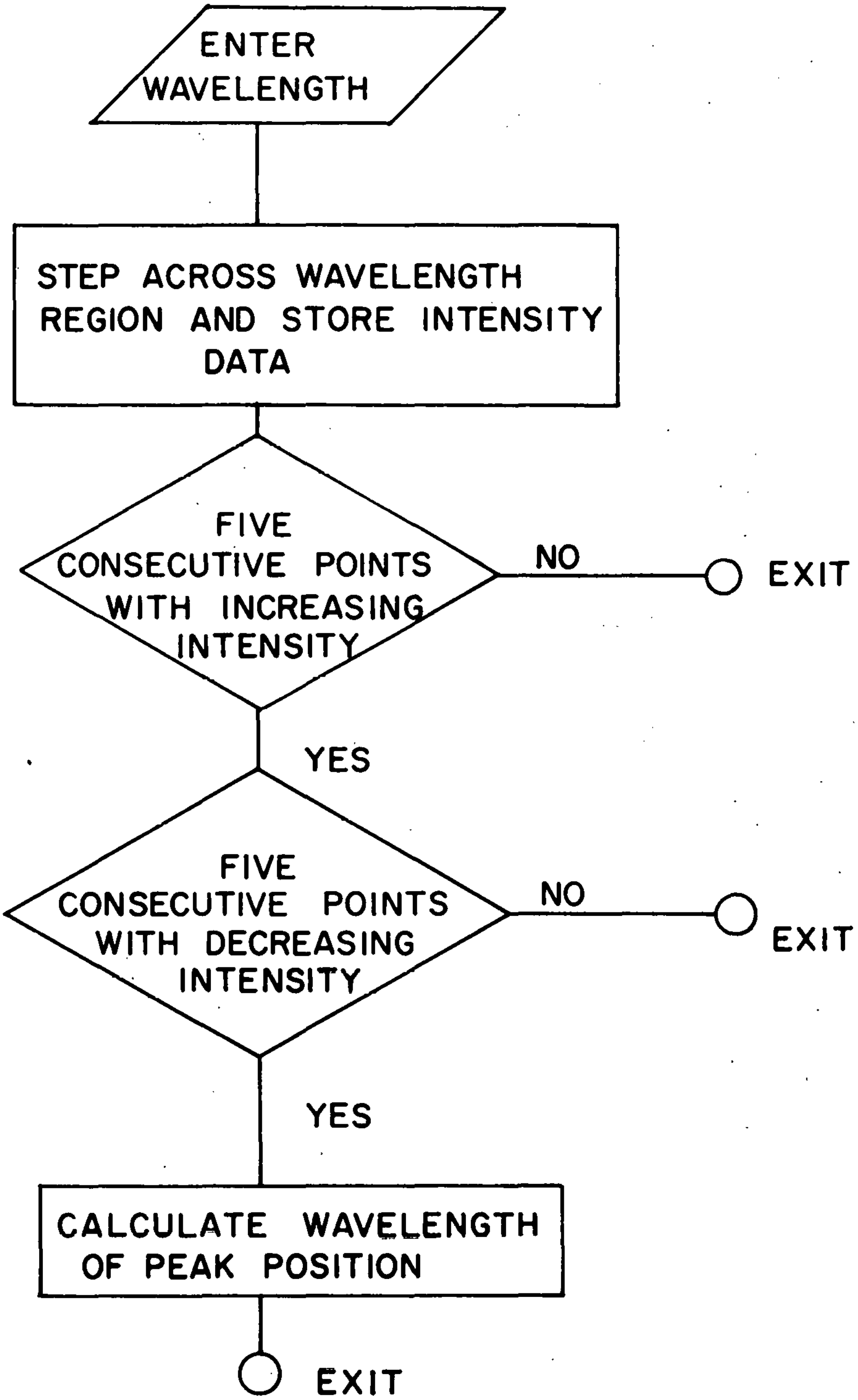

Figure 4. Program flowchart for the "peak search" subroutine

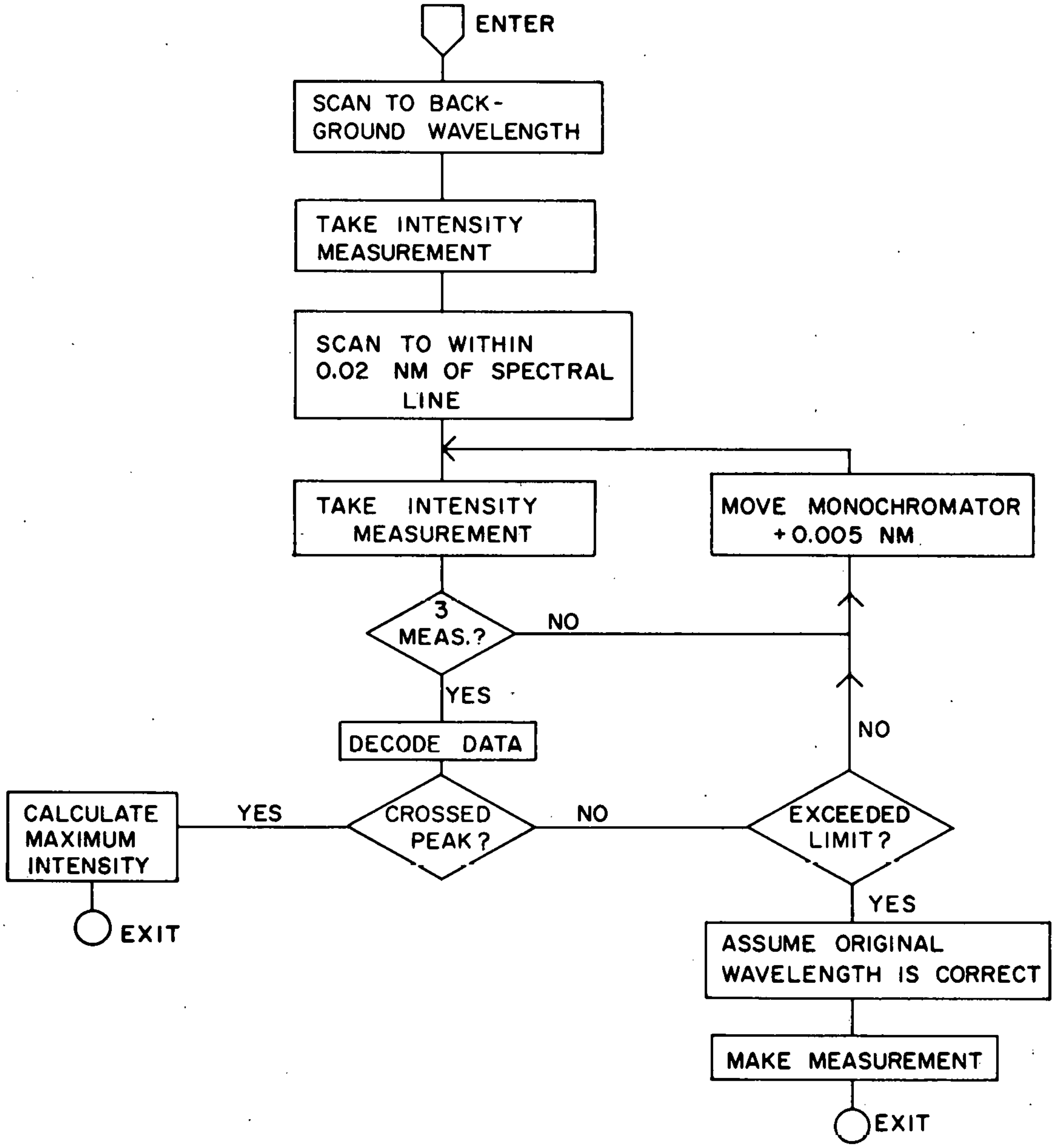

Figure 5. Program flowchart for the "maximum intensity" subroutine

less than 0.02 nm. In operation, the monochromator is scanned to within 0.02 nm of the analytical wavelength. Incremental steps of 0.005 nm are then taken until the computer determines that the maximum of the peak has been crossed. If a Gaussian form is assumed for the spectral line, the maximum intensity may be calculated very precisely, even if the maximum occurs between two wavelength steps. When no peak is discernible, as for a blank solution, the routine assumes that the previously determined "instrumental" wavelength is correct.

This approach to peak intensity measurement differs from those previously described. Spillman and Malmstadt (6) utilized a rotatable quartz refractor plate to correct any errors introduced by the mechanical wavelength drive. Kawaguchi, et al. (9) employed a vibrating quartz plate to optically scan a small spectral region. The peak intensity was then found by their computer program. Roldan (19) and Spillman and Malmstadt (6) have outlined some of the errors that may be introduced when a quartz window is used as a refractor plate.

As the computer starts the system for a multielement mode of operation, the monochromator is first initialized at the Carbon 247.856 nm spectral line. This line is used for reference purposes because it is easily observable in the spectrum of Ar supported ICP discharges and occurs in a region possessing little background structure. The monochromator is then scanned to a starting wavelength which is a few nanometers less than the lowest wavelength of the lines to be measured. Thus all data acquisition is performed with the monochromator scanning in the direction of increasing wavelengths. This procedure avoids errors caused by backlash of the wavelength drive mechanism. When the monochromator is scanning to higher wavelengths, the computer checks for the condition where the monochromator is 0.5 nm from the desired wavelength. When this boundary is crossed the computer reduces the scanning speed from fast to slow and finally halts the stepper motor when the wavelength has been reached.

Main software subroutines

A description of the various software subroutines follows:

INITIATE The INITIATE subroutine initializes the C/F converter measurement period, informs the computer of the current wavelength location of the monochromator, and sets up the input/output devices to be used. For all modes of operation, the operator has the option of storing the answers to the computer's questions in a disk file, or answering the questions via the video terminal. If the former is chosen, INITIATE will assign an operator specified file to be an input file and the computer will now operate without any interaction from the operator.

DETECTION LIMIT The DETECTION LIMIT subroutine is used to determine, from experimental data, detection limits of operator specified analytical lines. The detection limit is defined as the concentration of analyte required to give a net signal equal to three times the standard deviation of the background. The average and standard deviation of the background are obtained at the specified wavelength adjacent to each analyte line while a multielement reference solution is nebulized. The average and standard deviation of the gross signal are obtained at the analyte wavelength. The background and gross signal measurements are made in the order of ascending wavelength regardless of their relative positions. Detection limits determined in this manner provide a sensitive test of the analytical performance of the system.

TAKE DATA The TAKE DATA subroutine obtains data at a single wavelength in a repeat data mode. In this mode the operator can choose the number of intensity measurements to be made in a repetitive sequence. At the end of the measuring sequence the average and standard deviation of the individual measurements are printed. The sequence may be repeated under the initial conditions with a single key command to the computer. Precision values for sample, blank, and dark current measurements are quickly and easily obtained in this TAKE

DATA mode. Also, in this mode the operator has the option of printing the individual measurements of a sequence.

PROFILE The PROFILE subroutine provides profiles of emission intensity versus wavelength on a sequential multielement basis. Each profile consists of a small wavelength region which encompasses one of the analytical lines of interest. To obtain a profile the computer directs the monochromator to step across a wavelength region, taking data at each step. When standard computer plotting techniques are employed, these data can be readily plotted on the video terminal for operator inspection.

CALIBRATE The CALIBRATE subroutine allows an operator to construct analytical calibration curves using intensity versus concentration data collected from reference solutions. A disk file is prepared that contains the elements of interest and the concentrations of the individual analytes in the various reference solutions. The subroutine conversationally leads the operator through the data acquisition procedure for his reference solutions. The individual reference solutions may be run in any order, may be repeated, or may never be run at all. The integration time is specified by the operator, and a table of background corrected, net relative intensities can be requested during calibration so the operator can monitor the performance of the instrument. The net intensities, used in the calibration and analytical routines, correspond to the difference between the calculated maximum intensities at each analytical wavelength and the respective background intensities obtained at the preselected wavelengths above or below each analytical line. This approach for background correction differs from that used by previous authors. Kawaguchi, et al. (9), employed a vibrating quartz plate to produce a sinusoidal wavelength modulation. With sinusoidal modulation, the total integration time of the individual background and analyte line signals is small compared to the total measurement time. The procedure employed here allows for more efficient measurement because the monochromator can be set at a specific wavelength and halted for signal integration. Spillman and Malmstadt (6) also integrated the background signal at a specific point. However, they used a quartz refractor plate to shift the spectrum at the exit slit to the desired background wavelength.

After the operator has calibrated the instrument, the subroutine calculates curve coefficients and prints a value for each analytical line that indicates the quality of fit of the analytical curves to the experimental points. This quality of fit is expressed as the percent root mean square of the relative deviations of the calculated values from the observed data. The subroutine stores the element symbol, wavelength, concentrations, intensities, and curve coefficients for each analysis line in a user-named file on disk. This information forms a complete record of the entire calibration procedure for examination at a later time, should questions concerning the analytical results occur. The analytical calibration curves are calculated as first, second, or third order polynomial equations, as specified by the operator. The operator also has the option of weighting or not weighting the data by the reciprocal of net intensity squared (20).

ANALYTICAL The ANALYTICAL subroutine performs sequential multielemental quantitative analysis of samples using analytical calibration curves prepared by CALIBRATE. Upon entering the ANALYTICAL subroutine, the operator specifies a calibration file containing the analytical calibration curves prepared by CALIBRATE. This subroutine accepts input of sample identification information for the individual samples. If the operator desires to use an autosampler, sample identifications can be set up in a file prior to the analyses. In that case, the analyses may be performed without operator attention. As each sample is run, the computer steps the monochromator to the appropriate analysis line and background wavelengths, measures the emission intensities (for a time specified by the operator), and calculates the concentration of each element using the appropriate analytical calibration curves. The analytical results are printed and stored on disk immediately after each sample is analyzed.

CHAPTER III. INSTRUMENT PERFORMANCE

General Results

An important requirement for computer controlled monochromators is the capability of selecting the predetermined wavelength rapidly and accurately. The reliability of wavelength selection was tested with the "peak search" subroutine, which determines the "instrumental" wavelengths of the analytical lines. Ten measurements, one at the beginning and end of each day, were obtained for 16 wavelengths over a period of five days. The mean and standard deviation of these measurements, summarized in Table II, indicates that the monochromator is relatively free from mechanical drift.

The slewing time of the monochromator is given by the equations

$$t=0.120(x)+3.2 \qquad \text{for } x>0.5 \text{ nm}$$

$$t=6(x)+0.1 \qquad \text{for } x<0.5 \text{ nm}$$

where t is the time in seconds required for slewing x nanometers between wavelengths A and B. For example, the monochromator will slew from Zn 213.86 nm to Ba 455.40 nm in 32.17 seconds. Movement of the monochromator from 213.76 nm (background wavelength for Zn 213.86 nm) to 213.86 nm requires 0.70 seconds.

Table II. Precision of Wavelength Reproducibility

ELEMENT	"TRUE" WAVELENGTH (nm)	MEAN[a] "INSTRUMENTAL" WAVELENGTH (nm)	STANDARD DEVIATION (nm)
Mo	202.03	202.03	0.007
P	213.62	213.63	0.007
Zn	213.86	213.86	0.008
Cd	214.44	214.45	0.007
Ni	231.60	231.61	0.007
Ir	224.27	224.28	0.007
Co	228.62	228.62	0.008
Rh	233.48	233.48	0.007
B	249.77	249.79	0.003
Mn	257.61	257.62	0.004
Cr	267.72	267.73	0.008
V	310.23	310.27	0.010
Be	313.04	313.06	0.006
Nb	316.34	316.36	0.010
Cu	324.75	324.78	0.008
Ba	455.40	455.37	0.010

[a] Mean and standard deviation of 10 individual measurements over a five day period.

Maximum intensity routine

The operation of the "maximum intensity" subroutine was tested by a comparison study made of the precision of maximum intensity measurements that could be achieved, with and without using this subroutine, at ten different wavelengths ranging from Zn 213.86 nm to Cu 324.75 nm. First, the "instrumental" wavelengths of the ten lines were determined with the "peak search" routine. Then the instrument was slew-scanned to each of these ten lines, in turn, ten different times. Intensity measurements were made for each line for each scan. Next, the same ten line scan sequence was repeated again ten times but with application of the "maximum intensity" routine (as described in Chapter II). The results of these measurements, summarized in Table III, show that an improvement in the precision of the intensity measurements was generally achieved when the "maximum intensity" subroutine was employed. The abnormally high values observed for the Cd and Cu lines when the "maximum intensity" routine was not applied are representative of the loss of precision introduced by spurious off-peak intensity measurements.

When analyte concentrations are determined near the detection limit concentrations, the spectral background will normally comprise a large fraction of the total signal. Because quantitative determinations at detection limit concentrations are not feasible, it is useful to estimate the lowest quantitatively determinable concentration (LQD). If the reasonable assumption is made that a signal ten to fifteen times greater than the standard deviation of the background scatter (noise) is desirable for quantitative determination, then a concentration five times greater than the experimentally determined detection limits may be considered as a reliable estimate of the LQD. It is then of interest to inquire how reliably the "maximum intensity" routine can sense that a peak exists at LQD concentration levels and how

Table III. Precision of "Maximum Intensity" Routine

Set	"Instrumental" Wavelength, nm	Monochromator scan without maximum intensity subroutine	Monochromator scan with maximum intensity subroutine
A	Zn 213.86	5.87	2.96
	Cd 214.45	12.23	4.40
	Ni 231.61	6.60	5.00
	Co 228.62	1.42	2.62
	Fe 238.20	1.38	0.76
	Mn 257.62	1.29	0.64
	Cr 267.73	6.41	3.51
	V 310.27	5.59	2.15
	Cu 324.78	15.37	6.45
B	Zn 213.86	3.74	1.99
	Cd 214.45	6.21	3.23
	Ni 231.61	3.21	3.01
	Co 228.62	2.32	1.43
	Fe 238.20	1.98	0.83
	Mn 257.62	1.20	0.50
	Cr 267.73	2.36	2.04
	V 310.27	3.01	2.03
	Cu 324.78	5.43	3.40

A. %RSD of 10 points at analyte concentrations 10 times the detection limit.

B. %RSD of 10 points at analyte concentrations approximately 50 times the detection limit.

accurately the net intensities may be measured under these conditions. To provide an answer to these questions, the following experiment was designed and conducted. First, the assumption was made that the most reliable measurement of net relative intensities are made by background stripping from a complete wavelength scan profile, obtained in the manner described in Chapter II. Second, the net relative intensities were measured and calculated via the "maximum intensity" routine and the value so obtained was compared to the wavelength profile measurements. These measurements were performed at approximately LQD levels and for multiples of these levels. To provide the reader with some perspectives, Figure 6 shows typical wavelength profiles of the lines and background on which the above measurements are based. The data summarized in Table IV show that acceptable accuracy of the "maximum intensity" routine is maintained, even as the LQD values are approached.

Table IV. Comparison of Observed and Calculated Maximum Intensities

			PHOTOCURRENTS		
WAVE-LENGTH (nm)	APPROX NOMINAL CONC	CONC (ug/mL)	OBSERVED FROM WAVELENGTH PROFILES	CALCULATED BY "MAX INT" ROUTINE	%ERROR
Mn 257.6	LQD	0.0005	5.34×10^{-9}	5.21×10^{-9}	2.43
	5xLQD	0.0020	5.93×10^{-9}	5.88×10^{-9}	0.84
	10xLQD	0.0040	6.72×10^{-9}	6.80×10^{-9}	1.19
	100xLQD	0.0625	2.98×10^{-8}	2.96×10^{-8}	0.68
Y 371.0	LQD	0.0005	1.57×10^{-8}	1.52×10^{-8}	3.18
	5xLQD	0.0040	2.03×10^{-8}	2.07×10^{-8}	1.97
	10xLQD	0.0080	2.80×10^{-8}	2.75×10^{-8}	1.79
	100xLQD	0.0625	1.05×10^{-7}	1.06×10^{-7}	0.95
Ba 455.4	LQD	0.0020	1.43×10^{-8}	1.46×10^{-8}	2.10
	5xLQD	0.0080	2.63×10^{-8}	2.59×10^{-8}	1.52
	10xLQD	0.0160	4.14×10^{-8}	4.10×10^{-8}	0.97
	100xLQD	0.2500	5.20×10^{-7}	5.25×10^{-7}	0.57

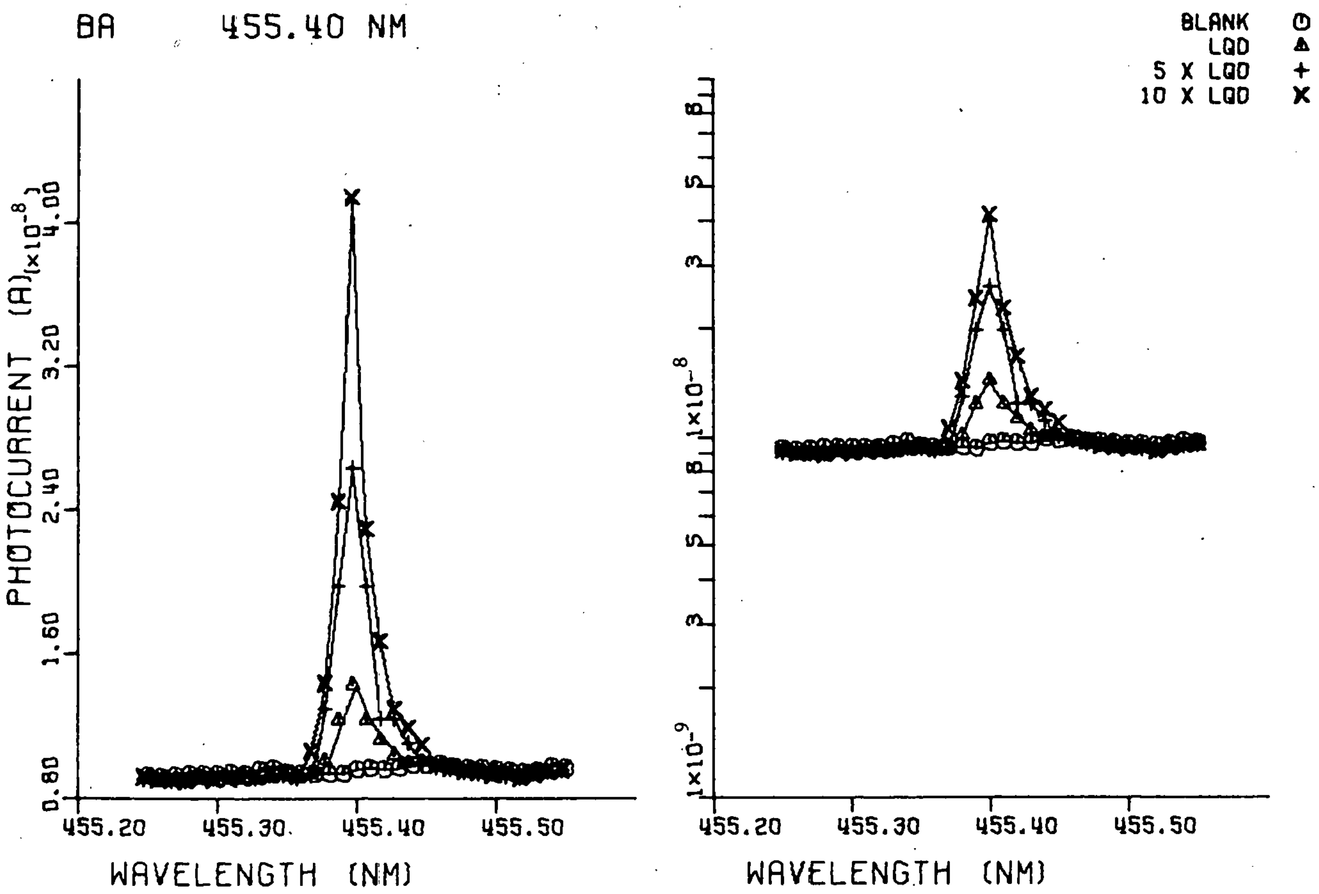

Figure 6. Wavelength profile of emission intensity vs. wavelength for the Ba 455.4 nm analytical line. Analyte solution concentrations are approximately LQD levels (see text)

Stability

Table V shows typical short term stabilities observed for a reference sample solution that was run 5 times at approximately equal time intervals throughout a normal analytical sequence. During this sequence 20 rock and flyash samples were analyzed. The reference sample was one of a series of standards used in the initial calibration. The total elapsed time from the beginning of the calibration to completion of the sample analyses was about eight hours. No adjustment of operating conditions was made during this period. The average and standard deviation for most of the elements in Table V indicate adequate stability for trace, minor, and even for many major constituent determinations.

Time comparisons

The gain in flexibility provided by the scanning monochromator approach is obviously achieved at the expense of the time required to perform multielement determinations. To draw comparisons on the time requirements between simultaneous and rapid sequential determinations, it is useful to divide the analytical operations into elapsed time requirements for preliminary operations such as instrument alignment, performance evaluation, and analytical calibrations, and for the actual sample analyses. These elapsed time estimates are summarized in Table VI for the determination of 30 elements in a sample. In making these estimates, we have assumed that all methods development phases have been completed, i.e., the wavelengths of the analyte lines and the associated background correction wavelengths are "programmed" for the polychromator and the scanning monochromator, and that background corrections are made for each analyte line. Integration periods of one second were assumed for each of the approximately five data points required for applying the "maximum intensity" routine and for background corrections for the monochromator. In the polychromator, integration periods of ten seconds were assumed for the line plus background and background measurements, performed by the programmable scanning entrance slit method (21). The time estimates in Table VI reveal that 30 elements are determinable in 30 samples during an elapsed time of two hours and six hours, respectively, for the polychromator and sequential monochromator systems. These time estimates reaffirm a well known fact, namely, that polychromators possess obvious advantages for the routine simultaneous determination of the same set of elements in large numbers of samples of similar composition. However, as stated in the introduction, scanning monochromators offer attractive features for the determination of a far broader range of elements in samples of widely varying composition.

Table V. Reproducibility of Analytical Data

ELEMENT	REFERENCE STANDARD (ng/mL)	MEASURED AVERAGE[a] (ng/mL)	RELATIVE STANDARD DEVIATION (%)
As	100	94	5.27
Ba	100	99	1.40
Be	100	101	1.20
Bi	100	96	11.30
Cd	100	97	4.00
Co	100	99	2.10
Cr	100	99	1.80
Cu	100	96	2.60
Eu	100	101	4.60
Ga	100	104	8.40
Ir	100	95	3.60
La	100	97	2.10
Mn	100	100	1.00
Mo	100	93	8.10
Ni	100	97	0.90
Sc	100	99	0.60
Sr	100	100	1.10
Y	100	98	2.30
Yb	100	101	1.90
Zn	100	98	2.10
Zr	100	102	4.20

[a] Reference solution was run five times at regular intervals during an 8 hour normal analytical sequence.

Table VI. Estimated Elapsed Times for Determining 30 Elements in a Sample

	Time Estimates (min)	
	Scanning Monochromator	Polychromator
Preliminary Operations		
Wavelength Alignment	10	2
Performance Monitoring	15	8
Analytical Calibrations (for five reference solutions)	60	20
Total	85	25
Analytical Operations (per sample)	12	4
Analytical Operations (30 samples)	360	120

CHAPTER IV. ANALYTICAL APPLICATIONS

Introduction

There are many technological fields in which knowledge of elemental composition of inorganic materials at the major, minor, trace, or ultratrace level is often necessary. These fields include geochemistry, geochemical prospecting, ore processing, soil nutrition, manufacturing of inorganic materials (e.g., glass, pigments), and environmental pollution (e.g., flyash, sediments, and air particulates).

Analytical techniques in common use

In years past, elemental determinations at the major, minor, and sometimes at the trace level in the sample matrices of the types listed above have been performed by classical gravimetric, titrimetric, and spectrophotometric techniques (22). During the past several decades the classical approaches have been largely replaced by a variety of instrumental techniques, which include X-Ray fluorescence, atomic absorption spectrometry, neutron activation analysis, and proton induced X-ray emission analysis.

Neutron activation analysis (NAA) is well-suited to the analysis of environmental and geochemical samples for a number of reasons (23-24). The detection limits obtainable range from parts per million down to 10^4 molecules, the chemical nature of the radioactive material can be controlled, and the method can become uniquely specific for the analytes. Furthermore, NAA is relatively free from interferences. While NAA satisfies the need for multielement determinations, it unfortunately involves sophisticated theory and application, adverse public reaction, use of proper tracers, and most of all, expensive reactor facilities. Long irradiation times and chemical separations that are usually required result in a high cost per analysis and low thruput of samples.

X-ray fluorescence (XRF) has the advantage over other analytical methods in that for certain elements of geochemical interest, for example, Th, Nb, Ta, and W, it is far superior in most cases (25-29). The disadvantages of XRF are the complications owing to interelement effects, especially when large amounts of sample are collected, and the difficulty involved in preparing proper standards (30). Also, most efficient X-ray emission is obtained by use of an X-ray tube with a target metal only a few atomic numbers higher than the analyte element. With a single X-ray tube high powers of detection for heavy elements requires sacrifice in detectability for light elements and vice-versa (31).

Proton induced X-ray emission analysis (PIXEA) is a newer technique which can determine all the elements from sulfur through uranium quantitatively in one analysis without sample preparation at levels as low as 1 ug/g, depending on the particular metal and sample (32-33). PIXEA is advantageous to use because of its high absolute sensitivity, it is economical for multielement analysis, and has rapid turnaround for obtaining results. Unfortunately, PIXEA requires access to a suitably equipped proton accelerator, volatile components in the sample may be lost because of placement in a vacuum, and interelement interferences will affect sensitivity limits (32).

Atomic absorption spectroscopy (AAS) is the most widely used technique in the analysis of the sample types discussed above (34-40). Although AAS has excellent selectivity and sensitivity, it is basically a single element at a time technique. Also, the methods of analysis by AAS require large amounts of time and can lead to erroneous results if careful matrix matching is not practiced (34). The elapsed time to complete the determination of an extended list of trace metals in a large number of samples is therefore much greater and unacceptable under some circumstances.

Inductively coupled plasma-atomic emission spectroscopy

The inductively coupled plasma-atomic emission excitation source (ICP) possesses certain unique properties and operating characteristics not found in spark or flame atomization systems (12, 41). Some of these properties and characteristics are discussed below, but first an examination into the nature of the ICP may be helpful. A plasma is defined as a gas in which a significant fraction of the atoms or molecules are ionized. In the case of the ICP, the plasma is formed and sustained at the open end of an assembly of quartz tubes known as a plasma torch through which argon flows, as shown in the schematic diagram of the ICP in Figure 7. An alternating current (I) generated by a high frequency current generator flows through the coupling coil which surrounds the open end of the plasma torch. The current, alternating at approximately 27 MHz in the system in this laboratory, generates a time-varying magnetic field (H) with force lines as shown in the figure.

Plasma initiation and thermal isolation To initiate the plasma, the argon in the plasma support and auxiliary argon streams is partially ionized by a tesla discharge. The resulting charged particles (electrons and ions) interact with the magnetic field; they are induced to flow in closed circular paths. The electron and ion flow, or eddy current as it is called, is analogous to a short circuited secondary of a transformer. Since the current in the coil and, therefore, the magnetic field are varying sinusoidally in direction and strength with time, the ions and electrons are accelerated every 1/2 cycle. In this acceleration, they meet resistance to flow which results in Joule heating, which in turn leads

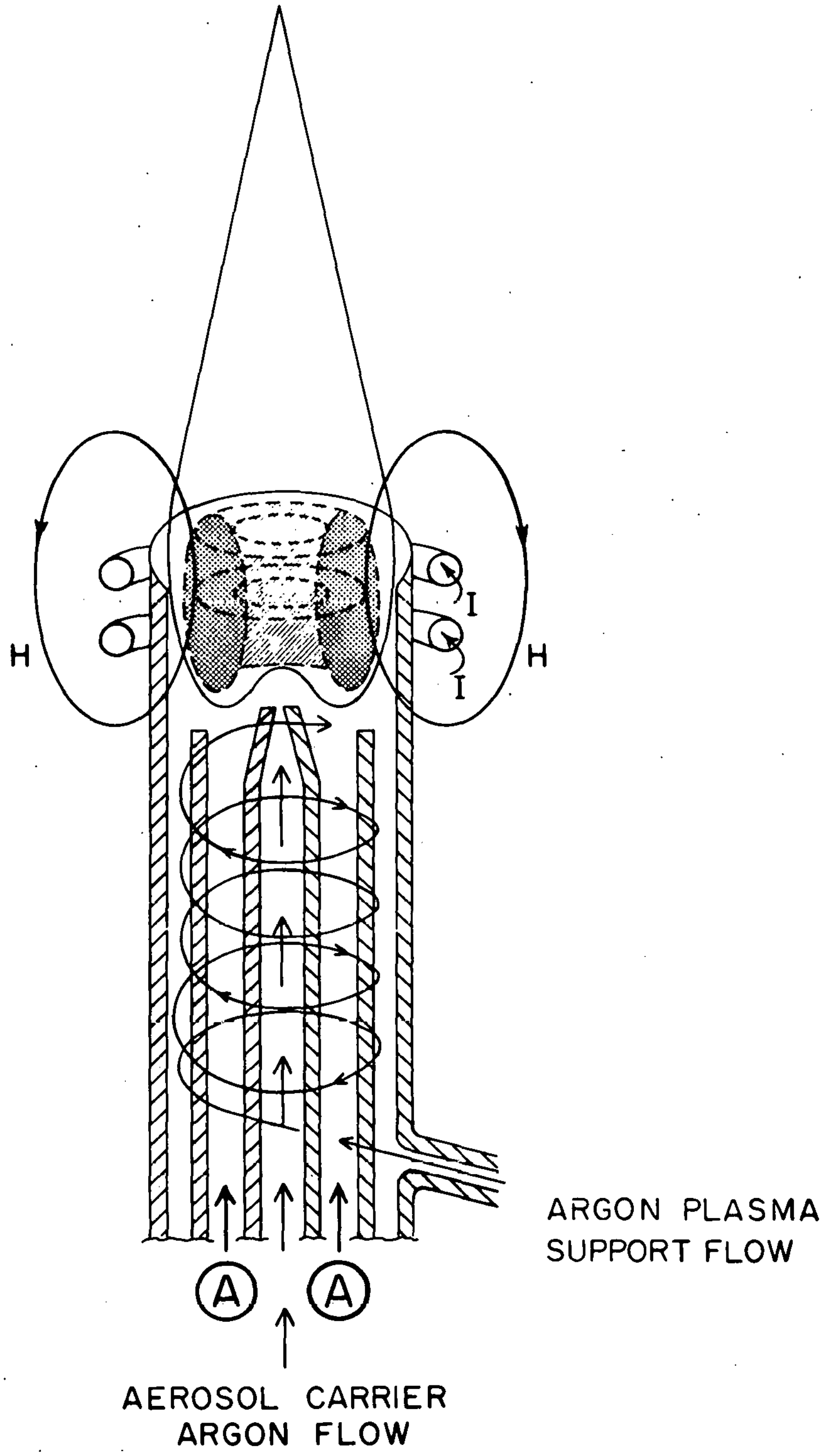

Figure 7. A schematic diagram of an inductively coupled plasma

to additional ionization. In this way, a plasma of extended dimensions is formed almost instantaneously.

The plasma is isolated thermally from the plasma torch by the vortex stabilization technique suggested by Reed (42-43), for which the support argon flow is introduced tangentially as shown in Figure 7. This tangential flow also serves to center the plasma in the torch.

Sample introduction   Once the plasma has been initiated, an argon flow of approximately 1 L/min is introduced in the central tube as shown in Figure 7. This argon flow carries the sample aerosol. In order for the ICP to act as an effective atomization and excitation source, the sample aerosol must be injected efficiently into the plasma and remain in the high temperature environment as long as possible. As Fassel has aptly noted (44), this physical situation is difficult, if not impossible, to achieve in many other plasmas. One of the main forces acting in opposition to the injection of sample into the ICP is expansion thrust pressure. As gases are heated in the ICP, they expand and are accelerated in a direction perpendicular to the exterior surface of the plasma. This outward movement of gas bombards the incoming particles or aerosol droplets and tends to deflect them around the plasma.

The presence of magnetic pressure in teardrop shaped ICP's which use laminar, coaxial argon introduction was first recognized by Chase (45-46) and later studied by several others (47-51). In an ICP, the electric field and, therefore, the induced current density are zero at the center of the plasma. The magnetic fields induced by the coil and eddy currents reinforce each other at the outside of the discharge and oppose each other at the center. These characteristics lead to the result that the Lorentz force produces magnetic compression around the sides of the plasma directed toward the axis. This compression opposes gas flow from the interior through the sides and confines the plasma in the radial direction. Thus, the pressure can be converted into kinetic energy only in the axial direction and an outward flow is produced along the axis in both the upstream and downstream directions. This "magnetic pumping" which is fed by an inward flow of gas through the sides of the plasma, has been observed experimentally by Waldie (47). The outward flow of gas along the axis opposes entry of particles or aerosol into the plasma and tends to deflect them around the plasma. Chase (46) has noted, however, that if the plasma support gas is introduced tangentially as is done in this laboratory, there is a low pressure zone in the axial channel of the plasma, as postulated by Reed (42), which obscures the magnetic pressure effect (45).

In addition to tangential gas introduction, the skin depth effect has been used to facilitate sample introduction.

The skin depth, i.e., the depth at which the eddy current is reduced to 1/e of its surface value, is inversely proportional to the square root of the current frequency. As the frequency is increased, the position of highest eddy current density moves toward the outer surface of the plasma. At the frequencies used in this work, an annular plasma is formed as shown in Figure 7. Because the axial region is somewhat cooler than the annulus, it offers less resistance to sample introduction. The annular shape can be further developed by optimizing the gas flows. This plasma configuration permits efficient sample introduction along the axis.

Advantages of the inductively coupled plasma By the time the sample or its decomposition products reach the observation height 15 to 20 mm above the load coil, they have had a residence time in the plasma of approximately 2 ms. During that time, they have experienced temperatures ranging from approximately 5500 K to approximately 7000 K. Both the temperature and the residence time are approximately twice as great as those found in a nitrous oxide-acetylene flame, the hottest flame normally used in analytical spectroscopy. The relative high temperatures and residence time and the inert environment provided by the plasma support and stabilizing gas lead to the expectation that the degree of atomization of the samples, included suspended particulates, should be greater than in flame or spark discharges. Because the atoms are released in a noble gas environment, free atom depopulation processes, such as monoxide formation, should be minimized. These favorable environmental factors should, in turn, overcome many of the interelement or matrix interference effects found in flames or arc or spark discharges. These expectations have been confirmed by observations on several classical interelement interference effects (52).

Other features of the annular ICP arise from the fact that the analyte atoms or ions are confined to a narrow channel along the axis. The analyte emission is concentrated in this region and can be utilized more efficiently by convential spectrometers. The temperature across the channel at the normal observation height is relatively uniform, and the number density of analyte atoms or ions in the surrounding argon is far lower than in the channel (53). Under these conditions, the central channel of the ICP acts as an optically thin emitting source. With the proper detection and measurement system, analytical calibration curves that are linear over five orders of magnitude change in concentration are possible.

Previous work

In recent reviews (1, 2, 5, 54, 55), several authors have suggested that inductively coupled plasma-atomic emission spectroscopy (ICP-AES) offered several advantages as an attractive alternative approach for the analysis of the sample types discussed above. These advantages include: (a) the applicability to virtually all elements; (b) simultaneous rapid determination at major, minor, trace, and ultratrace concentrations levels without changing experimental conditions; (c) unusually low interelement effects; and (d) high powers of detection.

In view of these advantages, it is not surprising that the ICP-AES has been applied, to a rapidly increasing extent, to the analysis of the sample types discussed above (56-74). Most of the applications discussed to date have involved the use of a polychromator for the multielement determination of selected elements of interest in matrices of closely similar composition. When the analyst is faced with the determination of a broader range of elements at various concentration levels in samples of widely varying composition, the fixed array of exit slits employed for isolation of the spectral lines in a polychromator becomes restrictive.

With this background, the present investigation was undertaken. The applicability of the ICP - scanning monochromator system to the determination of major, minor, and trace elements on a sequential multielement basis in a variety of matrices was explored. These matrices include flyash, coals, urban particulates, glass, river sediment, and rocks.

Experimental Procedures

Sample dissolution

0.20 g of sample was accurately weighed into a 15 mL graphite crucible (Vitrecarb, Anaheim, Ca), after which 2 g of NaOH pellets were added to the crucible. The crucible was gently heated with a Fisher blast burner to melt the contents, care being taken to assure that no spattering occurred. After the moisture was evaporated from the NaOH, which usually required two to three minutes, the sample was brought to red heat until a clear melt was obtained. The red hot fused melt was carefully and quickly poured into a 150 mL platinum dish. The upright crucible was also set into the dish. After the sample cooled, 5.0 mL of concentrated HCl was added to both the fused sample and the graphite crucible. Heat was gently applied on a hot plate to dissolve the melt, including the portions adhering to the crucible. The crucible was then thoroughly washed with deionized water and the contents added to the platinum dish. The sample was then diluted to approximately 40 mL with deionized water and heat applied gently until the solution was clear. Samples containing high silicon content were filtered or centrifuged to remove flocculent hydrous silicon oxide. The sample solution was then transferred to a 100 mL polypropylene volumetric flask and diluted to volume with deionized water. The time required for dissolution of a sample was approximately 30 minutes.

Although the ICP is normally operated without the auxiliary plasma argon flow (12) when aqueous solutions are nebulized, this flow is desirable for samples fused with NaOH because there is a tendency for the formation of NaCl deposits on the rims of the two inner tubes of the torch. The auxiliary argon flow provides an improved spatial separation of the plasma and these tubes.

Reagents

Stock solutions were prepared by dissolution of pure metals or reagent grade salts in dilute (1%) acid or deionized water. Reference solutions were prepared by the incremental addition of the stock solutions to an appropriate blank solution. Reference solutions were prepared and stored in Class A volumetric glassware which had been leached for 72 hours with a 10% HCl/deionized water solution. A complete set of reference solutions was prepared that provided multielement calibrations throughout specified analytical ranges. Each reference solution contained the equivalent of 2 g NaOH (same as the dissolved samples) and was confined to only one concentration level of the analytes so that any cross contamination due to impurities in the stock solutions was insignificant. This procedure compares with analytical calibrations where a range of concentration levels have been grouped together in individual reference solutions making contamination corrections necessary (60, 64).

Selection of analysis lines

The analyte lines were selected on the basis of their net and background intensity as well as their freedom from spectral interferences. The line selection process was facilitated by the list of prominent lines recently published by this laboratory and by the associated atlas of wavelength scans, which is now being prepared for publication (13). The wavelengths of the single set of selected analyte lines used throughout the course of our studies are shown in Table VII.

Results and Discussions

Detection limits

When the above procedure for dissolution of the samples is followed, all constituents which occur in these samples are diluted 500-fold. Thus, the quantitative determination of trace elements in these samples requires the ultimate in powers of detection. The overall performance of the ICP with reference to this figure of merit has been adequately documented (1, 11). Detection limits for 55 elements were determined under the operating conditions defined in Table I for

Table VII. Analyte Emission Lines Used For Analysis

ELEMENT	WAVELENGTH (nm)	ELEMENT	WAVELENGTH (nm)
Al	308.22	Nd	430.36
As	197.20	Ni	231.60
Au	242.80	Os	225.59
B	249.77	P	213.62
Ba	455.40	Pb	220.35
Be	313.04	Pd	340.46
Bi	223.06	Pr	390.84
Ca	315.89	Re	221.43
Cd	214.44	Rh	233.48
Ce	413.77	Ru	240.27
Cr	267.72	Sb	217.59
Co	228.62	Sc	361.38
Cu	324.75	Se	196.03
Dy	353.17	Sm	359.26
Er	337.27	Sn	235.48
Eu	420.51	Sr	407.77
Fe	238.20	Ta	226.23
Ga	294.36	Tb	350.92
Gd	342.25	Te	214.28
Ge	265.12	Th	283.73
Hf	277.34	Ti	334.94
Ho	389.10	Tl	351.92
In	230.61	Tm	346.22
Ir	224.27	U	385.96
La	394.91	V	310.23
Li	460.29	W	207.91
Lu	261.54	Y	371.03
Mg	279.55	Yb	328.94
Mn	257.61	Zn	213.86
Mo	202.03	Zr	343.82
Nb	316.34		

solutions containing 2.9% NaCl (same as the dissolved samples). These "solution" detection limits (ug of analyte/mL of sample) were then converted to "solid" detection limits (ug of analyte/g of sample) by multiplying each calculated value by a factor of 500. Detection limits obtained with the computer controlled monochromator are given in Table VIII.

The detection limits reported correspond to the concentration of analyte required to give a net signal equal to three times the standard deviation of the background. This is in accordance with IUPAC recommendations (75). The background intensity was determined by calculating the average of ten 3 second integrations at a preselected wavelength adjacent to each analyte line while a multielement reference solution was nebulized. The standard deviation of the background intensity was also calculated from the ten 3 second integrations. The average and standard deviation of the gross signal were determined in the same manner, except the monochromator was positioned at the analytical wavelength.

Analytical calibrations

Analytical calibration curves were generated by measuring the net relative intensities of selected analysis lines (see Table II) emitted when synthetic reference solutions were run under the conditions defined in Table I. The analytical calibration curves, plotted on a log I vs. log C coordinates, can be observed directly on the video terminal upon completion of the calibration subroutine, thus providing the operator with an immediate visual check of the shape of the curve and the fit of the points. The analytical calibration curves were generally linear over a concentration range of

Table VIII. Detection Limits

ELEMENT	DET. LIM. (ug/g)	ELEMENT	DET. LIM. (ug/g)
Al	1.	Ni	1.
As	5.	Pb	10.
Ba	0.25	Pd	0.5
Be	0.03	Pr	0.1
Bi	2.5	Re	2.
Ca	0.05	Rh	5.
Cd	0.5	Ru	4.
Co	1.	Sb	15.
Cr	0.5	Sc	0.05
Cu	0.2	Se	35.
Dy	0.5	Sm	6.
Er	0.5	Sn	5.
Eu	0.25	Sr	0.01
Fe	0.25	Ta	8.
Ga	3.	Tb	0.5
Gd	2.	Te	2.
Ge	10.	Th	5.
Hf	1.	Ti	0.2
Ho	0.5	Tl	4.
In	10.	Tm	2.
Ir	0.3	U	6.
La	1.	V	0.1
Lu	0.1	W	5.
Mg	0.02	Y	0.04
Mn	0.05	Yb	0.05
Mo	3.	Zn	0.1
Nb	0.5	Zr	0.5
Nd	1.5		

four or more orders of magnitude. However, analytical calibration curves for a few elements (i.e., Ca, Mg, Fe) deviated slightly from linearity at the high concentration range of the calibration, indicating either a departure from linearity of the readout system at high concentrations or deviation from the "optically-thin emission cell" ideal (63). In the present work the calibration data were fitted to third-order orthogonal polynomials which accommodated these slight deviations from linearity. As a result, both the major and trace constituents in a sample can be determined at a single dilution level. In contrast, analytical calibration curves obtained by flame atomic emission and absorption are generally linear over a 1.5 to 2 orders of magnitude change in concentration (76), and thus several sample dilutions are often necessary for the determination of major and trace elements in a sample. Figure 8 shows examples of analytical calibration curves obtained sequentially from 2.9% NaCl reference solutions. The data points for these curves represent net emission intensities (not intensity ratios).

Spectral line interferences

In general, environmental and geochemical samples provide diverse matrices that are exceptionally complex, both chemically and physically (77). The complexity of these samples raises the possibility that interelement effects of various types or spectral interferences may introduce errors in the analytical determinations. Observations on several classical solute vaporization interference systems (52, 64, 78-79) have shown that most interelement interference ettects are reduced to negligible proportions in the ICP. However, at analyte concentrations in the ug/mL to ng/mL concentration

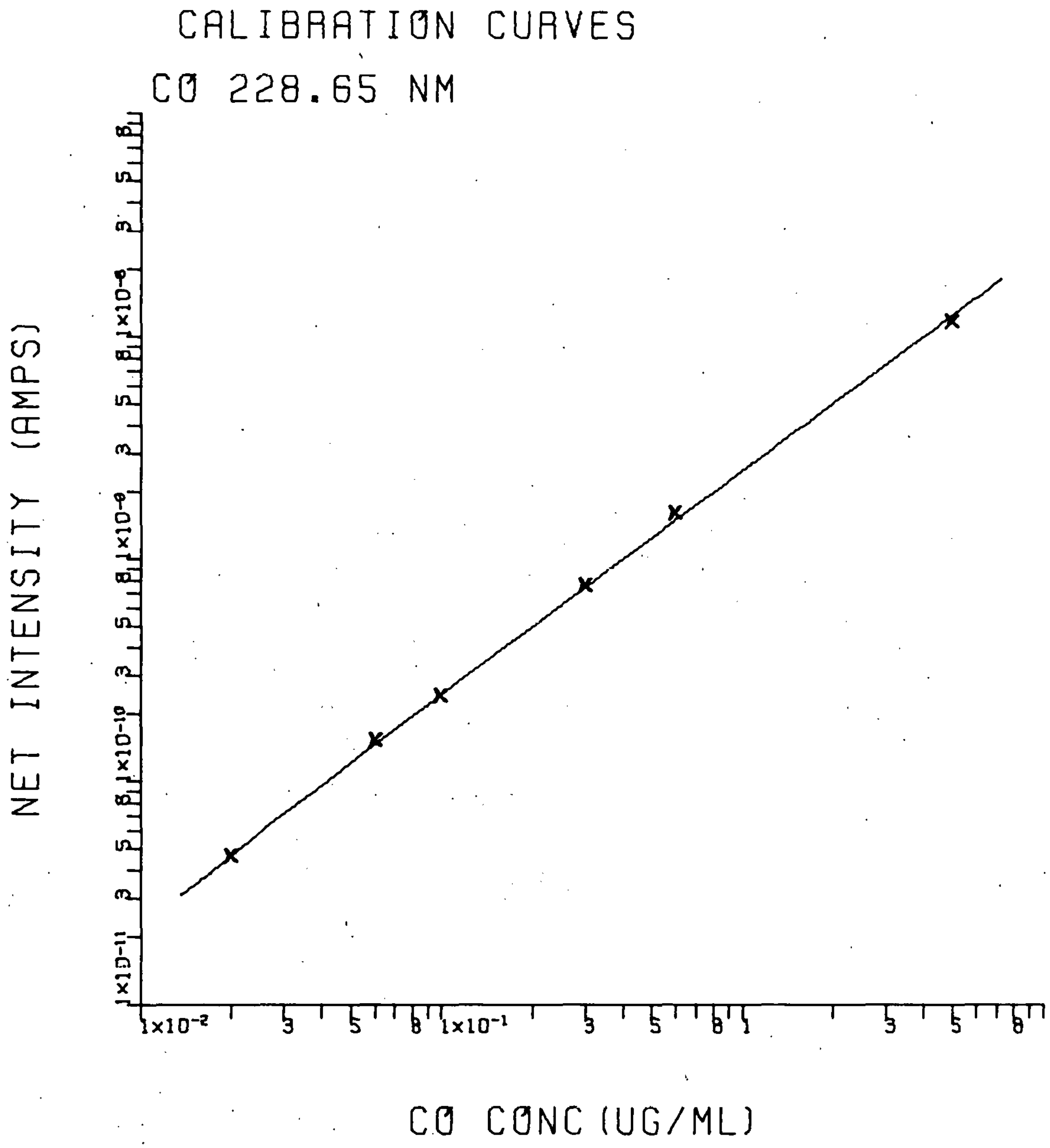

Figure 8. Logarithmic calibration curves of intensity vs. concentration obtained with the computer controlled monochromator. Data points represent net emission intensities

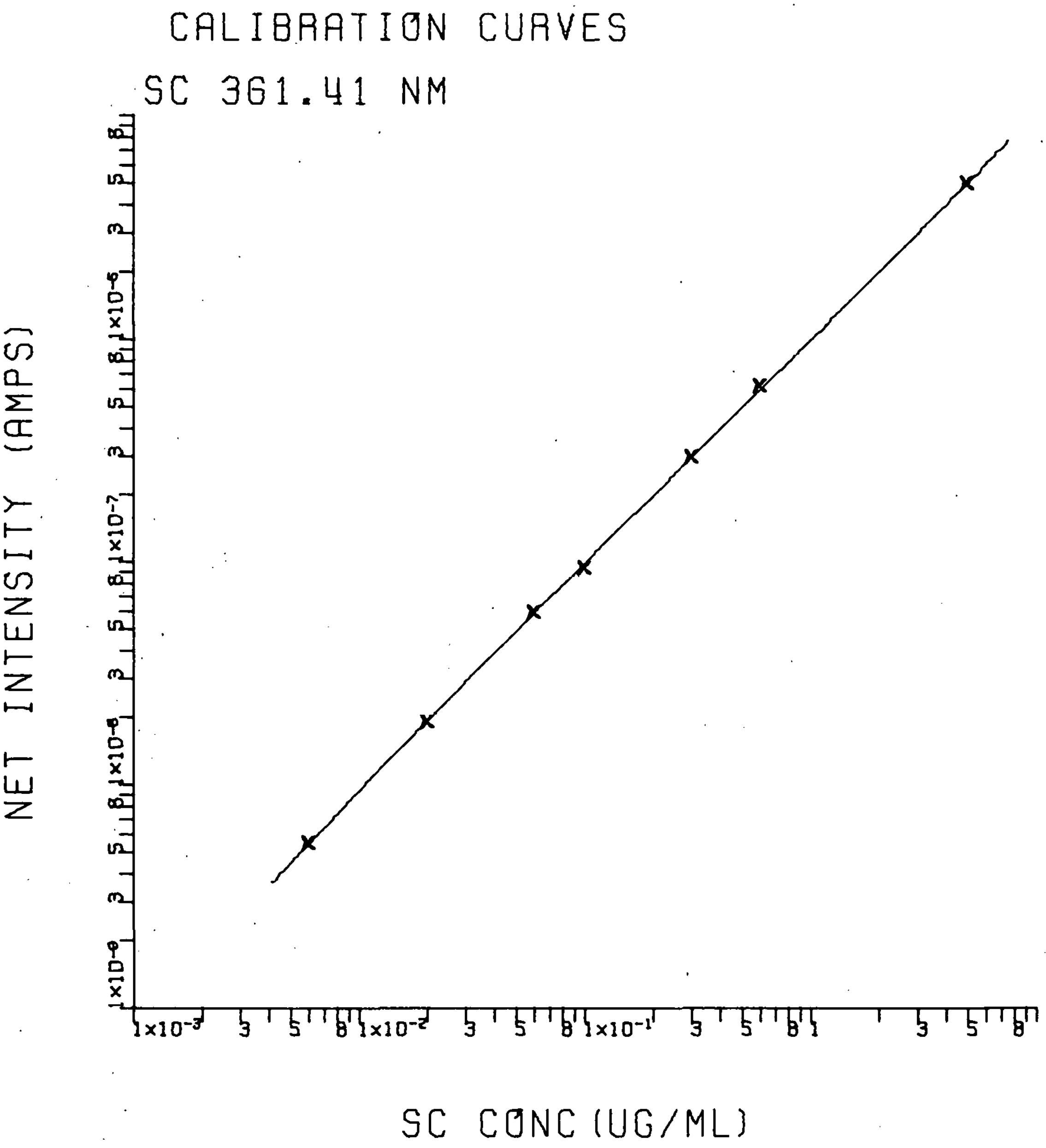

Figure 8. (continued)

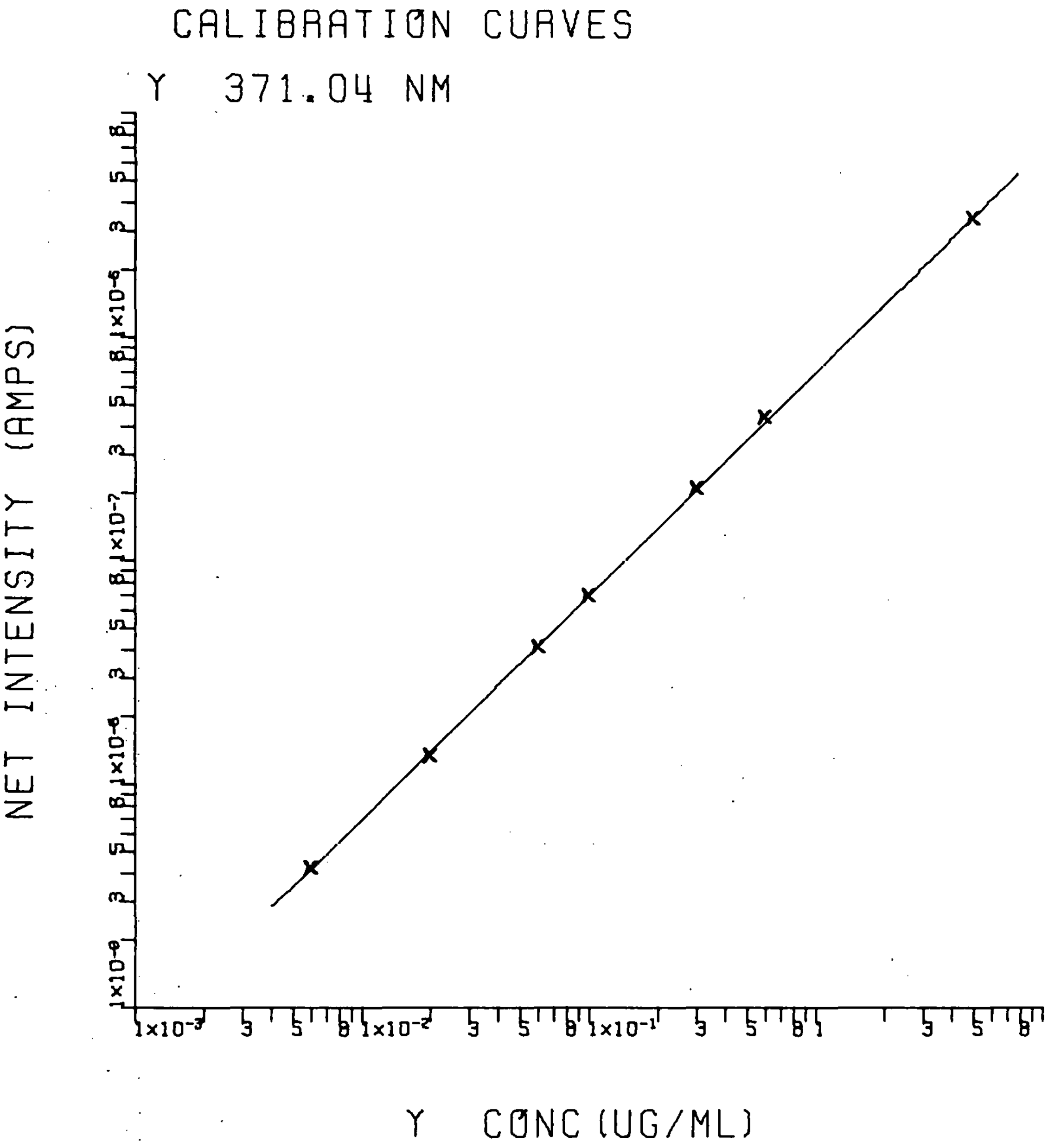

Figure 8. (continued)

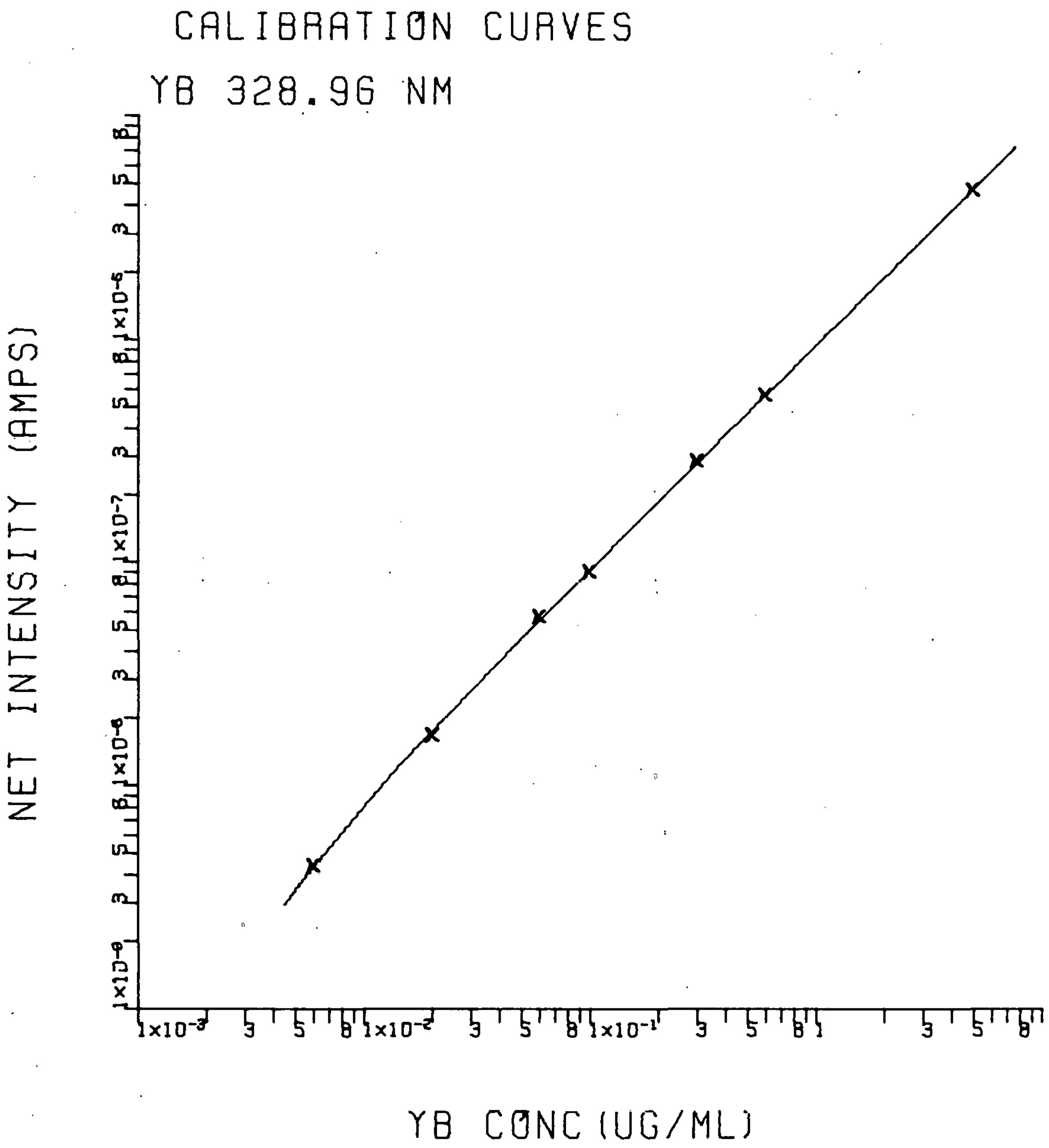

Figure 8. (continued)

range, spectral line interferences may arise from high concentrations of concomitants. In these cases the interfering emission usually lies within 0.04 nm of the analyte wavelength. Wavelength profiles provide the best assessment of potential stray light or spectral interference problems and are especially useful in the preliminary investigative stages for new sample composition. Data obtained with the profile subroutine can easily be plotted directly on the video terminal. The software utilized for plotting data on the video terminal is listed in Appendix 2. The potentiality of these interferences were investigated by obtaining profiles of emission intensity versus wavelength for the four strongest emission lines for each element of interest (80). Each profile consists of a small wavelength region (0.3 nm) which encompasses one of the analytical lines of interest. Additionally, the spectra of single element solutions containing 1000 ug/mL of Al, Ca, Fe, Mg, and 200 ug/mL of Ti, Cr, Cu, Mn, Ni, and V were profiled over the same wavelength region. All profiles were obtained by stepping the monochromator across the wavelength region and taking intensity measurements every 0.01 nm. By superimposing these eleven profiles onto one graph, potential spectral line interferences arising from these elements could be determined. The wavelength profiles were plotted for both linear and logarithmic photocurrent intensity scales. Several examples of typical graphs observed in our studies are shown in Figures 9 and 10. These profiles illustrate several problems that may arise when complex samples are analyzed. In Figure 9, 1000 ug/mL of Al, which would be equivalent to an Al content in a sample of 50%, increases the intensity of the spectral background significantly from that when no Al is present in the sample. Without an accurate background correction scheme, the measurement of the emission intensity at the respective analytical wavelengths would result in an analytical bias. Figure 10A shows an example of a direct spectral overlap among three analytical wavelengths. Both the Al 309.27 nm and Mg 309.30 nm spectral lines interfere with the V 309.31 nm wavelength. Because the interferences due to 1000 ug/mL of Al and Mg were equivalent to 100 and 8 ug/mL, respectively, of V, this line was not chosen for analytical purposes. Figure 10B shows a tracing of V 310.23 nm, which was chosen for analytical purposes. The absence of any significant spectral interference from the elements studied make this analytical wavelength a good choice for analytical work. The small hump underneath the analytical line is part of an OH band. Because this OH band is also present in the blank spectra, the software will automatically make corrections for its presence. In the current investigation a judicious effort was expended in selecting interference free spectral lines, occasionally to the detriment of the detection limit. When a

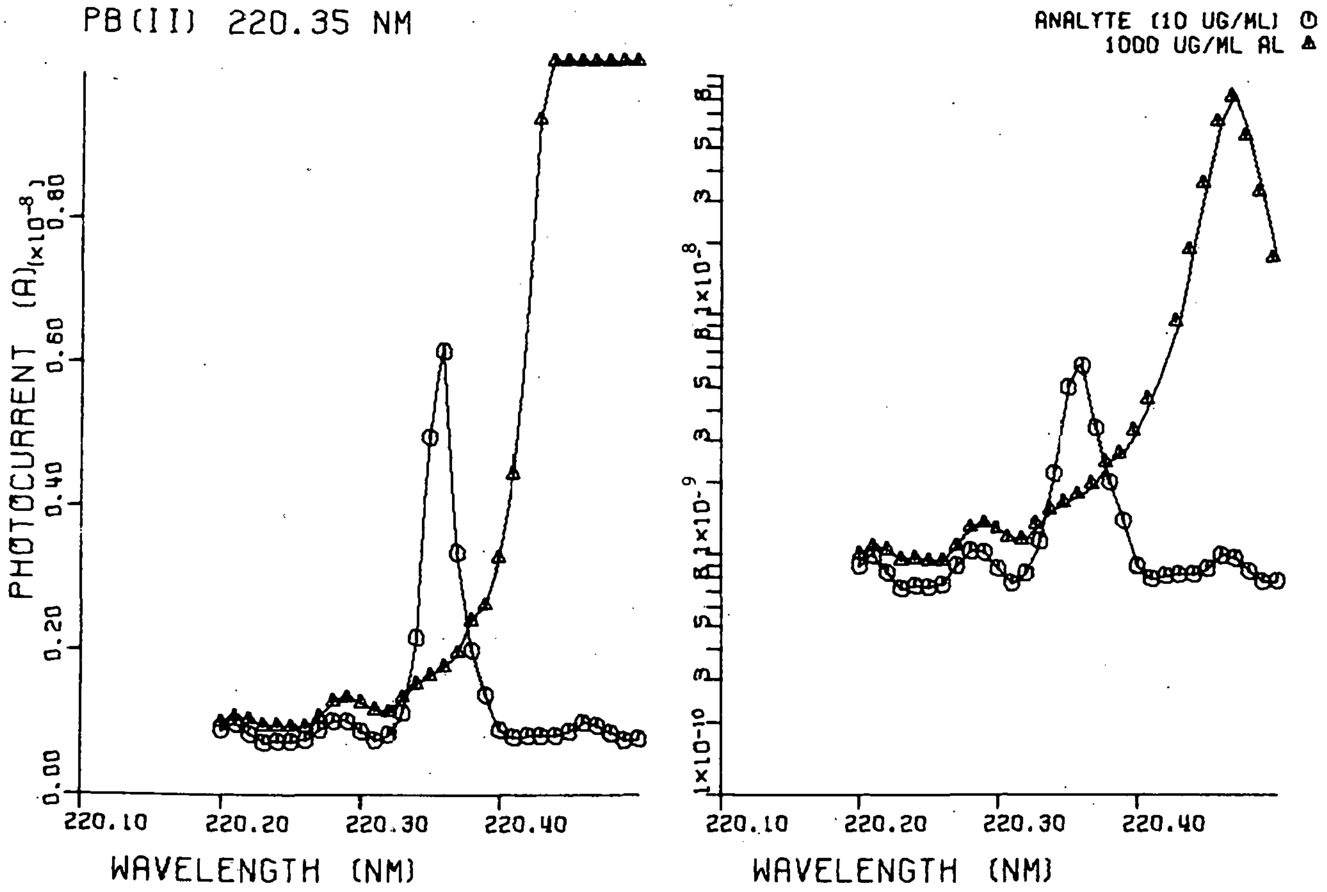

Figure 9. Wavelength profile of the Pb 220.3 nm analytical line demonstrating the magnitude of the background shift in the presence of 1000 ug/mL of Al

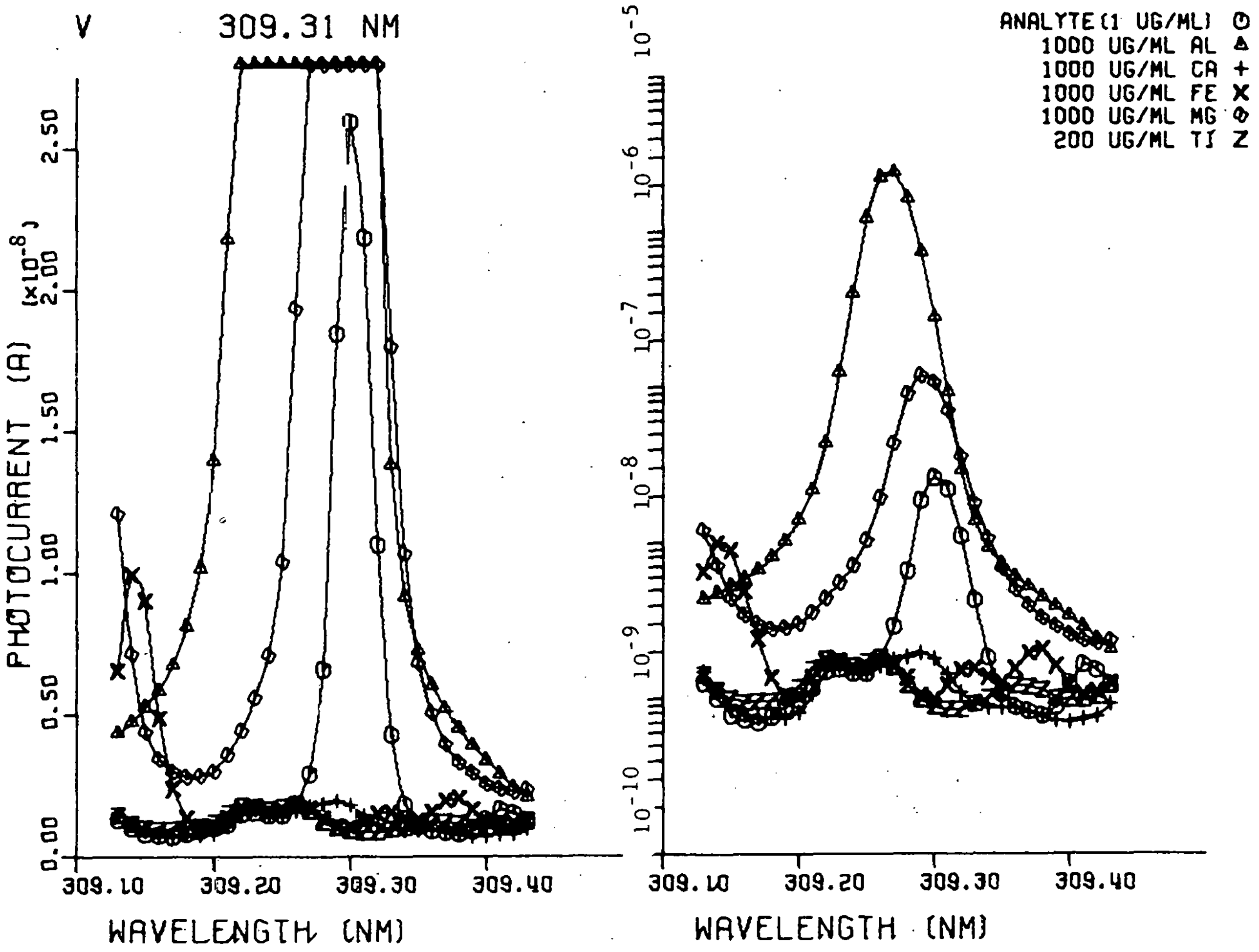

Figure 10. (A) Wavelength profile of the V 309.31 nm analytical line demonstrating the problem of spectral interferences. The Al 309.27 nm line and Mg 309.30 line overlap the V 309.31 line

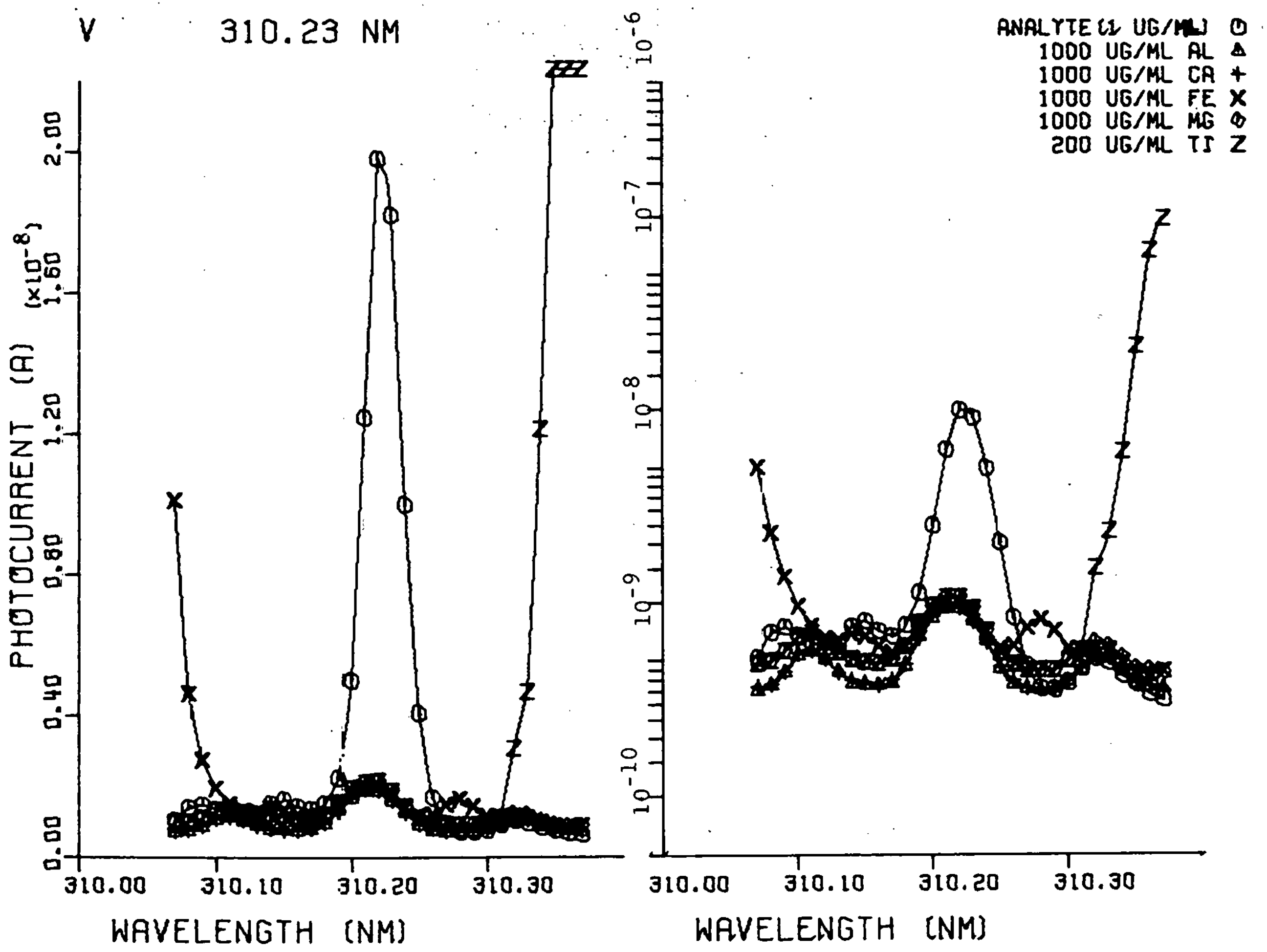

Figure 10. (B) Wavelength profile demonstrating the general absence of interferences at common element/V concentration ratios of 1000 at the V 310.23 nm analytical line

polychromator is utilized for the analysis of complex materials, the option of choosing appropriate analytical lines for trace analyis is not available.

Accuracy and precision

The accuracy of the ICP-AES technique for the determination of major, minor, and trace elements was assessed by analyzing a series of NBS standard reference materials (coal fly ash, urban particulates, and river sediment), three USGS synthetic glasses, and four USGS standard rocks of known elemental composition. The experimental conditions defined in Table I were used. The results of these determinations are presented in Tables IX-XIV. Each value represents the average of five or more determinations. The accuracy data are given only for those elements for which the concentrations exceeded the LQD values. As noted above, quantitative determinations can usually be made with satisfactory accuracy and precision above this concentration level (44). As shown by Tables IX-XIV, the ICP emission values obtained during this study agreed well with the reference values reported in the literature (81-85). Relative standard deviations ranged from a fraction of one percent to several percent. The fact that both major and trace elements exhibited acceptable accuracy indicates that the calibration scheme accommodates any nonlinearity in the analytical calibration curves. The accuracy achieved for the trace elements is a reflection of the fact that a single set of spectral lines may show negligible spectral interferences regardless of the sample matrix and that a single set of lines are applicable to a broad range of sample composition. Of the relatively large amount of data accumulated during this investigation on diverse types of samples, only one significant discrepancy was observed. In Table XIV, for the ore samples PCC-1 and DTS-1, the values for Cr are noted to be significantly lower than the USGS recommended values. It is of interest to note that Church (60), who also employed the ICP-AES approach but with a different dissolution procedure (HF-HClO4-HNO3), obtained values very close to ours (2330 and 1554 ug/g, respectively, for DTS-1 and PCC-1). The discrepancy between the the USGS recommended values and those found by ICP-AES is still under investigation.

The extensive data shown in Tables IX to XIV document that a single analytical calibration for each element suffices for the quantitative determination of that element in a variety of inorganic materials, even at the trace concentration level. To achieve this flexibility, the analyte line must be selected with care and capability for making background corrections must be provided. As shown in this thesis, both the careful line selection and background correction processes are facilitated through the use of a programmable scanning monochromator.

Table IX. Analytical Results for NBS Coal Flyash

	CONCENTRATIONS (ug/g unless % indicated)			
ELEMENT	AL[a]	NBS	ONDOV[b]	FISHER[c]
Al	12.8%	-	12.7%	12.7%
Ca	4.5%	-	4.7%	4.3%
Fe	6.0%	-	6.2%	6.4%
Mg	1.8%	-	1.8%	1.2%
Ti	7330	-	7400	6800
As	56	61	58	-
Ba	2800	-	2700	2210
Be	11	12	-	12
Ce	140	-	146	-
Co	41	38	42	42
Cr	123	131	127	128
Cu	120	128	-	121
Eu	2	-	25	-
Ga	72	-	77	-
La	78	-	82	-
Mn	510	493	496	485
Ni	94	98	98	99
Pb	81	70	75	70
Pr	24	-	-	-
Sc	20	-	27	-
Sr	1620	1380	1700	1340
V	233	214	235	-
Y	65	-	62	-
Yb	7	-	7	-
Zn	212	210	216	221
Zr	288	-	301	-

[a] Ames Laboratory result. Average of ten determinations.

[b] Reference No. 81.

[c] Reference No. 82.

Table X. Analytical Results for USGS Coals[a]

	CONCENTRATIONS (ug/g)					
	PAS 44		PAS 47		PAS 48	
	AL[b]	USGS[c]	AL	USGS	AL	USGS
Zn	141	140	185	200	85	90
Co	40	36	80	89	30	32
Ni	85	96	179	170	88	84
Mn	105	103	249	235	300	297
Cr	104	99	200	236	131	120
Ga	68	52	43	51	38	46
V	136	152	189	195	198	195
Be	28	27	14	15	17	17
Cu	61	54	60	56	61	52
Yb	9	7	5.8	4	6	4
La	-	<5	73	75	81	83
Sc	30	27	41	39	18	25
Y	85	89	78	82	78	71
Eu	2.7	-	2.0	-	2.1	-
Ba	200	193	340	337	405	381
Pr	4.6	-	17	-	14	-
Sr	423	498	401	406	1025	985

[a] Samples collected at a Pittsburgh coal bed, Washington County, Pa. Literature values are not certified.

[b] Ames Laboratory result. Average of five determinations.

[c] See reference 83.

Table XI. Analytical Results for NBS River Sediment

CONCENTRATIONS (ug/g unless % indicated)

ELEMENT	AL[a]	NBS
Al	3.9%	4.0%
Ca	4.2%	4.0%
Fe	11.5%	11.3%
Mg	4.1%	4.0%
As	71	66
Ba	400	-
Be	1	-
Cd	10	10
Co	11	8
Cr	3.02%	2.96%
Cu	119	109
Eu	0.7	-
Ga	38	-
La	15	9
Mn	750	785
Ni	45	46
Pb	725	714
Pr	14	-
Sc	5	2
Sr	1200	-
V	25	24
Y	7	-
Yb	0.6	-
Zn	1587	1720

[a] Ames Laboratory result. Average of five determinations.

Table XII. Analytical Results for NBS Urban Particulate Matter

CONCENTRATIONS (ug/g unless % indicated)

ELEMENT	AL[a]	NBS
Al	3.3%	3.3%
Fe	4.1%	3.9%
Mg	0.9%	0.8%
Ti	0.4%	0.4%
As	112	115
Ba	774	737
Be	3	-
Cd	73	75
Ce	61	55
Co	28	18
Cr	398	403
Cu	598	609
Eu	1	0.8
Ga	72	-
La	35	42
Mn	851	860
Ni	85	82
Pb	0.7%	0.7%
Pr	8	-
Sc	4	7
Sr	450	-
V	106	130
Y	5	-
Yb	2	-
Zn	4700	4760

---

[a] Ames Laboratory results. Average of five determinations.

Table XIII. Analytical Results for USGS Glasses

CONCENTRATIONS (ug/g)

	Sample GSC			Sample GSD			Sample GSE		
	AL[a]	%RSD	USGS[b]	AL	%RSD	USGS	AL	%RSD	USGS
As 197.2							430	4.3	450
Ba 455.4	43	6.1	39	92	3.8	90	510	1.6	500
Be 313.0	3.2	12.	3.5	40	5.3	44	512	2.3	500
Bi 233.1				45	9.8	40	464	5.1	480
Cd 214.4	2.6	12.	3.0	28	8.4	30	425	1.4	420
Co 228.6	6	5.1	6	33	4.2	35	458	0.9	450
Cr 267.7	11	12.	7	44	5.2	47	480	1.0	490
Cu 324.7	6	9.8	9	40	6.2	45	502	2.3	500
Eu 420.5				54	6.7	50	581	5.1	600
Ga 294.4							26	20.	20
Hf 277.3				56	20.	45	519	8.4	500
Ir 224.3				16	11.	20	151	7.1	140
La 394.9				51	4.5	47	558	2.4	550
Mn 257.6	215	4.3	200	212	4.5	210	596	1.0	600
Mo 202.0							521	12.	500
Ni 231.6				50	14.	54	507	4.5	500
Pd 340.5				43	11.	36	95	9.7	100
Ru 240.3							96	10.	90
Sc 361.4				3	16.	3	32	7.1	30
Sr 407.7	31	4.8	27	66	4.4	64	503	0.7	500
Ti 334.9	8	13.	11	50	7.1	44	499	3.2	490
W 207.9							403	11.	420
Y 371.0	7	6.2	8	47	3.4	48	501	0.7	490
Yb 328.9							27	8.2	30
Zn 213.8	15	6.4	12	39	5.7	43	506	2.1	500
Zr 343.8				55	6.2	48	483	3.1	480

[a] Ames Laboratory results. Average of ten determinations.

[b] See Reference 84.

Table XIV. Analytical Results for USGS Rocks

CONCENTRATIONS (ug/g unless % indicated)

	BCR-1		DTS-1		GSP-1		PCC-1	
	AL[a]	USGS[b]	AL	USGS	AL	USGS	AL	USGS
Al	16.8%	13.6%	0.2%	0.2%	14.5%	15.2%	1.0%	0.8%
Ca	6.9%	6.9%	0.1%	0.1%	2.2%	2.0%	0.5%	0.5%
Fe	13.7%	13.4%	8.4%	8.6%	4.3%	4.3%	9.0%	8.4%
Mg	3.7%	3.5%	[c]	49.8%	1.1%	1.0%	[c]	42.0%
Ti	2.1%	2.2%	120	130	0.7%	0.7%	137	150
Ba	661	675	2.5	2.4	1290	1300	1.4	1.2
Be	2.9	1.7	-	-	1.8	1.5	-	-
Ce	50	54	-	-	401	394	-	-
Co	41	38	141	133	12	7	130	112
Cr	19	18	2110	3600	16	12	1585	2730
Cu	18	18	6.6	7	40	33	9	11
Dy	7	6	-	-	6.8	5.4	-	-
Er	4	4	-	-	-	-	-	-
Eu	2.5	1.9	-	-	2.0	2.4	-	-
Ga	19.6	20	-	-	27	22	-	-
Gd	-	-	-	-	19	15	-	-
Hf	-	-	-	-	17	16	-	-
Ir	-	-	-	-	-	-	7	5
La	24	26	-	-	202	191	-	-
Lu	0.6	0.5	-	-	-	-	-	-
Mn	1455	1406	960	969	317	331	905	959
Nb	13.4	13.5	-	-	36	29	-	-
Nd	32	29	-	-	175	188	-	-
Ni	30	16	2390	2270	17	13	2681	2339
Pr	5	7	-	-	45	50	-	-
Sc	29	33	3.5	3.6	7.4	7.1	6.6	6.9
Sr	338	330	0.3	0.4	241	233	-	-
Th	7.5	6.0	-	-	92	104	-	-
V	375	399	10.2	10.3	60	53	38	30
Y	36	37	-	-	28	30	-	-
Yb	3.6	3.4	-	-	1.6	1.8	-	-
Zn	158	120	51	45	112	98	44	36
Zr	209	190	2.5	3	523	500	7	7

[a] Ames Laboratory result. Average of ten determinations.

[b] See Reference 85.

[c] Concentrations out of calibration range.

BIBLIOGRAPHY

1. Fassel, V. A. Science, 1978, 202, 183.

2. Greenfield, S.; Jones, I. L.; McGeachin, H. M.; Smith, P. B. Anal. Chim. Acta, 1975, 74, 225.

3. Boumans, P. W. J. M.; de Boer, F. J. Proc. Anal. Divn. Chem. Soc., 1975, 12, 140.

4. Winefordner, J. D.; Fitzgerald, J. J.; Omenetto, N. Appl. Spectrosc., 1975, 29, 369.

5. Barnes, R. M. CRC Critical Reviews in Analytical Chemistry, 1978, 7, 203.

6. Spillman, R. W.; Malmstadt, H. V. Anal. Chem., 1976, 48, 303.

7. Catterick, T.; Hickman, D. A. Analyst, 1979, 104, 516.

8. Johnson, D. J.; Plankey, F. W.; Winefordner, J. D. Anal. Chem., 1975, 47, 1739.

9. Kawaguchi, H.; Okada, M.; Ito, T.; Muzuike, A. Anal. Chim. Acta, 1977, 95, 145.

10. Boumans, P. W. J. M.; Van Gool, G. H.; Jansen, J. A. J. Analyst, 1976, 101, 585.

11. Olson, K. W.; Haas, W. J., Jr.; Fassel, V. A. Anal. Chem., 1977, 49, 632.

12. Fassel, V. A.; Kniseley, R. N. Anal. Chem., 1974, 46, 1110A, 1115A.

13. Winge, R. K.; Peterson, V. J.; Fassel, V. A. Appl. Spectrosc., 1979, 33, 206.

14. Meggers, W. F.; Corliss, C. H.; Scribner, B. F. "Tables of Spectral-Line Intensities, Part I - Arranged by Elements"; U. S. Department of Commerce: Washington, D. C., 1975.

15. "Massachusetts Institute of Technology Wavelength Tables"; MIT: Cambridge, Ma, 1969.

16. Zaidel', A. N.; Prokof'ev, V. K.; Raiskii, S. M.; Slavnyi, V. A.; Shreider, E. Ya. "Tables of Spectral Lines", 3rd ed.; Plenum: New York, 1970.

17. Kelly, R. L.; Palumbo, L. J. "Atomic, Ionic Emission Lines Below 2000 Angstroms - Hydrogen Through Krypton"; Naval Research Laboratory: Washington, D. C., 1973.

18. Parsons, M. L.; Smith, B. W.; Bentley, G. E. "Handbook of Flame Spectroscopy"; Plenum: New York, 1975.

19. Roldan, R. Rev. Sci. Instrum., 1969, 40, 1388.

20. Haas, W. J., Jr.; Butler, C. C.; Kniseley, R. N.; Fassel, V. A. Abstracts of the 1977 Pittsburgh Conference on Analytical Chemistry and Applied Spectroscopy.

21. Winge, R. K.; Fassel, V. A.; Kniseley, R. N.; Dekalb, E.; Haas, W. J., Jr. Spectrochim. Acta, 1977, 32b, 327.

22. Reeves, R. R.; Brooks, R. D. "Trace Element Analysis of Geological Materials; Wiley: New York, 1978.

23. Morrison, G. H.; Gerard, J. T.; Travesi, A.; Currie, R. L.; Peterson, S. F.; Potter, N. M. Anal. Chem., 1969, 41, 1633.

24. Grancini, G.; Strievana, M. B.; Girardi, F.; Guzzi, G.; Pietra, R. J. Radioanal. Chem., 1976, 34, 65.

25. Levinson, A. A.; De Pablo, L. J. of Geochemical Exploration, 1975, 4, 399.

26. Giauque, R. D.; Garrett, R. B.; Goda, L. Y. Anal. Chem., 1977, 49, 1012.

27. Giauque, R. D.; Garrett, R. B.; Goda, L. Y. Anal. Chem., 1977, 49, 62.

28. Williams, C. J. Sci. Fd. Agric., 1976, 27, 561.

29. Vanderstappen, M.; Van Grieken, R. Z. Anal. Chem., 1976, 282, 25.

30. Hwang, J. Y. Talanta, 1970, 17, 118.

31. Burkhalter, P. G. Washington, D. C., August 1973, NRL Report 7637.

32. Walter, R. L.; Willis, R. D.; Gutnecht, W. F.; Joyce, J. M. Anal. Chem., 1974, 46, 843.

33. Buam, R.; Gutnecht, W. F.; Willis, R. D.; Walter, R. L. Anal. Chim. Acta, 1976, 85, 323.

34. Bernas, B. Anal. Chem., 1968, 40, 1683.

35. Harstein, A. M.; Freedman, R. W.; Platter, D. W. Anal. Chem., 1973, 45, 611.

36. Agemian, H.; Chau, A. S. Y. Anal. Chim. Acta, 1975, 80, 61.

37. Boar, P. L.; Ingram, L. K. Analyst, 1970, 95, 124.

38. Theis, T. L.; Richter, R. O. Environ. Sci. Technol., 1979, 13, 219.

39. Scanlon, D. H.; Duggan, J. C. Environ. Sci. Technol., 1979, 13, 311.

40. Campbell, J. A.; Laul, J. C.; Nielson, K. K.; Smith, R. D. Anal. Chem., 1978, 50, 1032.

41. Fassel, V. A. American Society for Testing and Materials, Philadelphia, 1977, ASTM Spec. Tech. Pub. 618.

42. Reed, T. B. J. Appl. Phys., 1961, 32, 821.

43. Reed, T. B. Int. Science. Tech., 1962, 42.

44. Fassel, V. A. "XVI Colloquium Spectroscopicum Internationale Heidelberg, 1971"; Adam Hilger: London, 1972.

45. Chase, J. D. J. Appl. Phys., 1969, 40, 318.

46. Chase, J. D. J. Appl. Phys., 1971, 42, 4870.

47. Waldie, B. Conference Proceedings International Round Table On The Study and Application of Transport Phenomena in Thermal Plasmas, Odeillo-Fontromeu, France, September 1975, IV-7.

48. Boulos, M. I. Conference Proceeding International Round Table On The Study and Applications of Transport Phenomena in Thermal Plasmas, Odeillo-Fontromeu, France, September 1975, IV-7.

49. Boulos, M. I. IEEE Trans. Plasma. Sci., 1976, PS-4, No. 1, 28.

50. Dymshits, B. M.; Koretskii, Ya. P. Z. Tekhn. Fiz., 1969, 39, 1039.

51. Klubnikin, V. S. Teplofiz. Vys. Temp., 1975, 13, 473.

52. Larson, G. F.; Fassel, V. A.; Scott, R. H.; Kniseley, R. N. Anal. Chem., 1975, 47, 238.

53. Kalnicky, D. J.; Fassel, V. A.; Kniseley, R. N. Appl. Spectrosc., 1977, 31, 137.

54. Bennett, H. Analyst, 1977, 102, 153.

55. Fassel, V. A. Anal. Chem., 1979, 51, 1290A.

56. Robinson, A. Science, 1978, 199, 1324.

57. Burman, J. O.; Ponter, C.; Bostrom, K. Anal. Chem., 1978, 50, 79.

58. Burman, J. O.; Bostrom, K. Kemisk Tedskrift, 1978, 5, 18.

59. Burman, J. O.; Bostrom, B.; Bostrom, K. Geologiska Foreningsensi Stockholm Forhandlingar, 1977, 99, 102.

60. Church, S. E. Abstracts of the 1977 Pittsburgh Conference on Analytical Chemistry and Applied Spectroscopy.

61. Cruz, R. B.; Silvester, M. D., presented at the 2nd Joint Conference, Chemical Institute of Canada - American Chemical Society, Montreal, May 29 - June 2, 1977.

62. Corbett, J. A. CSIRO Miner. Res. Lab., Invest. Rep., 1977, 121, 18; Chem. Abstr., 87, 145174R.

63. Fassel, V. A. Pure Appl. Chem., 1977, 49, 1533.

64. Dahlquist, R. L.; Knoll, J. W. Appl. Spectrosc., 1978, 32, 1.

65. Jones, J. B., Jr. Communications in Soil Science and Plant Analysis, 1977, 8, 349.

66. Watters, R. L., Jr.; Norris, J. A. Abstracts of the 1978 Pittsburgh Conference on Analytical Chemistry and Applied Spectroscopy.

67. Ward, A. F.; Sobel, H. R.; Crawford, R. L., presented at the Third Annual Federation of Analytical Chemistry and Spectroscopy Societies meeting, Philadelphia, November 1976.

68. Scott, R. H.; Kokot, M. L. Anal. Chim. Acta, 1975, 75, 257.

69. Karacki, S. S.; Corcoran, F. L., Jr. Appl. Spectrosc., 1973, 27, 41.

70. Scott, R. H.; Strasheim, A.; Kokot, M. L. Anal. Chim. Acta, 1976, 82, 67.

71. Ward, A. F.; Marciello, L. F. Anal. Chem., 1979, 51, 2264.

72. Wemyss, R. B.; Scott, R. H. Anal. Chem., 1978, 50, 1694.

73. Motooka, J. M.; Mosier, E. L.; Sutley, S. J.; Viets, J. G. Appl. Spectrosc., 1979, 33, 456.

74. Irons, R. D.; Schenk, E. A.; Giauque, R. D. Clin. Chem., 1976, 22, 2018.

75. Commission on Spectrochemical and Other Optical Procedures for Analysis, Pure Appl. Chem., 1976, 45, 99.

76. Zeegers, P. J. T.; Smith, R.; Winefordner, J. D. Anal. Chem., 1977, 40, 27A.

77. Golightly, D. W. ICP Newsletter, 1977, Vol. 3, No. 4, 137.

78. Govindaraju, K.; Melvelle, G.; Chourard, C. Anal. Chem., 1976, 48, 1325.

79. Burman, J.; Bostrom, K. Anal. Chem., 1979, 51, 516.

80. Floyd, M. A.; Winge, R. K.; Fassel, V. A. Unpublished data, Ames Laboratory, Ames, Iowa 50011.

81. Ondov, J. M.; Zoller, W. H.; Olmez, I.; Aras, N. K.; Gordon, G. E.; Rancitelli, L. A.; Abel, K. H.; Filby, R. H.; Shah, K. R.; Ragaini, R. C. Anal. Chem., 1975, 47, 1102.

82. Silberman, D.; Fisher, G. Anal. Chim. Acta, 1979, 106, 299.

83. Medlin, J. H. USGS, Ralston, Va., private communication.

84. Myers, A. T.; Havens, R. G.; Conner, J. J.; Conklin, N. M.; Rose, H. J., Jr. USGS Professional Paper 1013.

85. Flanagan, F. J. Geochem. et Cosmochim. Acta, 1972, 37, 1189.

ACKNOWLEDGEMENTS

The patient encouragement of Dr. V. A. Fassel and Mr. Arthur P. D'silva is gratefully acknowledged. I also thank Chuck Taylor and the U. S. Environmental Protection Agency for their continued interest and financial support.

Mr. Royce K. Winge and Dr. William J. Haas, Jr. have contributed to this research through numerous discussions. I thank them for unselfishly placing their extensive experience in emission spectroscopy at my disposal. Rick Ethington and Dr. Joseph M. Katzenberger designed the computer interface used in this research.

Finally, I thank my family for their interest and encouragement throughout my education. Special thanks are due to my wife, Robyn, whose patience and understanding throughout my education are greatly appreciated. I hope my wife and my family share the sense of pride and accomplishment that this research has generated in me.

APPENDIX 1: SURVEY

SURVEY is the FORTRAN program which controls the operation of the scanning monochromator, collects and analyzes experimental data, and prints the results. A detailed discussion of this software can be found in Chapter II. This software was designed to execute on a Digital Equipment Corporation (DEC) PDP-11/03 minicomputer with 24K bytes of memory. A listing of the source statements of SURVEY is given on the following pages.

```
      PROGRAM SURVEY
C
C     AMES LABORATORY, AMES, IOWA  50011
C        1977/1978
C
      LOGICAL*1 L
      COMMON /IO/INPUT,LOG
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      COMMON /COUNT/ICOUNT
      COMMON /FILES/STRING(4),L
      REAL*4 ANAL,CALI,DETLM,DIR,INITIA,LIST,LOCAT,
     1 PROFI,REPEA,SCA,SEEK,TAPE,AHALT,CHOICE
      DATA ANAL,CALI,DETLM,DIR,INITIA,LIST,LOCAT,PROFI,
     1 REPEA,SCA,SEEK,TAPE,AHALT/'ANAL','CALI','DETL','DIRE',
     1 'INIT',
     1 'LIST','PEAK','PROF','TAKE','SCAN','SEEK','TAPE','HALT'/
      DATA STRING(1),CHOICE/'DX1:','CHOI'/
      CALL RCHAIN(IFLAG,INPUT,12)
      CALL SETLOC
      CALL RANGE(14)    !SET C/F RANGE TO 14
      IF(IFLAG.LT.0)GO TO 1
      INPUT=5     !LOGICAL UNIT 5=VIDEO TERMINAL
      LOG=0  !LOG IS LOGICAL UNIT # OF OUTPUT FILE
      ICOUNT=0
      CALL SETUP
      CALL INIT
1     CALL ASK('? ','C',INT,ANS,L)
      IF(ANS.EQ.AHALT)STOP 'PROGRAM TERMINATED BY OPERATOR.'
      IF(ANS.EQ.ANAL) CALL ANALYT
      IF(ANS.EQ.CALI) CALL CALIB
      IF(ANS.EQ.DETLM) CALL DETLIM
      IF(ANS.EQ.DIR) CALL DIRECT
      IF(ANS.EQ.INITIA) CALL INIT
      IF(ANS.EQ.LIST) CALL LISTVT
      IF(ANS.EQ.LOCAT) CALL LOCATE
      IF(ANS.EQ.PROFI) CALL PROFIL
      IF(ANS.EQ.REPEA) CALL REPT
      IF(ANS.EQ.SCA) CALL SCAN
      IF(ANS.EQ.SEEK)CALL POSITION
      IF(ANS.EQ.TAPE)CALL ATAPE
      IF(ANS.EQ.CHOICE)TYPE 10,ANAL,CALI,DETLM,DIR,INITIA,
      1  LIST,LOCAT,PROFI,REPEA,SCA,SEEK,TAPE,AHALT
10    FORMAT(11(' ',A4/))
      GO TO 1
      END
```

```
      FUNCTION AMAXIN(PEAK)


      LOGICAL*1 FLAG
      DIMENSION PEAK(3)
      FLAG=.FALSE.
      DO 1 I=1,3
1     IF(PEAK(I).LT.0.0)FLAG=.TRUE.
      IF(.NOT. FLAG)GO TO 2
      TYPE 3
3     FORMAT(' INTENSITIES ARE NEGATIVE!!!')
      AMAXIN=0.0
      RETURN
2     A=-((ALOG(PEAK(1))+ALOG(PEAK(3))-(2.0*ALOG(PEAK(2))))
     1 /2.0)
      D=(ALOG(PEAK(2))-ALOG(PEAK(3)))/(2.0*A)
      X2=(D-0.5)**2
      X3=(D+0.5)**2
      AMAXIN=SQRT(PEAK(2)*PEAK(3)*EXP(A*(X2+X3)))
      RETURN
      END


      SUBROUTINE ANALYT


      LOGICAL*1 L,LHEAD(40),LESCAP,LBKG
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      COMMON /IO/INPUT,LOG
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      EQUIVALENCE (LHEAD(1),WORD1)
      DATA STAND/'STAN'/
      CALL OUTPUT('O')
      CALL HEAD('ANALYTICAL ROUTINE')
      CALL DATFIL
      IF(L)GO TO 8
      NTERMS=KOUNT(0)
      LBKG=.FALSE.
      CALL ASK('INTEGRATION TIME? ','R',INT,SEC,L)
      IF(L)GO TO 7
      CALL NORY('PRINT BACK INTENSITY DATA? ',NPRINT)
      CALL NORY('SUBTRACT BLANK INTENSITY VALUES? ',IGO)
      IF(IGO.LE.0)GO TO 30
```

```
      CALL ZERO
      IF(L)GO TO 7
      LBKG=.TRUE.
      DO 15 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)GO TO 30
      CALL GDATA(CINTEN(16),AMP)
      IF(L)GO TO 51
15    CALL AWRITE(I)
30    CALL ZERO
      IF(L)GO TO 51
      CALL GETNAM(LHEAD)
      IF(WORD1.EQ.STAND)CALL UPDATE
      IF(L)GO TO 51
      CALL ASK('DILUTION FACTOR? ','R',INT,DILUTE,L)
      IF(L)GO TO 51
      IF(DILUTE.LE.0.0)DILUTE=1.0
      IF(NPRINT.GT.0)TYPE 99,LHEAD
      IF(LOG.NE.0 .AND. NPRINT.GT.0)WRITE(LOG,99)LHEAD
99    FORMAT('0',40A1/'0ELEM   WAVE       INTEN        BKG
      1       NET INT      CAL CONC.')
      IF(NPRINT.LE.0)TYPE 200, LHEAD
      IF(LOG.NE.0 .AND. NPRINT.LE.0)WRITE(LOG,200)LHEAD
      DO 50 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)GO TO 60
      CALL GDATA(SINTEN,AMP)
      IF(L)GO TO 51
      YCALC=ORTVAL(SINTEN-CINTEN(16),NTERMS)
      YCALC=YCALC*DILUTE
      IF(NPRINT.LE.0)GO TO 201
      TYPE 98,ISYMB,WAVE,SINTEN+AMP,AMP,SINTEN-CINTEN(16),
     1   YCALC
      IF(LOG.NE.0)WRITE(LOG,98)
      1 ISYMB,WAVE,SINTEN+AMP,AMP,SINTEN-CINTEN(16),YCALC
98    FORMAT(' ',1X,A2,1X,F9.4,3(1X,1PE12.3),G13.3)
      GO TO 50
200   FORMAT('0',40A1/
     1  '0  ELEM       WAVE        PRIOR     CAL CONC.')
201   TYPE 55,ISYMB,WAVE,IPRIOR,YCALC
      IF(LOG.NE.0)WRITE(LOG,55)ISYMB,WAVE,IPRIOR,YCALC
55    FORMAT(' ',3X,A2,4X,F11.4,6X,I2,6X,G13.3)
50    CONTINUE
60    CALL NORY('MORE UNKNOWNS? ',IGO)
      IF(IGO.GT.0)GO TO 30
51    IF(.NOT.LBKG)GO TO 7
      DO 52 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)GO TO 7
      CINTEN(16)=0.0
```

```
52    CALL AWRITE(I)
7     CALL CLOSE(3)
8     CALL OUTPUT('C')
      IF(L)CALL ERROR('ANALYT')
      RETURN
      END
```

```
      SUBROUTINE AREAD(NREC)



      LOGICAL*1 L
      COMMON /RECORD/DUMMY(47),L
      READ(3'NREC)DUMMY
      RETURN
      END
```

```
      SUBROUTINE ASK(Q,IR,INT,REAL,L)



      COMMON /IO/ INPUT,LOG
      LOGICAL*1 FIRST,Q(1),ESC,L,IR,I,R,C,STRING(16)
      REAL*4 ANS(4)
      EQUIVALENCE (FIRST,ANS,STRING)
      DATA ESC,I,R,C/'33,'I','R','C'/
5     CALL ITYPE(Q)  !TYPE QUESTION ON TERMINAL
      CALL GETSTR(INPUT,STRING,15)   !GET ANSWER
      L=FIRST.EQ.ESC  !WAS ALTMODE KEY TYPED?
      IF(L)RETURN     !YES - THEN RETURN
      IF(IR.EQ.I)DECODE(LEN(ANS),20,ANS,ERR=100)INT   !INTEGER
20    FORMAT(I16)
      IF(IR.EQ.R)DECODE(LEN(ANS),30,ANS,ERR=100)REAL   !REAL
30    FORMAT(E16.0)
      IF(IR.EQ.C)REAL=ANS(1)    !CHARACTER
      RETURN
100   CALL ITYPE('INPUT ERROR')   !WRONG ANSWER WAS INPUT
      GO TO 5
      END
```

```
      SUBROUTINE ATAPE



      COMMON /IO/INPUT,LOG
      LOGICAL*1 L
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  DUMMY(28),L
      DATA IEND/'EN'/
      DO 1 I=1,28
1     DUMMY(I)=0.0
      CORR2=-0.1
      IPRIOR=1
      DO 2 I=1,15
2     CONC(I)=0.0
      CALL NORY('MODIFY AN OLD FILE? ',IGO)
      IF(IGO.LE.0)GO TO 3
      CALL DATFIL
      IF(L) RETURN
      GO TO 100
3     CALL GETFIL('NAME OF FILE FOR DATA? ',3,'NEW','NC',L)
      DEFINE FILE 3(0,94,U,NREC3)
      CALL ITYPE('ENTER SYMBOLS AND WAVELENGTHS')
      DO 5 I=1,500
      READ(INPUT,10)ISYMB,WAVE
10    FORMAT(A2,1X,G15.0)
      IF(ISYMB.EQ.IEND)GO TO 15
      CORR1=WAVE-0.1
      CALL AWRITE(I)
5     CONTINUE
15    WAVE=0.0
      CALL AWRITE(I)
      CALL NORY('ADD CONCENTRATIONS OF REFERENCE SOLUTIONS? '
     1    ,IGO)
      IF(IGO.LE.0)GO TO 30
      CALL ASK('# OF STANDARDS? ','I',NSTD,REAL,L)
      IF(L)GO TO 500
      DO 20 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.0)GO TO 30
      TYPE 25,ISYMB,WAVE
25    FORMAT('$ENTER CONC. FOR ',A2,F11.4,2X)
      READ(INPUT,26) (CONC(J),J=1,NSTD)
26    FORMAT(15G15.0)
      CALL AWRITE(I)
20    CONTINUE
30    CALL WSORT(I)
      CALL NORY('ADD CORRECTION WAVELENGTHS? ',IGO)
      IF(IGO.LE.0)GO TO 100
```

```
        DO 31 I=1,500
        CALL AREAD(I)
        IF(WAVE.EQ.0.0)GO TO 100
        TYPE 32,ISYMB,WAVE
32      FORMAT('$ENTER CORRECTION FOR ',A2,F11.4,2X)
        READ(INPUT,26)CORR2
        CORR1=WAVE+CORR2
        CALL AWRITE(I)
31      CONTINUE
100     CALL TOUT
        IF(L)GO TO 500
        CALL NORY('MAKE CHANGES? ',IGO)
        IF(IGO.LE. 0) GO TO 110
        CALL CHANGE
        IF(L)GO TO 500
        GO TO 100
110     CALL CLOSE(3)
        RETURN
500     CALL ERROR('ATAPE')
        CALL CLOSE(3)
        RETURN
        END


        SUBROUTINE AWRITE(NREC)


        LOGICAL*1 L
        COMMON /RECORD/DUMMY(47),L
        WRITE(3'NREC)DUMMY
        RETURN
        END


        SUBROUTINE BIGSUB(J)


        LOGICAL*1 L,LESCAP
        COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
        1  CINTEN(16),B(4),C(4),D(4),L
        COMMON /IO/INPUT,LOG
        NPRINT=0
```

```
      IF(J.NE.16)CALL NORY('PRINT BACK INTENSITY DATA? ',
     1  NPRINT)
      IF(NPRINT.GT.0)TYPE 40,J
      IF(LOG.NE.0 .AND. NPRINT.GT.0) WRITE(LOG,40) J
40    FORMAT('OSTANDARD #',I4/
      1 'OELEM    WAVE        INTEN           BKG          NET INT')
      DO 141 N=1,500
      CALL AREAD(N)
      IF(WAVE.EQ.0.)RETURN
      IF(CONC(J).LT.0.0)GO TO 141
      CALL GDATA(CINTEN(J),BKG)
      IF(L)RETURN
      CALL AWRITE(N)
      IF(NPRINT.GT.0) TYPE 20,ISYMB,WAVE,CINTEN(J)+BKG
      1   ,BKG,CINTEN(J)-CINTEN(16)
      IF(LOG.NE.0 .AND. NPRINT.GT.0)WRITE(LOG,20)
      1  ISYMB,WAVE,CINTEN(J)+BKG,BKG,CINTEN(J)-CINTEN(16)
20    FORMAT(' ',1X,A2,1X,0PF9.4,1X,3(1PE12.3))
141   CONTINUE
      RETURN
      END


      SUBROUTINE BKG(AMP)



      LOGICAL*1 L,LESCAP
      COMMON /RECORD/ISYMB,WAVE,CORR1,IDUM(89),L
      CALL SEEK(CORR1)
      CALL TAKEDA(1,AMP,STDAMP,0)
      IF(.NOT.LESCAP())RETURN
      CALL ERROR('BKG')
      RETURN
      END



      SUBROUTINE CALIB



      LOGICAL*1 L
      COMMON /RECORD/DUMMY(47),L
      L=.FALSE.
```

```
      CALL NORY('DO YOU WANT TO LOOK AT AN OLD CALIBRATION? '
      1,IGO)
      IF(IGO)105,1,49
49    CALL DATFIL
      IF(L)GO TO 1
      CALL OUTPUT('O')
      CALL HEAD('CALIBRATION ROUTINE')
      GO TO 50
105   CALL CTAPE
      IF(L)GO TO 999
      CALL NORY('CONTINUE? ',IGO)
      IF(IGO.LE.0)GO TO 999
      CALL DATGET
      IF(L) GO TO 999
      CALL FIT(.FALSE.)
      IF(L) GO TO 999
50    CALL NORY('DO YOU WANT TO GRAPH YOUR CURVES? ',IGO)
100   IF(IGO.GT.0)CALL GRAPH
      IF(L) GO TO 999
      CALL NORY('DO YOU WANT TO LOOK AT THE CURVE DATA? '
     1  ,IGO)
      IF(IGO.GT.0)CALL LISTIT
      IF(L) GO TO 999
      CALL NORY('ELIMINATE ANY POINTS? ',IGO)
      IF(IGO.GT.0)CALL ELIM
      IF(L)GO TO 999
      CALL NORY('DO YOU WANT TO GRAPH YOUR CALIBRATION
     1   CURVES? ',IGO)
      IF(IGO.GT.0)GO TO 100
999   CALL CLOSE(3)
      CALL OUTPUT('C')
1     RETURN
      END


      SUBROUTINE CD(LASCUR)



      INTEGER*2 LASCUR(2)
      INTEGER*4 TIME,NTIME,DT
      LOGICAL*1 LWORD,HWORD
      INTEGER*2 HIGH,NTIM(2),CDSR,CDIN,CDOUT
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      EQUIVALENCE (NTIM(1),NTIME),(NT2,NTIM(2))
      DATA CDSR,CDOUT,CDIN/"167770,"167772,"167774/
      DATA LWORD,HWORD/"41,"40/
```

```
C
C      CDSR IS THE CURRENT TO DIGITAL STATUS REGISTER
C      CDOUT IS THE OUTPUT REGISTER
C      CDIN IS THE DATA INPUT REGISTER
C
       LOW=IPEEK(CDOUT).AND."176777
       HIGH=LOW.OR."1000
       CALL IPOKE(CDOUT,LOW)
       CALL GTIM(TIME)
       CALL JJCVT(TIME)
       CALL JADD(NT,TIME,DT)
15     CALL GTIM(NTIME)
       IF(NT2.EQ.TIME)GO TO 15  !WAIT TILL ONE TICK HAS PASSED
       CALL IPOKE(CDOUT,HIGH)
       CALL IPOKE(CDOUT,LOW)   !AND STARTS COUNTING.
20     CALL GTIM(NTIME)
       IF(NT2.NE.DT)GO TO 20  !ARE WE DONE YET?
       CALL IPOKE(CDOUT,HIGH)
       CALL IPOKE(CDOUT,LOW)
       CALL IPOKEB(CDSR,LWORD)
       LASCUR(2)=IPEEK(CDIN)/256
       CALL IPOKEB(CDSR,HWORD)
       LASCUR(1)=IPEEK(CDIN)  !GET MOST SIG. DIGITS FROM CDIN
       RETURN    !RETURN # OF COUNTS IN CONVERTER
       END


       SUBROUTINE CHANGE


       DIMENSION NSYMB(2),SCONC(15)
       LOGICAL*1 L,LC(15),LQ(4),BLANK,LWAVE
       COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IP,CONC(15),
      1  IDUM(56),L
       COMMON /IO/INPUT,LOG
       EQUIVALENCE(SYMB,NSYMB(1),LQ(1))
       DATA BLANK/' '/
       LWAVE=.FALSE.
81     CALL ASK('WHICH ELEMENT? ','C',INT,SYMB,L)
       IF(L)RETURN
       IF(LEN(LQ).LT.2)CALL IPOKEB(IADDR(LQ)+1,BLANK)
       CALL NORY('CHANGE WAVELENGTH? ',IGO)
       IF(IGO.LE.0)GO TO 101
       LWAVE=.TRUE.
       NLINES=0
       DO 102 I=1,500
```

```
        CALL AREAD(I)
        NLINES=NLINES+1
        IF(WAVE.EQ.0.0)GO TO 101
        IF(ISYMB.NE.NSYMB(1))GO TO 102
        TYPE 2,ISYMB,WAVE
2       FORMAT(' ',A2,F11.4)
        CALL NORY('CHANGE? ',IGO)
        IF(IGO.LE.0)GO TO 102
        CALL ASK('NEW WAVELENGTH? ','R',INT,WAVE,L)
        IF(L)RETURN
        CORR1=WAVE+CORR2
        CALL AWRITE(I)
102     CONTINUE
101     CALL NORY('CHANGE CONCENTRATIONS? ',IGO)
        IF(IGO.LE.0)GO TO 51
        DO 60 I=1,15
60      SCONC(I)=0.0
        TYPE 32
32      FORMAT(' WHICH STANDARD?(ANSWER T OR F)'/' ',9X,
        1  '111111'/' ','123456789012345')
        ACCEPT 33,LC
33      FORMAT(15L1)
        DO 40 I=1,15
        IF(.NOT.LC(I))GO TO 40
        TYPE 35,I
35      FORMAT(' STANDARD ',I2,1X,$)
        ACCEPT 38,SCONC(I)
38      FORMAT(G15.0)
40      CONTINUE
        DO 50 I=1,500
        CALL AREAD(I)
        IF(WAVE.EQ.0.)GO TO 51
        IF(ISYMB.NE.NSYMB(1))GO TO 50
        DO 49 K=1,15
49      IF(SCONC(K).NE.0.)CONC(K)=SCONC(K)
        CALL AWRITE(I)
50      CONTINUE
51      CALL NORY('MORE CHANGES? ',IGO)
        IF(IGO.GT.0)GO TO 81
        IF(LWAVE)CALL WSORT(NLINES)
        RETURN
        END
```

```
      SUBROUTINE CTAPE


      DIMENSION TCONC(15,14),ISYMB(14)
      LOGICAL*1 L
      REAL*8 TEMP(105)
      COMMON /RECORD/NSYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      COMMON /IO/INPUT,LOG
      EQUIVALENCE(TEMP,TCONC)
      DATA IEND/'EN'/,NULL/'  '/
      CALL NORY('DO YOU WANT TO PREPARE A CALIBRATION FILE? ',
     1  IGO)
      IF(IGO)9,10,12
9     CALL DATFIL
      IF(L)RETURN
      GO TO 21
12    CALL GETFIL('NAME OF FILE TO CONTAIN CALIBRATION
      1   DATA? ',3,'NEW','NC',L)
      IF(L)RETURN
      DEFINE FILE 3(0,94,U,NREC3)
      CALL ASK('HOW MANY STANDARDS? ','I',NSTD,REAL,L)
      IF(L)GO TO 420
      CALL ASK('# OF LINES/ELEMENT? ','I',NPRIO,REAL,L)
      IF(L)GO TO 420
      CALL ASSIGN(9,'DX0:DYOLF.DAT',0,'SCR')
      DEFINE FILE 9(0,434,U,NREC9)
      CALL ASSIGN(8,'DX0:LINES.MAS',0,'RDO')
      DEFINE FILE 8(267,6,U,NREC8)
      CALL ITYPE('ENTER ELEMENT NAMES AND CONCENTRATIONS ')
      NREC9=1
101   DO 1 I=1,105
1     TEMP(I)=0.0D0
      DO 17 I=1,14
      READ(INPUT,18)ISYMB(I),(TCONC(J,I),J=1,NSTD)
18    FORMAT(A2,1X,15G15.0)
      IF(ISYMB(I).EQ.IEND)GO TO 3
17    IF(I.EQ.14)WRITE(9'NREC9)ISYMB,TCONC
      GO TO 101
3     ISYMB(I)=NULL
      WRITE(9'NREC9)ISYMB,TCONC
      CALL LINES(ISYMB,TCONC,NPRIO)
      CALL CLOSE(8)
      CALL CLOSE(9)
      IF(L)RETURN
      CALL CLOSE(3)
      CALL GETFIL(' ',3,'OLD','NC',L)
      DEFINE FILE 3(500,94,U,NREC3)
```

```
21      CALL TOUT
        IF(L)RETURN
        CALL NORY('DO YOU WANT TO MAKE ANY CHANGES? ',IGO)
        IF(IGO.LE.0)RETURN
        CALL CHANGE
        IF(L)RETURN
        GO TO 21
420     CALL ERROR('CTAPE')
10      RETURN
        END



        FUNCTION CURRNT(MAN)



C
C       FUNCTION TO CONVERT THE NUMBER OF COUNTS IN THE C/F
C       INTO PHOTOCURRENTS. LASCUR IS THE INTEGER*4 VARIABLE
C       WHICH CONTAINS THE NUMBER OF COUNTS.
C
        COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
        LOGICAL*1 MAN
        INTEGER*4 LASCUR
1       CALL CD(LASCUR)   !GET COUNTS
        CURRNT=AJFLT(LASCUR)/TIMEAS/10.**MRNG
        IF(MRNG.EQ.14)CURRNT=CURRNT+OFFSET
        IF(MAN) RETURN
        IF((CURRNT-4.9E-8).GE.0.)GO TO 5
        IF(MRNG.EQ.14)RETURN
        CALL RANGE(14)
        GO TO 1
5       IF(MRNG.EQ.12)RETURN
        CALL RANGE(12)
        GO TO 1
        END
```

```
       FUNCTION CURWAV



C      A FUNCTION TO CONVERT THE BCD WAVELENGTH
C      DISPLAY READING TO A REAL NUMBER
C      THE CURRENT DISPLAY READING IS STORED AS CURLOC
C      AND HAS THE FOLLOWING FORMAT
C      IN THE FIRST WORD ARE THE 4 MOST SIGNIFICANT DIGITS
C      IN THE SECOND WORD ARE THE 2 LEAST SIGNIFICANT
C      DIGITS OF THE DISPLAY IN THE LEFT BYTE AND
C      THE INTEGER # OF STEPS IN THE RIGHT BYTE
C
       EXTERNAL CURLOC
       LOGICAL*1 LBYTE(8)
       INTEGER*2 DIG(6),BYTE(4)
       EQUIVALENCE (BYTE(1),LBYTE(1))
       K=IADDR(CURLOC)
C
C      GET 4 BYTES OF CURLOC AND STORE AS INTEGERS
C
1      DO 10 I=1,4
       BYTE(I)=IPEEKB(K+I-1)
10     LBYTE(2*I)=.FALSE.        !CLEAR LEFT BYTE OF BYTE(I)
C
C      CONVERT EACH BYTE INTO TWO DIGITS
C
       DIG(6)=BYTE(2)/16
       DIG(5)=MOD(BYTE(2),16)
       DIG(4)=BYTE(1)/16
       DIG(3)=MOD(BYTE(1),16)
       DIG(2)=BYTE(4)/16
       DIG(1)=MOD(BYTE(4),16)
C
C      CALCULATE WAVELENGTH EQUIVALENT OF #STEPS
C
       IF(IPEEKB(BYTE(3)).EQ."377)BYTE(3)=0
       IF(BYTE(3).GE.100)BYTE(3)=BYTE(3)-100
       CURWAV=0.0001*BYTE(3)
C
C      CALCULATE WAVELENGTH BY MULTIPLYING
C      EACH DIGIT BY THE APPROPRIATE VALUE
C
       DO 20 I=1,6
20     CURWAV=CURWAV+DIG(I)*10.**(I-3)
C
C      CHECK FOR A VALID VALUE
C
       IF(CURWAV.GE.0.AND.CURWAV.LT.10000.)GO TO 30
```

```
C
C     NOT VALID READ DISPLAY AND CONVERT AGAIN
C
      CALL SETLOC
      GO TO 1
30    IF(CURWAV.GT.9900.)CURWAV=CURWAV-10000.
      RETURN
      END


      SUBROUTINE DATFIL


      LOGICAL*1  L
      COMMON /RECORD/DUMMY(47),L
      CALL GETFIL('NAME OF FILE THAT CONTAINS ELEMENT
     1   DATA? ',3,'OLD','NC',L)
      IF(L)RETURN
      DEFINE FILE 3(500,94,U,NREC3)
      RETURN
      END


      SUBROUTINE DATGET


      LOGICAL*1 L,STRING,BLANK,ENDIT,LBKG
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
     1  CINTEN(16),B(4),C(4),D(4),L
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      EQUIVALENCE(STANO,STRING)
      DATA BLANK,ENDIT/'B','E'/
      LBKG=.FALSE.
      NSTD=KOUNT(1)
      CALL OUTPUT('O')
      CALL HEAD('CALIBRATION ROUTINE')
      CALL ASK('INTEGRATION TIME? ','R',INT,SEC,L)
      IF(L)GO TO 10
      CALL PEAKIT
      IF(L)GO TO 10
15    CALL ASK('STANDARD NUMBER? ','C',INT,STANO,L)
      IF(L)GO TO 100
```

```
      IF(STRING.EQ.BLANK)GO TO 49
      IF(STRING.EQ.ENDIT)GO TO 100
      DECODE(LEN(STANO),14,STANO,ERR=490)J
14    FORMAT(I4)
      IF(J.GT.0 .AND. J.LE.NSTD)GO TO 50
      TYPE 140,NSTD
140   FORMAT('0WARNING!!!   STANDARD NUMBER MUST BE BETWEEN
      1 1 AND '1X,I2)
      GO TO 15
49    J=16
      LBKG=.TRUE.
50    CALL ZERO
      IF(L)GO TO 100
      CALL BIGSUB(J)
      IF(.NOT. L)GO TO 15
100   IF(.NOT.LBKG)GO TO 10
      DO 200 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)RETURN
      DO 205 N=1,15
205   IF(CINTEN(N).NE.0.)CINTEN(N)=CINTEN(N)-CINTEN(16)
      CINTEN(16)=0.0
200   CALL AWRITE(I)
10    IF(L)CALL ERROR('DATGET')
      RETURN
490   CALL ITYPE('INPUT ERROR!')
      GO TO 15
      END


      SUBROUTINE DELAY(IT,SEC)


C
C     SUBROUTINE TO DELAY EXECUTION FOR A PERIOD OF TIME
C     SPECIFIED BY SEC.
C
      INTEGER*4 JT,IT,H24
      DIMENSION I24(2)
      EQUIVALENCE(H24,I24)
      DATA I24/"15000,"117/       ! 24 HOURS
      IW=IFIX(SEC*60.)
      CALL JJCVT(IT)
10    CALL GTIM(JT)
      CALL JJCVT(JT)
      IF(JSUB(JT,IT,JT).LT.0)CALL JADD(JT,H24,JT)
```

```
      IF(JT.LT.IW)GO TO 10
      RETURN
      END




      SUBROUTINE DETLIM




      LOGICAL*1 L,LESCAP
      COMMON /IO/INPUT,LOG
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      COMMON /RECORD/ISYMB,WAVE,CORR1,ICOR(3),CONC(15),
     1  IDUM(56),L
      CALL OUTPUT('O')
      CALL HEAD('DETECTION LIMIT')
      CALL DATFIL
      IF(L)RETURN
      CALL PEAKIT
      IF(L)GO TO 12
10    CALL ASK('STANDARD #? ','I',J,REAL,L)
      IF(L)GO TO 12
      CALL ASK('#REPEATS? ','I',NRPTS,REAL,L)
      IF(L) GO TO 12
      CALL ASK('#SECS/RPT? ','R',INT,SEC,L)
      IF(L) GO TO 12
      CALL NORY('PRINT BACK EACH REPEAT? ',NPRINT)
4     CALL ZERO
      IF(L)GO TO 12
      TYPE 107
      IF(LOG.NE.0)WRITE(LOG,107)
107   FORMAT('0ELEM   WAVE   DET. LIM.     CONC         NET
      1        BKG            STD BKG')
      DO 101 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.0)GO TO 102
      IF((WAVE-CORR1).LT.0.0)GO TO 1
2     CALL SEEK(CORR1)
      CALL TAKEDA(NRPTS,BKG,STDBKG,NPRINT)
      IF(LESCAP())GO TO 12
      IF((WAVE-CORR1).LT.0.0)GO TO 3
1     CALL ROBYN(STAND)
      IF(L)GO TO 12
      IF((WAVE-CORR1).LT.0.0)GO TO 2
3     DL=3.0*CONC(J)*BKG*STDBKG/100.0/(STAND-BKG)
      TYPE 108,ISYMB,WAVE,DL,CONC(J),STAND-BKG,BKG,STDBKG
      IF(LOG.NE.0)WRITE(LOG,108)ISYMB,WAVE,DL,CONC(J),
```

```
     1 STAND-BKG,BKG,STDBKG
108   FORMAT(1X,A2,F10.4,1PE12.3,G12.3,1PE12.3,1PE12.3,
     1  0PF11.2)
101   CONTINUE
102   CALL NORY('REPEAT EXPERIMENT? ',IGO)
      IF(IGO.GT.0)GO TO 4
      CALL NORY('CHANGE CONCENTRATION? ',IGO)
      IF(IGO.GT.0)GO TO 10
12    CALL ESCAPE(.FALSE.)
      CALL OUTPUT('C')
      CALL CLOSE(3)
      RETURN
      END



      SUBROUTINE DIRECT



C
C     READ THE DIRECTORY OF A SPECIFIED DEVICE
C
      LOGICAL*1 L,NAME(11),CHARS(4)
      INTEGER*2 COUNT,UNUSED,EXTEN,DATE,STATUS
      REAL*4 MONTH(13),DEVICE(2)
      DATA MONTH/'???-','JAN-','FEB-','MAR-','APR-','MAY-',
      1  'JUN-','JUL-','AUG-','SEP-','OCT-','NOV-','DEC-'/
      DATA DEVICE/'DX0:','DX1:'/
      COUNT=0
      LINK=1
      UNUSED=0
      CALL ASK('DEVICE(0 OR 1)? ','I',INT,REAL,L)
      IF(L)RETURN
      CALL ASSIGN(8,DEVICE(INT+1),4,'RDO','NC')
      DEFINE FILE 8(4000,1,U,NREC)
3     NREC=6*256+2+(LINK-1)*512
      READ(8'NREC)LINK
C
C     READ AND PRINT THE DIRECTORY
C
      NREC=NREC+3
10    READ(8'NREC)STATUS                !STATUS WORD FOR ENTRY
      STATUS=STATUS/256
      IF(STATUS.EQ.8)GO TO 200            !END OF SEGMENT
      READ(8'NREC)NAME1                 !1ST 3 CHARACTERS
      READ(8'NREC)NAME2                 !NEXT 3 CHARACTERS
      READ(8'NREC)EXTEN                 !FILE EXTENSION
```

```
      READ(8'NREC)LENGTH              !LENGTH OF FILE
      READ(8'NREC)JOB                 !JOB STATUS INFORMATION
      READ(8'NREC)DATE
C
      IDAY=0
      IMON=0
      IYR=0
      IF(DATE.EQ.0)GO TO 31
      DATE=DATE-70
      IYR=78+MOD(DATE,32)
      DATE=DATE/32+2
      IDAY=MOD(DATE,32)
      IMON=DATE/32
C
C     INTERPRET STATUS CODE
C
31    IF(STATUS.EQ.1)GO TO 100          !TENTATIVE FILE
      IF(STATUS.EQ.2)GO TO 150          !EMPTY FILE
      IF(STATUS.NE.4)GO TO 400          !NORMAL STATUS IS 4
C
C     UNPACK FILE NAME
C
      CALL R50ASC(3,NAME1,CHARS)      !CONVERT RAD50 TO ASCII
      CALL SCOPY(CHARS,NAME,3)
      CALL R50ASC(3,NAME2,CHARS)
      CALL CONCAT(NAME,CHARS,NAME,6)
      CALL CONCAT(NAME,'.',NAME,7)
      CALL R50ASC(3,EXTEN,CHARS)
      CALL CONCAT(NAME,CHARS,NAME,10)
C
C     PRINT DIRECTORY ENTRY
C
      TYPE 25,NAME,LENGTH,IDAY,MONTH(IMON+1),IYR
25    FORMAT(' ',11A1,4X,I5,4X,I2,'-',A4,I2)
      COUNT=COUNT+1
      GO TO 10
C
C     OTHER DIRECTORY ENTRIES
C
100   TYPE 105,LENGTH                        !STATUS CODE 1
105   FORMAT(' <TENTATIVE>',4X,I4)
      UNUSED=UNUSED+LENGTH
      GO TO 10
C
150   TYPE 155,LENGTH                        !STATUS CODE 2
155   FORMAT('   <EMPTY> ',4X,I5)
      UNUSED=UNUSED+LENGTH
      GO TO 10
C
200   IF(LINK.NE.0)GO TO 3
```

```
      TYPE 250,COUNT,UNUSED
250   FORMAT('0',I4,' FILES ',I6,' FREE BLOCKS')
      CALL CLOSE(8)
      RETURN
C
400   TYPE 405,STATUS
405   FORMAT('0ERROR***** ILLEGAL STATUS: ',I6)
      CALL CLOSE(8)
      RETURN
C
      END
```

```
      SUBROUTINE ELIM



      DIMENSION SCONC(5,15),NSYMB(5,15),ITEMP(5,15)
      LOGICAL*1 L
      COMMON /RECORD/IDUM(94),L
16    DO 500 I=1,15
      DO 500 J=1,5
500   SCONC(J,I)=0.0
      CALL GETSYM(NSYMB,SCONC,ITEMP,.FALSE.)
      IF(L)RETURN
      CALL FIT(.TRUE.)
      IF(L)RETURN
      CALL NORY('RESTORE ANY POINTS? ',IGO)
      IF(IGO.LE.0)GO TO 101
      CALL GETSYM(ITEMP,SCONC,NSYMB,.TRUE.)
      IF(L)RETURN
      CALL FIT(.TRUE.)
      IF(L)RETURN
101   CALL NORY('MORE CHANGES? ',IGO)
      IF(IGO.GT.0)GO TO 16
      RETURN
      END
```

```
      SUBROUTINE ENDPLT



C
C     PUT CRT TERMINAL BACK INTO TYPING MODE
C
      IMPLICIT LOGICAL*1 (L)
      DATA LUS/"37/
      CALL PLOT(0.,767.,0)  !MOVE CURSOR TO LEFT TOP CORNER
      TYPE 100,LUS
100   FORMAT('+',A1$)
      RETURN
      END



      SUBROUTINE ERROR(LQ)



      LOGICAL*1 LQ(1),L
      COMMON /RECORD/DUMMY(47),L
      CALL ESCAPE(.FALSE.)
      TYPE 1,LQ(1)
1     FORMAT('0ERROR OCCURED IN SUBROUTINE ',A1$)
      TYPE 2,(LQ(I),I=2,LEN(LQ))
2     FORMAT('+',A1$)
      L=.TRUE.
      RETURN
      END



      SUBROUTINE ESCAPE(ENABLE)



C
C     SUBROUTINE TO ENABLE OR DISABLE ESCAPE
C     WHEN NO I/O IS PENDING
C     ENABLE=.TRUE. TO ENABLE ESCAPE
C
      LOGICAL*1 ENABLE
      INTEGER*2 TTINSR,ION,IOFF
```

```
C
C     TTINSR IS THE TTY KEYBOARD STATUS REGISTER
C
      DATA TTINSR,ION,IOFF/"177560,"100,"177677/
C
C     TURN KEYBOARD INTERRUPT ON - DISABLES ESCAPE
C
      IF(.NOT.ENABLE)CALL IPOKE(TTINSR,IPEEK(TTINSR).OR.ION)
C
C     TURN KEYBOARD INTERRUPT OFF - ENABLES ESCAPE
C
      IF(ENABLE)CALL IPOKE(TTINSR,IPEEK(TTINSR).AND.IOFF)
      RETURN
      END



      FUNCTION FILCHK(FNAME)



C
C     FUNCTION TO CHECK DIRECTORY OF DISK TO SEE IF A FILE IS
C     PRESENT. FUNCTION IS .TRUE. IF PRESENT, .FALSE. IF NOT
C
      LOGICAL*1 FILCHK,FNAME(1),LNAME(10),DECPT,BLANK
      LOGICAL*1 CHARS(4),NULL,POINT(2)
      INTEGER*2 STATUS,EXTEN
      EQUIVALENCE (POINT(1),DECPT),(POINT(2),NULL)
      DATA DECPT,BLANK,NULL/'.',' ',"0/
      IDEC=INDEX(FNAME,DECPT)
      IF(IDEC.GT.0)GO TO 8
      K=LEN(FNAME)+1
      FNAME(K)=DECPT
      DO 9 I=K+1,K+3
9     FNAME(I)=BLANK
      FNAME(K+4)=NULL
      GO TO 12
8     DO 11 I=IDEC+1,IDEC+3
11    IF(FNAME(I).EQ.NULL)FNAME(I)=BLANK
      FNAME(IDEC+4)=NULL
12    LINK=1
      DO 7 I=1,7
      IF(FNAME(I).EQ.DECPT)GO TO 1
7     LNAME(I)=FNAME(I)
1     IF(I.EQ.7)GO TO 2
      DO 4 J=I,6
4     LNAME(J)=BLANK
```

```
2      DO 6 J=7,9
       I=I+1
6      LNAME(J)=FNAME(I)
       CALL ASSIGN(8,'DX1:',4,'RDO','NC')
       DEFINE FILE 8(4000,1,U,NREC)
3      NREC=6*256+2+(LINK-1)*512
       READ(8'NREC)LINK
       NREC=NREC+3
10     READ(8'NREC)STATUS
       STATUS=STATUS/256
       IF(STATUS.EQ.8)GO TO 200
       IF(STATUS.EQ.4)GO TO 15
       NREC=NREC+6
       GO TO 10
15     READ(8'NREC)NAME1
       CALL R50ASC(3,NAME1,CHARS)
       DO 21 I=1,3
       IF(CHARS(I).EQ.LNAME(I))GO TO 21
       NREC=NREC+5
       GO TO 10
21     CONTINUE
       READ(8'NREC)NAME2
       CALL R50ASC(3,NAME2,CHARS)
       DO 22 I=1,3
       IF(CHARS(I).EQ.LNAME(I+3))GO TO 22
       NREC=NREC+4
       GO TO 10
22     CONTINUE
       READ(8'NREC)EXTEN
       CALL R50ASC(3,EXTEN,CHARS)
       DO 23 I=1,3
       IF(CHARS(I).EQ.LNAME(I+6))GO TO 23
       NREC=NREC+3
       GO TO 10
23     CONTINUE
       FILCHK=.TRUE.
       CALL CLOSE(8)
       RETURN
200    IF(LINK.NE.0)GO TO 3
       FILCHK=.FALSE.
       CALL CLOSE(8)
       RETURN
       END
```

```
      SUBROUTINE FINDPK


      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IDUM(87),L
      REAL*4 LO,HI,CURDAT(31),CURMAX(5)
      LOGICAL*1 L,LESCAP
      DO 1 I=1,5
1     CURMAX(I)=0.0
      CWAVE=WAVE
      DO 5 K=1,31
      CALL SEEK(CWAVE-(0.15-(K-1)*0.01))
      IF(LESCAP())CALL ERROR('FINDPK')
      IF(L)RETURN
5     CURDAT(K)=CURRNT(.FALSE.)
      DO 10 K=1,2
      CURTM1=CURDAT(1)
      DO 10 I=2,30
      CURTM2=CURDAT(I)
      CURDAT(I)=(CURTM1+CURTM2+CURDAT(I+1))/3.
10    CURTM1=CURTM2
      NEND=4
      NPKS=1
21    K=0
      DO 25 I=NEND,28
      IF(SLOPE(CURDAT(I-3)).LE.0.)GO TO 24
      K=K+1
      IF(K.LT.5)GO TO 25
      NSTART=I-4
      GO TO 26
24    K=0
25    CONTINUE
      RETURN
26    K=0
      DO 30 I=NSTART,28
      IF(SLOPE(CURDAT(I-3)).GT.0.)GO TO 29
      K=K+1
      IF(K.LT.5)GO TO 30
      NPEAK=I-4
      GO TO 31
29    K=0
30    CONTINUE
      RETURN
31    HALFPK=CURDAT(NSTART)+(CURDAT(NPEAK)-CURDAT(NSTART))/2.
      DO 35 I=NPEAK,NSTART,-1
      IF(CURDAT(I).LE.HALFPK)GO TO 36
35    CONTINUE
36    LO=(HALFPK-CURDAT(I))/(CURDAT(I+1)-CURDAT(I))+I
      DO 40 I=NPEAK,31
```

```
      IF(CURDAT(I).LE.HALFPK)GO TO 41
40    CONTINUE
      RETURN
41    HI=(CURDAT(I-1)-HALFPK)/(CURDAT(I-1)-CURDAT(I))+I-1.
      PEAK=LO+(HI-LO)/2.
      NPKS=NPKS+1
      NEND=NPEAK+4
      CURMAX(NPKS)=CURDAT(IFIX(PEAK))
      IF(CURMAX(NPKS).LT.CURMAX(NPKS-1))GO TO 21
      WAVE=CWAVE-(0.15-0.01*(PEAK-1.))-0.005
      CORR1=WAVE+CORR2
      GO TO 21
      END



      SUBROUTINE FIT(MODE)



      DIMENSION W(15)
      LOGICAL*1 L,MODE
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      COMMON /IO/INPUT,LOG
20    CALL ASK(' DEGREE OF FIT(0<=N<=3)? ','I',NTERMS,REAL,L)
      IF(L)GO TO 40
      IEXP=0
      CALL NORY('DO YOU WANT TO WEIGHT YOUR DATA BY
      1 INTENSITY**-2? ',IGO)
      IF(IGO.GT.0)IEXP=2
      CALL IDATE(IM,ID,IY)
      TYPE 780,IM,ID,IY,NTERMS+1
      IF(LOG.NE.0)WRITE(LOG,780)IM,ID,IY,NTERMS+1
780   FORMAT(' CALIBRATION CURVES OF ',I2,'/',I2,'/',I2/
      1 ' # OF TERMS=',I2/'0ELEMENT      WAVE       %RMS')
      DO 710 I=1,500
      NCOUNT=0
      CALL AREAD(I)
      IF(WAVE.EQ.0.)GO TO 30
      IF(MODE .AND. CINTEN(16).GE.0.0)GO TO 710
      DO 600 K=1,15
600   IF(CONC(K).GT.0. .AND. CINTEN(K).NE.0.)NCOUNT=NCOUNT+1
      DO 620 K=1,15
      W(K)=1.0
620   IF(CINTEN(K).NE.0. .AND. IEXP.NE.0)
     1  W(K)=1./CINTEN(K)**IEXP
      NCURVE=NTERMS+1
```

```
      CALL ORTPOL(W,NCURVE)
      RMS=0.0
      DO 635 K=1,15
      IF(CONC(K).LE.0. .OR. CINTEN(K).EQ.0.)GO TO 635
      YCALC=ORTVAL(CINTEN(K),NCURVE)
      RMS=RMS+((CONC(K)-YCALC)/CONC(K))**2
635   CONTINUE
      RMS=SQRT(RMS/FLOAT(NCOUNT))*100.
      TYPE 720,ISYMB,WAVE,RMS
      IF(LOG.NE.0)WRITE(LOG,720)ISYMB,WAVE,RMS
720   FORMAT(' ',2X,A2,3X,F11.4,F12.4)
      CINTEN(16)=0.0
      CALL AWRITE(I)
710   CONTINUE
30    CALL NORY('CHANGE THE FIT? ',IGO)
      IF(IGO.GT.0)GO TO 20
      RETURN
40    CALL ERROR('CURVE')
      RETURN
      END


      SUBROUTINE GDATA(X,AMP)


      LOGICAL*1 L
      COMMON /RECORD/ISYMB,WAVE,CORR1,IDUM(89),L
      IF((WAVE-CORR1).GT.0.0)CALL BKG(AMP)
      IF(L)RETURN
      CALL ROBYN(X)
      IF(L)RETURN
      IF((WAVE-CORR1).LT.0.0)CALL BKG(AMP)
      IF(L)RETURN
      X=X-AMP
      RETURN
      END
```

```
      SUBROUTINE GETDAT(MODE,N)



      LOGICAL*1 MODE,DATLAB(15,100),T,GLAB(20),XLAB(16)
      LOGICAL*1 YLAB(18)
      COMMON /POINTS/X(100),Y(100),IPRPLT(10),NPTS
      COMMON /XTRM/XMAX,XMIN,YMAX,YMIN,XSCALE,YSCALE
      COMMON /LABELS/ISYMB(10),T(12),GLAB,XLAB,YLAB,DATLAB
      K=IPRPLT(N)
      IF(.NOT.MODE)GO TO 101
      CALL SYMBOL(4.5,4.85-N*0.15,0.11,DATLAB(1,K),
     1     0.0,LEN(DATLAB(1,K)))
      CALL SYMBOL(6.4,4.85-N*0.15,0.11,ISYMB(N),0.0,1)
101   IK=(K-1)*NPTS+1
      IJ=IK+NPTS-1
      I=1
      DO 102 J=IK,IJ
      READ(3'J)X(I),Y(I)
      IF(.NOT.MODE)GO TO 102
      X(I)=(X(I)-XMIN)/XSCALE
      Y(I)=(Y(I)-YMIN)/YSCALE
102   I=I+1
      RETURN
      END



      SUBROUTINE GETFIL(Q,LUNIT,TYPE,CTRL,L)



C
C     Q IS VARIABLE LENGTH ASCII STRING
C     LUNIT IS LOGICAL UNIT # OF FILE TO BE OPENED
C     TYPE IS THE TYPE OF FILE (NEW OR OLD)
C     CTRL IS THE CONTROL PARAMETER (CC OR NC)
C
      LOGICAL*1 FILCHK,Q(1),TYPE(1),CTRL(1),L,STRING,NULL
      LOGICAL*1 FIRST,NEW,OLD,ESC,FILNAM(11)
      COMMON /IO/INPUT,LOG
      COMMON /FILES/STRING(17)
      EQUIVALENCE (FIRST,STRING(5),FILNAM)
      DATA ESC/"33/,NEW/'N'/,OLD/'O'/,NULL/"0/
      IF(LEN(Q).EQ.1)GO TO 10
6     CALL ITYPE(Q)        !TYPE QUESTION ON TERMINAL
      DO 5 I=1,11
```

```
5       FILNAM(I)=NULL
        CALL GETSTR(INPUT,FILNAM,10,L)   !GET FILENAME
        L=FIRST.EQ.ESC       !WAS ALTMODE TYPED?
        IF(L)RETURN    !YES-THEN RETURN
10      IF(TYPE(1).EQ.OLD)GO TO 1
        IF(.NOT.FILCHK(FILNAM))GO TO 3
        TYPE 2,FILNAM
2       FORMAT('0',11A1,' ALREADY EXISTS. ')
        CALL NORY('REPLACE? ',IGO)
        IF(IGO)6,6,3
1       IF(FILCHK(FILNAM))GO TO 3
        TYPE 4,FILNAM
4       FORMAT('0',11A1,' IS NOT ON DISK. ')
        GO TO 6
3       CALL ASSIGN(LUNIT,STRING,0,TYPE,CTRL)
        RETURN
        END



        SUBROUTINE GETNAM(LARRAY)



        LOGICAL*1 LARRAY(40),L
        COMMON /IO/INPUT,LOG
        CALL ITYPE('SOLUTION NAME? ')
        CALL GETSTR(INPUT,LARRAY,39,L)
        RETURN
        END



        SUBROUTINE GETSYM(IK,SCONC,JK,LGO)



        DIMENSION IK(5,15),SCONC(5,15),JK(5,15)
        LOGICAL*1 L,LGO,LANS(15),NO,ANS
        COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
        1  CINTEN(16),B(4),C(4),D(4),L
        DATA NULL/'  '/,NO/'N'/
        DO 1 I=1,15
        DO 1 J=1,5
1       IK(J,I)=NULL
        TYPE 5
```

```
5     FORMAT(' WHICH STANDARD?(ANSWER T OR F) '/
      1  ' ',9X,'111111'/' ','123456789012345')
      ACCEPT 10,LANS
10    FORMAT(15L1)
      DO 15 I=1,15
      IF(.NOT.LANS(I))GO TO 15
      TYPE 20,I
20    FORMAT(' ELEMENTS FOR STANDARD #',I2)
      ACCEPT 25,(IK(K,I),K=1,5)
25    FORMAT(10(A2,1X))
15    CONTINUE
      DO 30 I=1,15
      IF(.NOT.LANS(I))GO TO 30
      DO 35 K=1,500
      CALL AREAD(K)
      IF(WAVE.EQ.0.)GO TO 30
      DO 35 J=1,5
      IF(ISYMB.NE.IK(J,I))GO TO 35
      TYPE 2,I,ISYMB,WAVE
2     FORMAT('$ELIMINATE (OR RESTORE) STANDARD #'
      1  ,I2,' FOR ',A2,F11.4,2X)
      ACCEPT 3,ANS
3     FORMAT(A1)
      IF(ANS.EQ.NO)GO TO 35
      CINTEN(16)=-1.0
      IF(LGO)GO TO 48
      SCONC(J,I)=CONC(I)
      CONC(I)=-1.
      CALL AWRITE(K)
      GO TO 35
48    DO 51 M=1,5
      IF(IK(J,I).NE.JK(M,I))GO TO 51
      CONC(I)=SCONC(M,I)
      CALL AWRITE(K)
51    CONTINUE
35    CONTINUE
30    CONTINUE
      RETURN
      END
```

SUBROUTINE GRAPH

```
C
C     SUBROUTINE FOR PLOTTING CALIBRATION CURVES ON CRT.
C
      LOGICAL*1 L,LQ(4),BLANK
      DIMENSION NSYMB(2)
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      EQUIVALENCE(SYMB,NSYMB(1),LQ(1)),(BLANK,NULL)
      DATA NULL/'  '/
      NTERMS=KOUNT(0)
5     CALL NORY('DO YOU WANT TO GRAPH EVERY LINE? ',IGO)
      IF(IGO.LE.0)GO TO 100
      ITEST=0
      NSYMB(1)=NULL
15    DO 50 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)GO TO 90
      IF(ITEST.EQ.1 .AND. ISYMB.NE.NSYMB(1))GO TO 50
      PAUSE 'TYPE (CR) TO BEGIN.'
      YMIN=1.E30
      YMAX=0.0
      DO 40 J=1,15
      IF((CONC(J).LE.0.) .OR. (CINTEN(J).EQ.0.0))GO TO 40
      YMIN=AMIN1(YMIN,CINTEN(J))   !FIND MINIMUM INTENSITY
      YMAX=AMAX1(YMAX,CINTEN(J))   !FIND MAXIMUM INTENSITY
40    CONTINUE
      CALL LABEL
      IF(YMAX.LE.0.0)GO TO 901
      YMAX=YMAX*1.5
      YCON=0.0
      IF(YMIN.GT.0.)GO TO 45
      YCON=-YMIN
      YMIN=1.E-14
45    YMIN=YMIN/1.5
      YVAL=YMIN
      YINC=2.*ALOG10(YMAX/YMIN)/50.
      DO 600 K=1,100
      XPT=ORTVAL(YVAL+YCON,NTERMS)
      IF(XPT.GT.0.0)GO TO 601
      YINC=YINC*0.986
600   YVAL=YMIN*10.0**(K*YINC)
601   YMIN=YVAL
      YMIN1=ALOG10(YMIN+YCON)
      XMAX=ORTVAL(YMAX+YCON,NTERMS)
      XMIN=ALOG10(ORTVAL(YMIN+YCON,NTERMS))
```

```
      XSCALE=(ALOG10(XMAX)-XMIN)/7.0
      YSCALE=(ALOG10(YMAX+YCON)-YMIN1)/5.0
      YINC=2.*ALOG10(YMAX/YMIN)/50.
      IC=3
      YVAL=YMIN
      DO 60 K=1,200
      XPT=ORTVAL(YVAL+YCON,NTERMS)
      XPT=(ALOG10(XPT)-XMIN)/XSCALE
      YPT=(ALOG10(YVAL+YCON)-YMIN1)/YSCALE
      CALL PLOT(XPT,YPT,IC)
      IC=2
      YINC=YINC*.986
      IF(YVAL.GT.YMAX)GO TO 55
60    YVAL=YMIN*10.**(K*YINC)
55    DO 70 J=1,15
      IF((CONC(J).LE.0.) .OR. (CINTEN(J).EQ.0.))GO TO 70
      YC=AMAX1(1.E-13,CINTEN(J))
      YC=(ALOG10(YC+YCON)-YMIN1)/YSCALE
      XC=(ALOG10(CONC(J))-XMIN)/XSCALE
      CALL PLOT(XC-0.05,YC-0.05,3)
      CALL PLOT(XC+0.05,YC+0.05,2)
      CALL PLOT(XC-0.05,YC+0.05,3)
      CALL PLOT(XC+0.05,YC-0.05,2)
70    CONTINUE
      CALL ENDPLT
      TYPE 46,ISYMB,WAVE,IPRIOR
46    FORMAT(' ',1X,A2,1X,F11.4,1X,I2)
      GO TO 50
901   CALL ENDPLT
      TYPE 126,ISYMB,WAVE,IPRIOR
126   FORMAT(' THIS GRAPH ('A2,1X,F9.4,1X,I2,')  REQUIRED'/
      1  ' TAKING THE LOG OF A NEGATIVE NUMBER.I AM SKIPPING'/
      1  ' THIS ONE AND GOING TO THE NEXT ONE')
50    CONTINUE
90    CALL NORY('MORE GRAPHS? ',IGO)
      IF(IGO.GT.0)GO TO 5
120   RETURN
100   CALL ASK('ELEMENT TO GRAPH? ','C',INT,SYMB,L)
      IF(L)RETURN
      IF(LEN(LQ).LT.2)CALL IPOKEB(IADDR(LQ)+1,BLANK)
      ITEST=1
      GO TO 15
      END
```

```
      SUBROUTINE HEAD(Q)



C
C     SUBROUTINE TO WRITE A HEADING WHEN
C     ENTERING A ROUTINE
C
      COMMON /IO/INPUT,LOG
      LOGICAL*1 Q(1),DATTIM(20)
      DATA DATTIM(1),DATTIM(11),DATTIM(20)/' ',' ',' '/
      IF(LOG.EQ.0)RETURN
      CALL TIME(DATTIM(12))
      CALL DATE(DATTIM(2))
      WRITE(LOG,100)Q(1)
100   FORMAT('0',A1$)
      WRITE(LOG,101) (Q(K),K=2,LEN(Q))
101   FORMAT('+',A1$)
      WRITE(LOG,102) (DATTIM(K),K=1,20)
102   FORMAT('+',20A1)
      RETURN
      END



      SUBROUTINE INCDAT



      INTEGER*2 DAYS(12),DATE
      DATA DAYS/31,28,31,30,31,30,31,31,30,31,30,31/
      IADATE=IPEEK("54)+"262      !SYSTEM DATE IS STORED HERE
      DATE=IPEEK(IADATE)
      DATE=DATE-70
      IYR=78+MOD(DATE,32)
      DATE=DATE/32+2
      IDAY=MOD(DATE,32)
      IMON=DATE/32
      IDAY=IDAY+1
      IF(MOD(IYR,4).EQ.0)DAYS(2)=29
      IF(IDAY.LE.DAYS(IMON))GO TO 30
      IMON=IMON+1
      IDAY=1
      IF(IMON.LE.12)GO TO 30
      IYR=IYR+1
      IMON=1
30    CALL IPOKE(IADATE,((IMON*32+(IDAY-2))*32+(IYR-8)))
```

```
      RETURN
      END



      SUBROUTINE INIT



C
C     SUBROUTINE TO INITIALIZE I/O DEVICES,SET UP C/F
C     MEASUREMENT PERIOD, AND MEASURE C/F OFFSET.
C
      LOGICAL*1 L,NULL,DECPT,LESCAP,ALPHA(8),DATSTR
      REAL*4 ALPHAB(2)
      EQUIVALENCE (ALPHA(1),ALPHAB)
      COMMON /IO/INPUT,LOG
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,INTA(2)
      COMMON /COUNT/ICOUNT
      COMMON /FILES/DATSTR(17)
      DATA NULL/"0/,DECPT/'.'/,ALPHAB/'ABCD','EFGH'/
      CALL HEAD('INITIATE ROUTINE')
      CALL RANGE(14)
      CALL SETLOC
      CALL ASK('MEAS. TIME? ','R',INT,TIMEAS,L)
      IF(L)RETURN
      NT=IFIX(TIMEAS*60.)  !CHANGE TIMEAS INTO CLOCK TICKS
      CALL JICVT(NT+1,INTA)
      CALL NORY('MEASURE C/F OFFSET? ',IGO)
      IF(IGO.LE.0)GO TO 1
      PAUSE 'PLEASE CLOSE SLITS.TYPE (CR) TO BEGIN.'
      SEC=10.0    !INTEGRATION TIME FOR OFFSETS
      CALL TAKEDA(10,OFFSET,STD,0)
      IF(LESCAP())CALL ESCAPE(.FALSE.)
      OFFSET=-OFFSET
1     CALL NORY('KEYBOARD INPUT? ',IGO)
      IF(IGO.LE.0)GO TO 3
      IF(INPUT.EQ.1)CALL CLOSE(1)
      INPUT=5
      GO TO 4
3     IF(INPUT.EQ.1)GO TO 4
      CALL GETFIL('FILE NAME? ',1,'OLD','NC',L)
      INPUT=1
      IF(L)INPUT=5
4     IF(LOG.NE.0)CALL CLOSE(LOG)
      LOG=0
      CALL NORY('LOG OUTPUT? ',IGO)
      IF(IGO.LE.0)RETURN
```

```
      IF(ICOUNT.EQ.8)ICOUNT=0
      ICOUNT=ICOUNT+1
      LOG=2
      CALL TIME(DATSTR(5))
      CALL SCOPY('LOG',DATSTR(5),3)
      CALL DATE(DATSTR(9))
      DATSTR(11)=DECPT
      DO 5 I=15,17
5     DATSTR(I)=NULL
      DATSTR(8)=ALPHA(ICOUNT)
      CALL GETFIL(' ',LOG,'NEW','CC',L)
      RETURN
      END
```

```
        ;
        ;
        ;
        .TITLE BYTERW
        ;
        ;ROUTINES TO READ OR WRITE TO A BYTE FROM FORTRAN
        ;
        ;THE IPEEKB FUNCTION IS SIMILAR TO THE SYSLIB
        ;IPEEK FUNCTION AND THE IPOKEB SUBROUTINE
        ;IS SIMILAR TO THE SYSLIB IPOKE SUBROUTINE
        ;
        ;THE ONLY DIFFERENCE IS THAT IPEEKB AND IPOKEB
        ;ARE BYTE OPERATIONS WHEREAS IPEEK AND IPOKE
        ;ARE WORD OPERATIONS.
        ;
        .MCALL .REGDEF
        .REGDEF
        ;
        ;ROUTINE ENTRY POINT FOR IPEEKB
        ;
        .GLOBL IPEEKB
        ;
IPEEKB:TST (R5)+        ;ADV. POINTER
        MOV @(R5),R0    ;GET ADDRESS
        MOVB (R0),R0    ;GET BYTE
        RTS PC          ;RETURN
        ;
        ;ROUTINE ENTRY POINT FOR IPOKEB
        ;
        .GLOBL IPOKEB
        ;
IPOKEB:TST (R5)+         ;ADV. POINTER
        MOV @(R5)+,R0    ;GET ADDRESS
```

```
       MOVB @(R5),(R0)  ;STORE BYTE
       RTS PC           ;RETURN
      ;
       .END



      SUBROUTINE ITYPE(Q)



      LOGICAL*1 Q(1),CR,LF,QMARK
      DATA CR,LF,QMARK/"15,"12,'?'/
C
C     Q IS AN ASCII STRING
C
      J=LEN(Q)   !HOW LONG IS THE STRING?
      TYPE 5,Q(1)
5     FORMAT(' ',A1$)
      TYPE 10,(Q(K),K=2,J)
10    FORMAT('+',A1$)
      IF(Q(J-1).NE.QMARK)TYPE 10,CR,LF
      RETURN
      END



      SUBROUTINE JOG(DIR,SPEED,NJOGS)



C
C     SUBROUTINE TO RUN THE SPECTROMETER IN JOG
C     MODE IN A GIVEN DIRECTION AT A GIVEN RATE FOR
C     A GIVEN NUMBER OF JOGS(#STEPS=#JOGS*10)
C
C     HIGH SPEED IS 0.05 NM PER JOG
C     LOW SPEED IS 0.001 NM PER JOG
C
      LOGICAL*1 DIR,UP,DOWN,HIGH,LOG,JOGOUT,SPEED
      INTEGER*2 NJOGS,SPOUT
      INTEGER*4 IT
      DATA UP,DOWN,HIGH,LOW,SPOUT/'U','D','H','L',"167772/
      CALL SETWAV(999.)       !SETUP DUMMY TARGET WAVELENGTH
C
C     SET UP BIT 4 FOR THE PROPER DIRECTION
```

```
C
      IF(DIR.EQ.UP)JOGOUT=0
      IF(DIR.EQ.DOWN)JOGOUT=16
C
C     SET UP BITS 6,5,3
C     BIT 6 FOR JOG MODE ENABLE
C     BIT 5 TO TURN THE GO BIT ON
C     BIT 3 FOR HIGH OR LOW SPEED
C
      IF(SPEED.EQ.HIGH)JOGOUT=JOGOUT+104
      IF(SPEED.EQ.LOW)JOGOUT=JOGOUT+96
      CALL IPOKEB(SPOUT,JOGOUT)
      DO 10 I=1,NJOGS
      CALL GTIM(IT)
      CALL DELAY(IT,0.025)
C
C     SET JOG (BIT 7)
C
      CALL IPOKEB(SPOUT,IPEEKB(SPOUT).OR."200)
      CALL DELAY(IT,0.05)
C
C     CLEAR JOG (BIT 7)
C
10    CALL IPOKEB(SPOUT,IPEEKB(SPOUT).AND."177)
      RETURN
      END



      FUNCTION KOUNT(MODE)



      LOGICAL*1 L
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1 CINTEN(16),B(4),C(4),D(4),L
      CALL AREAD(1)
      KOUNT=0
      IF(MODE.EQ.1)GO TO 10
      DO 25 I=1,4
25    IF(D(I).NE.0.)KOUNT=KOUNT+1
      RETURN
10    DO 15 I=1,15
15    IF(CONC(I).NE.0.0)KOUNT=KOUNT+1
      RETURN
      END
```

```
      SUBROUTINE LABEL



C
C     ROUTINE TO PAGE THE CRT AND INITIALIZE IT FOR PLOTTING.
C
      IMPLICIT LOGICAL*1 (L)
      COMMON /PLTORG/ IXORG,IYORG
      DATA LESC,LFF/"33,"14/
      IXORG=0
      IYORG=0
      TYPE 100,LESC,LFF
100   FORMAT('+',2A1$)
      T0=SECNDS(0.0)
1     IF(SECNDS(T0).LE.1.0)GO TO 1
      CALL PLOT(0.,0.,0)
      RETURN
      END



      FUNCTION LESCAP



      IMPLICIT LOGICAL*1 (L)
      INTEGER*2 TTBUF,SPOUT
      DATA LBEEP,TTBUF,SPOUT/"7,"177562,"167772/
C
C     TTBUF IS THE VIDEO INPUT BUFFER
C     SPOUT IS THE SPECTROMETER OUTPUT REGISTER
C
      LESCAP=.FALSE.
      IF(IPEEK(TTBUF).NE."175) RETURN  !"175 IS ALTMODE
      LESCAP=.TRUE.
      CALL IPOKE(SPOUT,IPEEK(SPOUT).AND."177737)   !STOP SCAN
      TYPE 100,(LBEEP,I=1,4)
100   FORMAT('+',4A1$)
      RETURN
      END
```

```
      SUBROUTINE LINES(ISYMB,TCONC,NPRIO)


      DIMENSION ISYMB(14),TCONC(15,14)
      LOGICAL*1 L
      COMMON /RECORD/NSYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      DATA NULL/'  '/
      NLINES=0
      DO 1 I=1,16
1     CINTEN(I)=0.0
      DO 2 I=1,4
      B(I)=0.0
      C(I)=0.0
2     D(I)=0.0
      CORR1=0.0
      DO 21 J=1,100
      READ(9'J)ISYMB,TCONC
      DO 19 I=1,267
      READ(8'I,ERR=26)NSYMB,WAVE,IPRIOR,CORR2
      DO 20 K=1,14
      IF(ISYMB(K).EQ.NULL)GO TO 19
      IF(NSYMB.NE.ISYMB(K))GO TO 20
      IF(IPRIOR.GT.NPRIO)GO TO 20
      NLINES=NLINES+1
      CORR1=WAVE+CORR2
      WRITE(3'NLINES,ERR=26)NSYMB,WAVE,CORR1,CORR2,IPRIOR,
      1  (TCONC(N,K),N=1,15),CINTEN,B,C,D
20    CONTINUE
      K=14
19    CONTINUE
      IF(ISYMB(K).EQ.NULL)GO TO 23
21    CONTINUE
23    WAVE=0.0
      NLINES=NLINES+1
      CALL AWRITE(NLINES)
      IF(J.GT.1)CALL WSORT(NLINES)
      RETURN
26    CALL ERROR('LINES')
      RETURN
      END
```

```
      SUBROUTINE LISTIT



      LOGICAL*1 L
      COMMON /IO/INPUT,LOG
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      NTERMS=KOUNT(0)   !NTERMS IS # OF TERMS IN POLYNOMIAL
      TYPE 900
      IF(LOG.NE.0) WRITE(LOG,900)
900   FORMAT('OCALIBRATION DATA'/
      1'0 CONC. USED     CALC. CONC     %ERROR',4X,
      2'INTENSITY    ')
      DO 901 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)RETURN
      TYPE 902,ISYMB,WAVE,IPRIOR
      IF(LOG.NE.0)WRITE(LOG,902)ISYMB,WAVE,IPRIOR
902   FORMAT('0',A2,3X,F9.4,2X,I2)
      DO 904 N=1,15
      IF(CONC(N).LE.0. .OR. CINTEN(N).EQ.0.)GO TO 904
      YCALC=ORTVAL(CINTEN(N),NTERMS)
      CERROR=0.0
      CERROR=(CONC(N)-YCALC)*100./CONC(N)
      TYPE 903,CONC(N),YCALC,CERROR,CINTEN(N)
      IF(LOG.NE.0)WRITE(LOG,903)CONC(N),YCALC,CERROR,
     1   CINTEN(N)
904   CONTINUE
903   FORMAT(' ',1PE12.5,2(1PE14.5,0PF9.2))
      IF(MOD(I,2).EQ.0)PAUSE 'TYPE (CR) TO CONTINUE'
901   CONTINUE
      RETURN
      END



      SUBROUTINE LISTVT



      LOGICAL*1 INPUT(133),FILNAM(12),L
      COMMON /IO/IFILE,LOG
      CALL GETFIL('FILE TO LIST? ',8,'OLD','NC',L)
      IF(L)RETURN
      CALL NORY('MAKE HARD COPIES? ',NPRINT)
      I=0
```

```
      CALL PAGE(I,NPRINT)
2     READ(8,101,END=10,ERR=30)INPUT
101   FORMAT(133A1)
      CALL SCOPY(INPUT,INPUT,133)
      CALL TRIM(INPUT)
      IF(I.NE.0)TYPE 103
103   FORMAT(' '$)
      TYPE 104,(INPUT(K), K=1,MINO(LEN(INPUT),72))
104   FORMAT('+',72A1)
      I=I+1
      IF(I.GE.34)CALL PAGE(I,NPRINT)
      GO TO 2
10    CALL PAGE(I,NPRINT)
20    CALL CLOSE(8)
      RETURN
30    TYPE 31
31    FORMAT('0READ ERROR!! PROCESS HALTING!')
      CALL CLOSE(8)
      RETURN
      END



      SUBROUTINE LOCATE



      LOGICAL*1 L,LESCAP
      DIMENSION PEAK(5)
      COMMON /IO/INPUT,LOG
      CALL HEAD('LOCATE ROUTINE')
      NEXPAN=1
      CALL ASK('ENTER WAVELENGTH? ','R',INT,PEAK(1),L)
      IF(L)RETURN
2     NPKS=0
      CALL PEAKUP(PEAK,NPKS,NEXPAN)
      IF(NPKS.EQ.0) GO TO 10
      TYPE 101,(PEAK(I),I=2,NPKS+1)
101   FORMAT(' PEAK IS AT',4F10.4)
      IF(NPKS.GT.1)
      1 CALL ASK('PEAK TO CENTER ON? ','R',INT,PEAK(2),L)
      IF(L)RETURN
      CALL SEEK(PEAK(2))
      CALL ESCAPE(.FALSE.)
      RETURN
10    CALL NORY('NO PEAKS FOUND!!  EXPAND REGION? ',IGO)
      IF(IGO.LE.0)RETURN
      NEXPAN=NEXPAN+1
```

```
      GO TO 2
      END



      SUBROUTINE MOVE(DIR,SPEED)



C
C     SUBROUTINE TO MOVE THE SPECTROMETER IN A
C     GIVEN DIRECTION AT A GIVEN SPEED
C
      LOGICAL*1 DIR,UP,DOWN
      INTEGER*2 SPOUT,SCANO
      INTEGER*4 IT
      DIMENSION ASPEED(13)
      DATA SPOUT,UP,DOWN/"167772,'U','D'/,
     1 ASPEED/.05,.10,.2,.5,1.,2.,5.,10.,25.,50.,100.,
     1  250.,500./
C
C     SET UP BIT 4 FOR DIRECTION
C
      IF(DIR.EQ.UP)SCANO=0
      IF(DIR.EQ.DOWN)SCANO=16
C
C     FIND INTEGER REPRESENTATION OF SPEED
C
      DO 10 I=1,13
      IF(SPEED.LE.ASPEED(I))GO TO 15
10    CONTINUE
      I=13
15    K=I-1
C
C     INTEGER SPEEDS 8,9,10 ARE NOT ALLOWED THEREFORE
C     ADD 3 TO ALL NUMBERS OVER 7
C
      IF(K.GT.7)K=K+3
      SCANO=SCANO+K
C
C     SET UP SCAN
C
      CALL IPOKEB(SPOUT,SCANO)
      CALL GTIM(IT)
      CALL DELAY(IT,0.2)
C
C     START SCAN (SET BIT 5)
C
```

```
      CALL IPOKEB(SPOUT,IPEEKB(SPOUT).OR."40)
      RETURN
      END



      SUBROUTINE NORY(Q,IGO)



      LOGICAL*1 Q(1),ESC,YES,NO,ANS
      COMMON /IO/INPUT,LOG
      DATA ESC,YES,NO/"33,'Y','N'/
1     CALL ITYPE(Q)
      READ(INPUT,30)ANS
30    FORMAT(A1)
      IF((ANS.NE.NO).AND.(ANS.NE.YES).AND.(ANS.NE.ESC))
     1     GO TO 1
      IF(ESC.EQ.ANS) IGO=0
      IF(ANS.EQ.NO) IGO=-1
      IF(ANS.EQ.YES) IGO=1
      RETURN
      END



      SUBROUTINE ORTPOL(W,NCURVE)



      LOGICAL*1 L
      DIMENSION W(15),S(4),PJM1(15),PJ(15),ERROR(15)
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      DO 808 I=1,15
808   ERROR(I)=0.0
      DO 809 I=1,4
      B(I)=0.0      !B,C,AND D ARE CURVE COEFFICIENTS
      C(I)=0.0
      D(I)=0.0
809   S(I)=0.0
      DO 810 I=1,15
      IF(CONC(I).LE.0. .OR. CINTEN(I).EQ.0.)GO TO 810
      B(1)=B(1)+CINTEN(I)*W(I)   !W IS WEIGHTING FACTOR
      D(1)=D(1)+CONC(I)*W(I)
      S(1)=S(1)+W(I)
```

```
810   CONTINUE
      D(1)=D(1)/S(1)
      DO 811 I=1,15
      IF(CONC(I).LE.0. .OR. CINTEN(I).EQ.0.)GO TO 811
      ERROR(I)=CONC(I)-D(1)
811   CONTINUE
      IF(NCURVE.EQ.1)RETURN
      B(1)=B(1)/S(1)
      DO 812 I=1,15
      IF(CONC(I).LE.0. .OR. CINTEN(I).EQ.0.)GO TO 812
      PJM1(I)=1.
      PJ(I)=CINTEN(I)-B(1)
812   CONTINUE
      N=1
820   N=N+1
      DO 821 I=1,15
      IF(CONC(I).LE.0. .OR. CINTEN(I).EQ.0.)GO TO 821
      P=PJ(I)*W(I)
      D(N)=D(N)+ERROR(I)*P
      P=P*PJ(I)
      B(N)=B(N)+CINTEN(I)*P
      S(N)=S(N)+P
821   CONTINUE
      IF(S(N).EQ.0.0)S(N)=1.E-38
      D(N)=D(N)/S(N)
      DO 822 I=1,15
      IF(CONC(I).LE.0. .OR. CINTEN(I).EQ.0.)GO TO 822
      ERROR(I)=ERROR(I)-D(N)*PJ(I)
822   CONTINUE
      IF(N.EQ.NCURVE)RETURN
      B(N)=B(N)/S(N)
      C(N)=S(N)/S(N-1)
      DO 827 I=1,15
      IF(CONC(I).LE.0. .OR. CINTEN(I).EQ.0.)GO TO 827
      P=PJ(I)
      PJ(I)=(CINTEN(I)-B(N))*PJ(I)-C(N)*PJM1(I)
      PJM1(I)=P
827   CONTINUE
      GO TO 820
      END
```

```
      FUNCTION ORTVAL(X,NCURVE)



      LOGICAL*1 L
      COMMON /RECORD/IDUM(70),B(4),C(4),D(4),L
      K=NCURVE
      ORTVAL=D(K)
      PREV=0.0
414   K=K-1
      IF(K.EQ.0) RETURN
      PREV2=PREV
      PREV=ORTVAL
      ORTVAL=D(K)+(X-B(K))*PREV-C(K+1)*PREV2
      GO TO 414
      END



      SUBROUTINE OUTPUT(C)



      COMMON /IO/INPUT,LOG
      LOGICAL*1 C,O,L
      DATA O/'O'/
      IF(LOG.EQ.2)RETURN
      IF(C.NE.O) GO TO 3  !CLOSE FILE IF 'C' NE 'O'
10    CALL NORY('OUTPUT TO DISK? ',IGO)
      IF(IGO.LE.0)RETURN
      CALL GETFIL('ENTER FILENAME? ',4,'NEW','CC',L)
      IF(L)GO TO 10
      LOG=4
      RETURN
3     IF(LOG.NE.0)CALL CLOSE(LOG)
      LOG=0
      RETURN
      END
```

```
      SUBROUTINE PAGE(I,NPRINT)



      IMPLICIT LOGICAL*1 (L)
      DATA LESC,LFF,LETB/"33,"14,"27/
      IF(I.EQ.0) GO TO 2
      IF(NPRINT.LE.0)GO TO 4
      TYPE 100,LESC,LETB   !MAKE HARD COPIES OF FILE
100   FORMAT('+',2A1$)
      T0=SECNDS(0.0)
1     IF(SECNDS(T0).LE.10.0) GO TO 1  !DELAY 10 SEC
      GO TO 2
4     PAUSE 'TYPE (CR) TO CONTINUE'
2     TYPE 100,LESC,LFF  !PAGE CRT TERMINAL
      T0=SECNDS(0.0)
3     IF(SECNDS(T0).LE.1.0) GO TO 3  !DELAY 1 SEC
      I=0
      RETURN
      END



      SUBROUTINE PEAKIT



      LOGICAL*1 L
      COMMON /RECORD/IDUM(94),L
      EQUIVALENCE (IDUM(2),WAVE)
      CALL NORY('LOCATE PEAKS? ',IGO)
      IF(IGO.LE.0)RETURN
      CALL HEAD('PEAK LOCATION ROUTINE')
      CALL ZERO
      IF(L)RETURN
      DO 21 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)RETURN
      CALL FINDPK
      IF(L)RETURN
21    CALL AWRITE(I)
      RETURN
      END
```

```
      SUBROUTINE PEAKUP(WAVE,NPKS,NEXPAN)



      REAL*4 CURDAT(51),LO,HI,WAVE(1)
      LOGICAL*1 LESCAP
      DO 5 I=1,51    !STEP ACROSS WAVELENGTH REGION
      CALL SEEK(WAVE(1)-NEXPAN*(0.25-(I-1)*0.01))
      IF(.NOT.LESCAP())GO TO 5
      CALL ESCAPE(.FALSE.)
      RETURN
5     CURDAT(I)=CURRNT(.FALSE.)
      DO 10 K=1,2  !SMOOTH DATA
      CURMAX=-1.E-20
      CURTM1=CURDAT(1)
      DO 10 I=2,50
      CURTM2=CURDAT(I)
      CURDAT(I)=(CURTM1+CURTM2+CURDAT(I+1))/3.
      CURTM1=CURTM2
10    CURMAX=AMAX1(CURMAX,CURDAT(I))
      NPLT=0
      CALL LABEL
      ASSIGN 20 TO IGO
      GO TO 45    !PLOT DATA
20    CALL ENDPLT
      NPKS=0
      NEND=4
21    K=0
      DO 25 I=NEND,48
      IF(SLOPE(CURDAT(I-3)).LE.0.) GO TO 24
      K=K+1
      IF(K.LT.5) GO TO 25
      NSTART=I-4
      GO TO 26
24    K=0
25    CONTINUE
      CALL ENDPLT
      RETURN
26    K=0
      DO 30 I=NSTART,48
      IF(SLOPE(CURDAT(I-3)).GT.0.) GO TO 29
      K=K+1
      IF(K.LT.5) GO TO 30
      NPEAK=I-4
      GO TO 31
29    K=0
30    CONTINUE
      CALL ENDPLT
      RETURN
```

```
31      HALFPK=CURDAT(NSTART)+(CURDAT(NPEAK)-CURDAT(NSTART))/2.
        CURMAX=CURDAT(NPEAK)
        IF(NPLT.LT.2) CALL LABEL
        ASSIGN 32 TO IGO
        GO TO 45
32      DO 35 I=NPEAK,NSTART,-1
        IF(CURDAT(I).LE.HALFPK) GO TO 36
35      CONTINUE
36      LO=(HALFPK-CURDAT(I))/(CURDAT(I+1)-CURDAT(I))+I
        CALL PLOT(3.5+LO*0.05,HALFPK/CURMAX*5.,3)
        CALL PLOT(3.5+LO*0.05,HALFPK/CURMAX*5.+0.1,2)
        DO 40 I=NPEAK,51
        IF(CURDAT(I).LE.HALFPK) GO TO 41
40      CONTINUE
        CALL ENDPLT
        RETURN
41      HI=(CURDAT(I-1)-HALFPK)/(CURDAT(I-1)-CURDAT(I))+I-1
        CALL PLOT(3.5+HI*0.05,HALFPK/CURMAX*5.,3)
        CALL PLOT(3.5+HI*0.05,HALFPK/CURMAX*5.+0.1,2)
        PEAK=LO+(HI-LO)/2.
        NPKS=NPKS+1
        WAVE(NPKS+1)=WAVE(1)-NEXPAN*(0.25-0.01*(PEAK-1.))-0.005
        NEND=PEAK+4
        CALL PLOT(3.5+PEAK*0.05,CURDAT(IFIX(PEAK+0.5))/
       1  CURMAX*5.,3)
        CALL PLOT(3.5+PEAK*0.05,CURDAT(IFIX(PEAK+0.5))/
       1   CURMAX*5.+0.1,2)
        GO TO 21
45      DO 50 I=1,51
        CALL PLOT(3.5+I*0.05,CURDAT(I)/CURMAX*5.,3)
50      CALL PLOT(3.5+I*0.05,CURDAT(I)/CURMAX*5.,2)
        NPLT=NPLT+1
        GO TO IGO
        END



        SUBROUTINE PLOT(XX,YY,IC)



        IMPLICIT LOGICAL*1 (L)
        COMMON /PLTORG/ IXORG,IYORG
        DATA LGS/"35/
        IF(IABS(IC).LT.2) GO TO 10
        IF(IABS(IC).EQ.3) TYPE 100,LGS
100     FORMAT('+',A1$)
        IF(ABS(XX).GT.239.) XX=SIGN(239.,XX)
```

```
      IX=IFIX(1024.*XX/7.5)-1
      IF(ABS(YY).GT.239.) YY=SIGN(239.,YY)
      IY=IFIX(1024.*YY/7.5)-1
      IX=IX+IXORG
      IY=IY+IYORG
1     IF(IX.LT.0) IX=0
      IF(IX.GE.1024) IX=1023
      IF(IY.LT.0) IY=0
      IF(IY.GE.1024) IY=1023
      LHIY=IY/32+32
      LOY=MOD(IY,32)+96
      LHIX=IX/32+32
      LOX=MOD(IX,32)+64
      TYPE 101,LHIY,LOY,LHIX,LOX
101   FORMAT('+',4A1$)
      IF(IC.GE.0)RETURN
      IXORG=IX
      IYORG=IY
      RETURN
10    IF(IC.EQ.0) TYPE 100,LGS
      IX=IFIX(XX)
      IY=IFIX(YY)
      GO TO 1
      END



      SUBROUTINE POSITION



C
C     SUBROUTINE TO MOVE THE SPECTROMETER TO A WAVELENGTH
C     SPECIFIED BY THE OPERATOR.
C
      LOGICAL*1 L
      CALL HEAD('POSITION ROUTINE')
      CALL ASK('WAVELENGTH? ','R',INT,START,L)
      IF(L)RETURN
      CALL SEEK(START)
      CALL ESCAPE(.FALSE.)
      RETURN
      END
```

```
      SUBROUTINE PROFIL



      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      COMMON /IO/INPUT,LOG
      COMMON /RECORD/ISYMB,WAVE,IDUM(91),L
      REAL*8 FSPEC
      LOGICAL*1 TODAY(15),NAME(40),DATE1(6),LESCAP,L
      EQUIVALENCE (TODAY(1),DATE1(1))
      DATA DATE1/'D','A','T','E',':',' '/
      CALL IRAD50(12,'DXOPLOT  SAV',FSPEC)
      CALL NORY('PLOT OLD DATA? ',IGO)
      IF(IGO.GT.0)CALL CHAIN(FSPEC,INPUT,12)
      CALL OUTPUT('O')
      CALL HEAD('PROFILE ROUTINE')
      CALL DATFIL
      IF(L)RETURN
      CALL PEAKIT
      IF(L)GO TO 7
      NPLOT=0
      CALL NORY('OUTPUT DATA FOR PLOTTING? ',IGO)
      IF(IGO.LE.0)GO TO 11
      IGO=-1
      NPLOT=1
      CALL GETFIL('NAME OF FILE TO STORE PLOTTING DATA? ',
     1 8,'NEW','NC',L)
      IF(L)NPLOT=0
      IF(NPLOT.GT.0)DEFINE FILE 8(0,4,U,NREC8)
      NREC8=1
11    CALL ASK('WAVELENGTH RANGE? ','R',INT,WINT,L)
      IF(L)GO TO 7
      CALL ASK('INCREMENT? ','R',INT,SCINC,L)
      IF(L)GO TO 7
      NPTS=(WINT/SCINC)+1.0
      CALL ASK('INTEGRATION TIME? ','R',INT,SEC,L)
      IF(L)GO TO 7
      CALL DATE(TODAY(7))
4     IGO=-1
      CALL ZERO
      IF(L)GO TO 7
      CALL GETNAM(NAME)
      TYPE 109,NPTS,TODAY,NAME
      IF(LOG.NE.0)WRITE(LOG,109)NPTS,TODAY,NAME
109   FORMAT(' ','#PTS=',I5/' ',15A1/' ',40A1)
      DO 51 J=1,500
      CALL AREAD(J)
      IF(WAVE.EQ.0.0)GO TO 6
      START=WAVE-(WINT/2.0)
```

```
      DO 50 I=1,NPTS
      CALL SEEK(START+(I-1)*SCINC)
      IF(LESCAP()) GO TO 7
      CALL TAKEDA(1,AMP,STD,0)
      IF(LESCAP()) GO TO 7
      IF(LOG.NE.0)WRITE(LOG,110)START+(I-1)*SCINC,J,AMP,MRNG
      IF(NPLOT.GT.0)WRITE(8'NREC8)START+(I-1)*SCINC,AMP
50    TYPE 110,START+(I-1)*SCINC,J,AMP,MRNG
110   FORMAT('0POS. =',F9.4,/'0AMP #    CURRENT    RANGE'/
      1 ' ',I2,1PE14.6,I6)
51    CONTINUE
6     CALL NORY('REPEAT EXPERIMENT? ',IGO)
      IF(IGO.GT.0)GO TO 4
      CALL NORY('CHANGE PARAMETERS? ',IGO)
      IF(IGO.GT.0)GO TO 11
      IF(NPLOT.GT.0)CALL NORY('PLOT DATA? ',IGO)
7     CALL OUTPUT('C')
      CALL CLOSE(3)
      CALL ESCAPE(.FALSE.)
      IF(LOG.EQ.2 .AND. IGO.GT.0)CALL CLOSE(LOG)
      IF(NPLOT.GT.0)CALL CLOSE(8)
      IF(IGO.GT.0)CALL CHAIN(FSPEC,INPUT,12)
      RETURN
      END



      SUBROUTINE RANGE(RNG)


C
C     SUBROUTINE TO CHANGE THE CURRENT TO DIGITAL
C     CONVERTER TO THE INDICATED RANGE
C
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      INTEGER*2 CDOUT,RNG12,RNG14,RNG
C
C     CDOUT IS THE CURRENT TO DIGITAL OUTPUT REGISTER
C
      DATA CDOUT,RNG12,RNG14/"167772,"2000,"175777/
      IF(.NOT.((RNG.EQ.12).OR.(RNG.EQ.14)))RETURN
      MRNG=RNG
      IF(RNG.EQ.12)CALL IPOKE(CDOUT,IPEEK(CDOUT).OR.RNG12)
      IF(RNG.EQ.14)CALL IPOKE(CDOUT,IPEEK(CDOUT).AND.RNG14)
      RETURN
      END
```

```
      SUBROUTINE REPT



      COMMON /IO/INPUT,LOG
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      LOGICAL*1 NAME(40),L,LESCAP
      CALL OUTPUT('O')
      CALL HEAD('REPEAT DATA ROUTINE')
3     CALL ASK('# OF REPEATS? ','I',NRPTS,REAL,L)
      IF(L) GO TO 10
      CALL ASK('#SEC/REPEAT? ','R',INT,SEC,L)
      IF(L) GO TO 10
      CALL NORY('PRINT EACH REPEAT? ',IGO)
      IF(IGO.EQ.0)GO TO 10
      NPRINT=IGO
6     CALL GETNAM(NAME)
      WAVE=CURWAV()
      IF(LOG.NE.0)WRITE(LOG,108)NAME,WAVE
108   FORMAT('0'40A1,/' WAVELENGTH=',F9.4)
      CALL TAKEDA(NRPTS,AMP,STD,NPRINT)
      IF(LESCAP())GO TO 10
      IF(LOG.NE.0)WRITE(LOG,107)AMP,STD
      TYPE 107,AMP,STD
107   FORMAT('0','AVERAGE=',1PE12.3,4X,'%RSD =',0PF9.2)
      CALL NORY('REPEAT EXPERIMENT? ',IGO)
      IF(IGO.GT.0)GO TO 6
      CALL NORY('CHANGE PARAMETERS? ',IGO)
      IF(IGO.GT.0)GO TO 3
10    CALL ESCAPE(.FALSE.)
      CALL OUTPUT('C')
      RETURN
      END



      SUBROUTINE ROBYN(X)



      LOGICAL*1 L,LESCAP
      DIMENSION PEAK(3)
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
     1  CINTEN(16),B(4),C(4),D(4),L
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      EWAVE=WAVE+0.025   !ENDING WAVELENGTH OF SCAN
      TSEC=SEC
```

```
      SEC=1.0
      CWAVE=WAVE-0.020  !STARTING WAVELENGTH OF SCAN
      I=1
5     CALL SEEK(CWAVE)
      CALL TAKEDA(1,PEAK(I),STDX,0)
      IF(LESCAP())GO TO 7
      CWAVE=CWAVE+0.005
      IF(CWAVE.EQ.EWAVE)GO TO 3
      I=I+1
      IF(I.LE.3)GO TO 5
      IPEAK=NPEAK1(PEAK)
      GO TO (1,2,3),IPEAK
1     X=AMAXIN(PEAK)
      SEC=TSEC
      RETURN
2     PEAK(1)=PEAK(2)
      PEAK(2)=PEAK(3)
      I=3
      GO TO 5
3     X=PEAK(3)
      SEC=TSEC
      RETURN
7     CALL ERROR('ROBYN')
      RETURN
      END
```

```
      SUBROUTINE SCAN


      COMMON /IO/INPUT,LOG
      LOGICAL*1 L,LESCAP,LANS
      EQUIVALENCE (ANS,LANS)
      CALL HEAD('SCAN ROUTINE')
      CALL ASK(' SCAN DIRECTION(UP OR DOWN)? ','C',INT,ANS,L)
      IF(L)RETURN
      CALL ASK('SPEED IN NM/MIN.? ','R',INT,SPEED,L)
      IF(L)RETURN
      CALL SETWAV(999.)  !SET DUMMY TARGET WAVELENGTH
      CALL MOVE(LANS,SPEED)
      CALL ESCAPE(.TRUE.)
      CALL ITYPE('TYPE ALTMODE TO STOP THE SCAN.  ')
10    IF(.NOT.LESCAP())GO TO 10
      CALL ESCAPE(.FALSE.)
      RETURN
      END
```

```
      SUBROUTINE SEEK(TARGET) !GO TO A WAVELENGTH



      REAL*4 INTARG
      LOGICAL*1 LESCAP,SPRUN
      CALL ESCAPE(.TRUE.)
      INTARG=TARGET-5.0        !CALC INITIAL TARGET
1     IF(LESCAP()) RETURN
      CURLOC=CURWAV()          !GET CURRENT LOCATION
      IF(CURLOC-TARGET)4,8,2
2     CALL SETWAV(INTARG)
      CALL MOVE('D',500.)      !SCAN DOWN FAST
3     IF(.NOT.SPRUN()) GO TO 1
      IF(LESCAP()) RETURN
      GO TO 3
4     INTARG=TARGET-.5
      IF((CURLOC-INTARG).GE.0.)GO TO 6
      CALL SETWAV(INTARG)
      CALL MOVE('U',500.)      !SCAN UP FAST
6     IF(.NOT.SPRUN()) GO TO 7
      IF(LESCAP()) RETURN
      GO TO 6
7     CALL SETWAV(TARGET)
      CALL MOVE('U',10.)       !SCAN UP SLOW
8     IF(.NOT.SPRUN()) GO TO 9
      IF(LESCAP()) RETURN
      GO TO 8
9     CALL ESCAPE(.FALSE.)
      RETURN
      END



      SUBROUTINE SETLOC



C
C     SUBROUTINE TO INITIALIZE CURLOC AND TURN
C     SPECTROMETER INTERRUPT ON
C
      INTEGER*2 SPSR,SPIN
      EXTERNAL CURLOC,STEPS
      DATA SPSR,SPIN/"167770,"167774/
C
C     GET ADDRESS OF CURLOC AND CURLOC+2
```

```
C
      J=IADDR(CURLOC)
      K=IADDR(STEPS)
C
C     SET CSR BITS TO 2 AND ENABLE INTERRUPT
C
      CALL IPOKE(SPSR,"42)
C
C     STORE MOST SIGNIFICANT DIGITS OF CURLOC
C
      CALL IPOKE(J,IPEEK(SPIN))
C
C     SET CSR BITS TO 3
C
      CALL IPOKE(SPSR,"43)
C
C     STORE LEAST SIGNIFICANT DIGITS OF CURLOC
C
      CALL IPOKE(K,(IPEEK(SPIN).AND."177400).OR."377)
C
C     SET CSR BITS TO 0
C
      CALL IPOKE(SPSR,"40)
      RETURN
      END



      SUBROUTINE SETUP



      DIMENSION NTIME(2)
      LOGICAL*1 LESC,LFF,L
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,NT(2)
      COMMON /IO/INPUT,LOG
      COMMON /RECORD/IDUM(94),L
      DATA LESC,LFF/"33,"14/
      TIMEAS=0.1   !COUNTING PERIOD OF C/F
      NT(1)=7
      NT(2)=0
      L=.FALSE.
      OFFSET=0.0    !OFFSET CURRENT OF C/F
      TYPE 1,LESC,LFF   !PAGE CRT TERMINAL
1     FORMAT('+',2A1$)
      T0=SECNDS(0.0)
11    IF(SECNDS(T0).LE.1.0)GO TO 11 !WAIT ONE SECOND
      CALL ITYPE('SET WAVELENGTH DISPLAY TO SAME
```

```
     1 WAVELENGTH AS MECHANICAL DISPLAY')
      CALL ITYPE('INPUT THE DATE (FORMAT MO,DA,YR)? ')
      READ(INPUT,102)MO,IDA,IYR
102   FORMAT(3I4)
      IRMON=IPEEK("54) !IRMON=BEGINNING OF RESIDENT MONITOR
      CALL IPOKE(IRMON+178,((MO*32+(IDA-2))*32+(IYR-8)))
      CALL ITYPE('INPUT THE TIME (FORMAT HR,MN,SE)? ')
      READ(INPUT,102)IHR,MIN,ISEC
      CALL JTIME(IHR,MIN,ISEC,0,NTIME)
      CALL IPOKE(IRMON+208,NTIME(1))
      CALL IPOKE(IRMON+210,NTIME(2))
      CALL ITYPE('THE SPECTROMETER WILL NOW BE INITIALIZED TO
      1 CARBON 247.86 NM.')
      CALL SETLOC
      CALL ZERO
      RETURN
      END



      SUBROUTINE SETWAV(WAVE)



C
C     CONVERT A WAVELENGTH TO BCD AND MOVE IT TO NEWLOC
C
      LOGICAL*1 LARRAY(10)
      INTEGER*2 DIG(6),BYTE(4),CURLOC,STEPS
      EXTERNAL NEWLOC,CURLOC
      K=IADDR(NEWLOC)
      WAVLEN=WAVE
C
C     IF WAVE IS LESS THAN 0 CONVERT
C
      IF(WAVLEN.LT.0.)WAVLEN=10000.+WAVLEN
C
C     GET INDIVIDUAL DIGITS OF WAVELENGTH
C
C
C     CREATE BYTES FROM DIGITS
C
      ENCODE(10,100,LARRAY)WAVLEN
100   FORMAT(F10.4)
      DECODE(10,101,LARRAY)(DIG(J),J=6,1,-1),BYTE(3)
101   FORMAT(1X,4I1,1X,2I1,I2)
C
C     CREATE BYTES FROM DIGITS
```

```
C
      BYTE(1)=DIG(4)*16+DIG(3)
      BYTE(2)=DIG(6)*16+DIG(5)
      BYTE(4)=DIG(2)*16+DIG(1)
C
C     STORE BYTES AT NEWLOC
C
      DO 20 I=1,4
20    CALL IPOKEB(K+I-1,BYTE(I))
C
C     CHECK TO SEE IF STEPS SHOULD BE SET TO -1
C
      I=IADDR(CURLOC)
C
C     ARE 4 MOST SIGNIFICANT DIGITS EQUAL
C
      IF(IPEEK(I).NE.IPEEK(K))GO TO 30
C
C     ARE TWO LEAST SIGNIFICANT DIGITS EQUAL
C
      IF(IPEEKB(K+3).NE.IPEEKB(I+3)) GO TO 30
C
C     ARE OLD STEPS LESS THAN NEW STEPS
C
      IF(IPEEKB(I+2).LT.IPEEKB(K+2))RETURN
C
C     SET STEPS TO -1
C
30    CALL IPOKEB(I+2,-1)
      RETURN
      END



      FUNCTION SLOPE(CURDAT)



      DIMENSION CURDAT(7)
      DATA A,B,C/7.0,28.0,140.0/
      D=0.
      E=0.
      DO 10 I=1,7
      D=D+CURDAT(I)*FLOAT(I)
10    E=E+CURDAT(I)
      SLOPE=(D*A-B*E)/(A*C-B*B)
      RETURN
      END
```

```
;
      .TITLE SPINT
;
;INTERRUPT ROUTINE TO KEEP TRACK OF THE CURRENT
;WAVELENGTH LOCATION AS INDICATED BY THE DISPLAY
;
;SPECTROMETER DISPLAY CONTROL REGISTERS
      SPSR =      167770     ;STATUS REGISTER
      SPOUT =     SPSR+2     ;OUTPUT REGISTER
      SPIN =      SPSR+4     ;INPUT REGISTER
      SPVEC =     304        ;INTERRUPT VECTOR
      SPINTB =     40        ;INTERRUPT ENABLE BIT
      SPGO =       40        ;SCAN START BIT
;
;
      .MCALL .REGDEF
      .REGDEF
      .GLOBL SPINT,CURLOC,STEPS,NEWLOC,NEWSTP
;LOAD POINT
      .ASECT
      . = SPVEC
      .WORD SPINT           ;INTERRUPT ENTRY
      .WORD 240             ;PRIORITY 5
      .CSECT
;INTERRUPT SERVICE ROUTINE
SPINT: BISB #3,@#SPSR             ;SET CSR BITS
       MOV @#SPIN,CURLOC          ;GET TWO LSD OF DISPLAY
       MOVB CURLOC+1,CURLOC+3     ;MOVE TO CURLOC+3
       DECB @#SPSR                ;CLR CSR 0 BIT
       MOV @#SPIN,CURLOC          ;GET 4 MSD OF POSITION
       BICB #3,@#SPSR             ;CLR CSR BITS
       TSTB STEPS                 ;ONLY STEPS TO CHECK
       BGE STEPCK                 ;YES - TO STEP CHECK
       CMP CURLOC,NEWLOC          ;4 MSD EQUAL?
       BNE EXIT                   ;NO - TO EXIT
       CMPB CURLOC+3,NEWLOC+3     ;2 LSD EQUAL?
       BNE EXIT                   ;NO - TO EXIT
STEPCK:INCB STEPS                 ;YES - INCREMENT STEPS
       CMPB STEPS,NEWSTP          ;STEPS EQUAL?
       BGE STOP                   ;YES - TO STOP
       RTI                        ;RETURN
STOP:  BICB #SPGO,@#SPOUT         ;STOP SCAN
EXIT:  RT1                        ;RETURN
CURLOC:.WORD 0                    ;SAVE ONE WORD FOR POSITION
STEPS: .WORD 0                    ;SAVE ONE LOC, LBYTE FOR
                      ;2 LSD OF CURLOC, RBYTE FOR STEPS
NEWLOC:.WORD 0
NEWSTP:.WORD 0
       .END
```

```
      FUNCTION SPRUN



C
C     FUNCTION TO CHECK IF SPECTROMETER IS RUNNING
C     FUNCTION IS TRUE IF RUNNING FALSE IF NOT
C
      LOGICAL*1 SPRUN
      INTEGER*2 SPOUT
      DATA SPOUT,ION/"167772,"40/
      SPRUN=IPEEK(SPOUT).AND.ION
      RETURN
      END



      SUBROUTINE TAKEDA(NRPTS,AMP,STD,NPRINT)



C
C     NRPTS IS # OF REPEATS
C     AMP IS THE AVERAGE INTENSITY FOR N REPEATS
C     STD IS THE STANDARD DEVIATION OF AMP
C
      COMMON /CURR/TIMEAS,MRNG,OFFSET,SEC,INT(2)
      COMMON /IO/INPUT,LOG
      REAL*8 CURDAT,SUM,SUMSQ
      LOGICAL*1 LESCAP
      CALL ESCAPE(.TRUE.)
      NSEC=IFIX(SEC/TIMEAS)  !NSEC=# OF MEASUREMENTS TO TAKE
      SUM=0.0
      SUMSQ=0.0
      DO 2 I=1,NRPTS
      CURDAT=0.0
      DO 1 K=1,NSEC
      IF(LESCAP())RETURN
1     CURDAT=CURDAT+CURRNT(.FALSE.)  !GET INTENSITY
      CURDAT=CURDAT/NSEC  !CALC AVERAGE INTENSITY
      IF(NPRINT.GT.0.AND.LOG.NE.0)WRITE(LOG,100)CURDAT
      IF(NPRINT.GT.0)TYPE 100,CURDAT
100   FORMAT(' ',1PE12.3)
      SUM=SUM+CURDAT
      SUMSQ=SUMSQ+CURDAT**2
2     CONTINUE
      STD=0.0
```

```
      AMP=SUM/NRPTS
      IF(NRPTS.GT.1)
      1  STD=100.*SQRT(ABS(SUMSQ-SUM**2/NRPTS)/(NRPTS-1))/AMP
      CALL ESCAPE(.FALSE.)
      RETURN
      END



      SUBROUTINE TIME(TIMSTR)



      LOGICAL*1 TIMSTR(8),BLANK,ZERO
      INTEGER*2 ITIME(2),IRMON,OFFTIM
      DATA IRMON,OFFTIM,BLANK,ZERO/"54,"320,' ','0'/
10    CALL GTIM(ITIME)
      CALL CVTTIM(ITIME,IHR,IMIN,ISEC,ITICK)
      IF(IHR.LT.24)GO TO 30
      IHR=IHR-24
      CALL JTIME(IHR,IMIN,ISEC,ITICK,ITIME)
      IATIME=IPEEK(IRMON)+OFFTIM
      CALL IPOKE(IATIME,ITIME(1))
      CALL IPOKE(IATIME+2,ITIME(2))
      CALL INCDAT
      GO TO 10
30    ENCODE(8,400,TIMSTR)IHR,IMIN,ISEC
400   FORMAT(I2,':',I2,':',I2)
      DO 40 I=1,8
40    IF(TIMSTR(I).EQ.BLANK)TIMSTR(I)=ZERO
      RETURN
      END



      SUBROUTINE TOUT



      LOGICAL*1 L
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      CALL NORY('DO YOU WANT TO SEE THE TAPE INFORMATION? '
      1,IGO)
      IF(IGO.LE.0)RETURN
      NSTD=KOUNT(1)
```

```
      K=5
28    KK=K-4
      J=MINO(K,NSTD)
      TYPE 24,(N,N=KK,J)
24    FORMAT('0',8X,5I11)
      DO 34 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.)GO TO 100
34    TYPE 27,ISYMB,WAVE,(CONC(N),N=KK,J)
27    FORMAT(' 'A2,1X,F9.4,1X,5G11.4)
100   IF(J.EQ.NSTD) RETURN
      K=K+5
      GO TO 28
      END



      SUBROUTINE UPDATE



      LOGICAL*1 L
      COMMON /RECORD/ISYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1 CINTEN(16),B(4),C(4),D(4),L
      CALL NORY('UPDATE CALIBRATION CURVES? ',IGO)
      IF(IGO.LE.0) RETURN
      CALL ASK('STANDARD #? ','I',J,REAL,L)
      IF(L) RETURN
      DO 1 I=1,500
      CALL AREAD(I)
      IF(WAVE.EQ.0.0)GO TO 10
      IF(CONC(J).LE.0.0 .OR. CINTEN(J).EQ.0.0)GO TO 1
      CALL GDATA(TEMP,AMP)
      IF(L) RETURN
      FACTOR=TEMP/CINTEN(J)
      DO 3 K=1,15
3     CINTEN(K)=CINTEN(K)*FACTOR
1     CALL AWRITE(I)
10    CALL FIT(.FALSE.)
      RETURN
      END
```

```
      SUBROUTINE WSORT(NLINES)



      DIMENSION IDUM(94)
      LOGICAL*1 L
      COMMON /RECORD/NSYMB,WAVE,CORR1,CORR2,IPRIOR,CONC(15),
      1  CINTEN(16),B(4),C(4),D(4),L
      EQUIVALENCE(IDUM(2),WAVE1)
      DO 10 I=1,NLINES-2
      CALL AREAD(I)
      IF(WAVE.EQ.0.)RETURN
      DO 11 J=I+1,NLINES-1
      READ(3'J)IDUM
      IF(WAVE1.EQ.0.)GO TO 10
      IF(WAVE.LE.WAVE1)GO TO 11
      WRITE(3'I,ERR=25)IDUM
      CALL AWRITE(J)
      CALL AREAD(I)
11    CONTINUE
10    CONTINUE
      RETURN
25    CALL ERROR('WSORT')
      RETURN
      END



      SUBROUTINE ZERO



      LOGICAL*1 L,LESCAP
      DIMENSION AWAVE(4)
      COMMON /RECORD/ISYMB,WAVE,IDUM(91),L
      NPKS=0
      NEXPAN=1
      DO 11 I=2,4
11    AWAVE(I)=0.0
      AWAVE(1)=247.860    !CARBON WAVELENGTH FOR REFERENCE
      CALL PEAKUP(AWAVE,NPKS,NEXPAN)
      NPKS=2
      IF(AWAVE(3).NE.0.0)NPKS=3
      CALL SEEK(AWAVE(NPKS))
      IF(LESCAP())CALL ERROR('ZERO')
      IF(L)RETURN
      PAUSE 'SET DISPLAY TO 247.86. TYPE (CR) TO BEGIN'
```

```
      CALL SETLOC
      RETURN
      END
```

APPENDIX 2: PLOTPR

PLOTPR is the FORTRAN program which plots the experimental data on a video terminal. This software was designed to execute on a DEC PDP-11/03 minicomputer and a Tektronix 4006-1 computer display terminal. A listing of the source statements of PLOTPR is given on the following pages.

```
      PROGRAM PLOTPR


C     PROGRAM FOR PLOTTING PROFILE DATA ON TERMINAL
C     AMES LABORATORY  AMES,IOWA  50011
C
      COMMON /IO/INPUT,LOG,IFILL(10)
      COMMON /FILES/STRING(4)
      LOGICAL*1 L,DATLAB(15,100),GLAB(20),T,YLAB(18),XLAB(16)
      REAL*8 FILNAM
      COMMON /POINTS/X(100),Y(100),IPRPLT(10),NPTS
      COMMON /LABELS/ISYMB(10),T(12),GLAB,XLAB,YLAB,DATLAB
      DATA ISYMB,STRING(1)/1,2,3,4,5,6,7,8,9,10,'DX1:'/
      CALL RCHAIN(IFLAG,INPUT,12)  !WAS THIS CHAINED TO?
2     INPUT=5
      LOG=0
      CALL NORY('KEYBOARD INPUT? ',IGO)
      IF(IGO.GT.0)GO TO 151
      CALL GETFIL('FILENAME? ',1,'OLD','NC',L)
      IF(L)GO TO 2
      INPUT=1
151   CALL SCOPY('0123456789?',T,11)
      CALL SCOPY('INTENSITY(X10-   )',YLAB,17)
      CALL SCOPY('WAVELENGTH(NM)',XLAB,14)
      CALL IRAD50(12,'DX0SURVEYSAV',FILNAM)
      CALL GETFIL('NAME OF FILE THAT CONTAINS DATA TO PLOT? ',
      1  3,'OLD','NC',L)
      IF(L)GO TO 101
      DEFINE FILE 3(5000,4,U,NREC3)
      CALL ITYPE('GRAPH LABEL? ')
      CALL GETSTR(INPUT,GLAB,19,L)
      CALL ASK('# OF POINTS/PROFILE? ','I',NPTS,REAL,L)
      IF(L)GO TO 101
36    CALL ASK('TOTAL # OF PROFILES? ','I',NPROF,REAL,L)
      IF(L)GO TO 101
      DO 20 I=1,NPROF
      TYPE 10,I
10    FORMAT('$NAME OF PROFILE #',I4,1X)
      CALL GETSTR(INPUT,DATLAB(1,I),14,L)
      IF(I.EQ.1)GO TO 20
      IF(LEN(DATLAB(1,I)).EQ.0)
     1  CALL SCOPY(DATLAB(1,I-1),DATLAB(1,I),14)
20    CONTINUE
25    CALL ASK('TOTAL # OF PROFILES TO PLOT? ','I',NPLTS,
     1  REAL,L)
      IF(L)GO TO 101
      IF(NPLTS.LE.10)GO TO 26
      CALL ITYPE('MAXIMUM OF TEN PLOTS')
```

```
      GO TO 25
26    CALL ITYPE('PROFILE #''S TO PLOT? ')
      READ(INPUT,35) (IPRPLT(I),I=1,NPLTS)
35    FORMAT(10I4)
      CALL PLTPRO(NPLTS)
      CALL ENDPLT
      CALL NORY('MORE PLOTS? ',IGO)
      IF(INPUT.NE.5)PAUSE 'TYPE (CR) TO CONTINUE'
      IF(IGO.GT.0) GO TO 25
101   CALL CLOSE(3)
C     IF THIS PROGRAM WAS CHAINED TO RETURN TO SURVEY.SAV
      IF(IFLAG.LT.0)CALL CHAIN(FILNAM,INPUT,12)
      STOP 'NORMAL STOP'
      END



      SUBROUTINE AXIG



      LOGICAL*1 TIC(9),T,GLAB(20),XLAB(16),YLAB(18)
      LOGICAL*1 DATLAB(15,100)
      COMMON /XTRM/XMAX,XMIN,YMAX,YMIN,XSCALE,YSCALE
      COMMON /LABELS/ISYMB(10),T(12),GLAB,XLAB,YLAB,DATLAB
C
C     LOOP TO DRAW TICS AND LABEL TICS
C
      IEXP=ABS(ALOG10(ABS(YMAX))-1.0)   !GET EXPONENT FOR YLAB
      YLAB(15)=T(IEXP/10+1)
      YLAB(16)=T(MOD(IEXP,10)+1)
      CALL SYMBOL(0.01,5.05,0.15,GLAB,0.0,LEN(GLAB))
      CALL SYMBOL(2.0,-0.75,0.15,XLAB,0.0,14)
      CALL SYMBOL(-0.75,1.25,0.15,YLAB,90.0,17)
      X = 0.0
      TICVAL = XMIN
   70 IF (X.GT. 5.0) GO TO 100
      CALL PLOT(X,0.0,3)     !DRAW TIC
      CALL PLOT(X,-0.10,2)
      CALL NUMBER (TICVAL,TIC,-1)   !MAKE REAL NUMBER ASCII
      CALL SYMBOL(X-0.12,-0.30,0.10,TIC,0.0,9)   !PLOT NUMBER
      TICVAL = TICVAL + XSCALE
      X = X + 1.0
      GO TO 70
C
C     LOOPING TO DRAW TICS AND LABEL TICS
C
  100 CALL NUMBER(YMAX,TIC,1)   !SET EXPONENT IN SUB 'NUMBER'
```

```
      Y = 0.0
      TICVAL = YMIN
  180 IF (Y .GT.5.0) RETURN
      CALL PLOT(0.0,Y,3)
      CALL PLOT(-0.10,Y,2)    !DRAW TIC
      CALL NUMBER(TICVAL,TIC,0)
      CALL SYMBOL(-0.30,Y-0.12,0.10,TIC,90.0,9)
      TICVAL = TICVAL + YSCALE
      Y = Y + 1.0
      GO TO 180
C
      END
```

```
;
;
      .TITLE CHRGEN
;
;A ROUTINE TO UNPACK THE X AND Y POSITIONS USED BY
;SYMBOL TO GRAPH CHARACTERS
;
;ROUTINE IS ACCESSED WITH A FORTRAN CALL
;
;I.E. CALL CHRGEN(S,IX,IY)
;
;S IS THE PACKED X AND Y POSITIONS
;
;IX IS AN INTEGER VARIABLE TO RECEIVE THE
;X POSITION
;
;IY IS AN INTEGER VARIABLE TO RECEIVE THE
;Y POSITION
;
      .MCALL .REGDEF
      .REGDEF
;
;ROUTINE ENTRY POINT
;
      .GLOBL CHRGEN
CHRGEN: TST (R5)+             ;ADV. POINTER
        MOVB @(R5)+,R0        ;GET PACKED X AND Y
        TSTB (R5)+            ;ADV. POINTER
        MOV R0,R1
        BIC #177417,R0        ;CLR ALL BUT BITS 4-7
        ASH #-4,R0            ;SHIFT RIGHT 4
        MOV R0,@(R5)+         ;RETURN X
        BIC #177760,R1        ;CLR ALL BUT BITS 0-3
        MOV R1,@(R5)          ;RETURN Y
```

```
        RTS PC
        .END


      SUBROUTINE CURVE



      DIMENSION X(100),Y(100),XS(4),YS(4)
      COMMON /POINTS/X,Y,IPRPLT(10),NPTS
      IF (NPTS .LT. 4) RETURN
      L=0
      K=1
      K3=1
      XP=X(1)
      YP=Y(1)
      N1=3
      CALL PLOT(XP,YP,3)
  900 N11=N1
      DO 1000 I=N11,NPTS
      IF ((N1-4).GT.0) N1=4
      M=I+L
   82 DO 100 J=1,4
      IDUM=I-N1+J
      YS(J)=Y(IDUM)
  100 XS(J)=X(IDUM)
      XPS=XP
  125 XPS=XPS+0.07
      GO TO (126,127,126),K3
126   K3=K3+1
      IF ((XPS-XS(K3)).GE.0.0)GO TO 300
      YPS=PINTR(XS(K),YS(K),XPS,2)
      K3=K3-1
      GO TO 151
  127 IF ((XPS-XS(3)).GE.0.0)GO TO 300
      YPS1=PINTR(XS,YS,XPS,2)
      YPS2=PINTR(XS(2),YS(2),XPS,2)
      YSTRL=PINTR(XS(2),YS(2),XPS,1)
      YPS=YPS1
      IF ((ABS(YPS2-YSTRL)-ABS(YPS1-YSTRL)).LT.0.0)YPS=YPS2
  151 XP=XPS
      YP=YPS
      CALL PLOT(XP,YP,2)
      GO TO 125
  300 N1=4
      CALL PLOT(X(M-1),Y(M-1),2)
      XP=X(I-1)
```

```
      YP=Y(I-1)
1000  CONTINUE
      K3=K3+1
      K=2
      L=1
      N1=NPTS
      IF ((K3-4).LT.0)GO TO 900
      RETURN
      END


      FUNCTION IDIG4(TNUM,TEXP,XNUM)


      IMPLICIT INTEGER*2 (I-N,T)
200   IF ( (TNUM/10000) .LT. 1 ) GO TO 250
      TNUM = TNUM/10
      GO TO 200
250   IF ( (TNUM/1000)  .GE. 1 ) GO TO 300
      TNUM = XNUM/10.0**(TEXP-4)
      GO TO 250
300   IDIG4 = TNUM
      RETURN
      END


      FUNCTION NPEAK1(PEAK)


      DIMENSION PEAK(3)
      NPEAK1=3
      IF(PEAK(1).LT.PEAK(2) .AND. PEAK(2).LT.PEAK(3))NPEAK1=2
      IF(PEAK(1).LT.PEAK(2) .AND. PEAK(2).GT.PEAK(3))NPEAK1=1
      RETURN
      END
```

```
      SUBROUTINE NUMBER(FNUM,NSTR,MODE)



      LOGICAL*1 NSTR(9),BLANK,MINUS,POINT,T,LARRAY(10)
      LOGICAL*1 GLAB(20)
      LOGICAL*1 XLAB(16),YLAB(18),DATLAB(15,100)
      COMMON /LABELS/ISYMB(10),T(12),GLAB,XLAB,YLAB,DATLAB
      DATA BLANK/' '/,MINUS/'-'/,POINT/'.'/
      DO 111 I=1,9
  111 NSTR(I) = BLANK
      XNUM = ABS(FNUM)
      IF(MODE.LT.0)GO TO 500
      IF (XNUM .EQ.0.0) GO TO 80
      IF (FNUM .LT. 0.0) NSTR(1) = MINUS
      IF(MODE.EQ.1)IEXP = ALOG10(XNUM) - 1.0
      NEXP=IEXP
      INUM = XNUM/10.0**(IEXP-3)
      INUM = IDIG4(INUM,IEXP,XNUM) + 5
      NUM  = IDIG4(INUM,IEXP,XNUM) / 10
      GUESS = NUM*10.0**(IEXP-2)
      GUESSL = GUESS / 10.0
      GUESSH = GUESS * 10.0
      IF ( XNUM .GT. GUESS ) GO TO 350
      IF ( (XNUM - GUESSL) .GE. (GUESS - XNUM) ) GO TO 70
      NEXP = IEXP - 1
      GO TO 70
  350 IF ( (GUESSH - XNUM) .GE. (XNUM - GUESS) ) GO TO 70
      NEXP = IEXP+1
   70 IF(NEXP-IEXP)41,51,31
   31 NSTR(2)=T(NUM/100+1)
      NSTR(3)=T(MOD(NUM,100)/10+1)
      NSTR(4)=POINT
      NSTR(5)=T(1)
      RETURN
   41 NSTR(2)=T(1)
      NSTR(3)=POINT
      NSTR(4)=T(NUM/100+1)
      NUM=MOD(NUM/100)
      NSTR(5)=T(NUM/10+1)
      RETURN
   51 NSTR(2) = T(NUM/100 +1)
      NUM =  MOD(NUM,100)
      NSTR(3) = POINT
      NSTR(4) = T(NUM/10 +1)
      NSTR(5) = T(MOD(NUM,10) +1)
      RETURN
   80 NSTR(2) = T(1)
      NSTR(3) = POINT
```

```
      NSTR(4)  =  T(1)
      NSTR(5)  =  T(1)
      RETURN
500   ENCODE(10,100,LARRAY)XNUM
  100 FORMAT(F10.4)
      DECODE(10,101,LARRAY) (NSTR(J),J=2,5),(NSTR(J),J=7,8)
  101 FORMAT(1X,4I1,1X,2I1,2X)
      DO 102 I=3,8
  102 IF(I.NE.6)NSTR(I-1)=T(NSTR(I)+1)
      NSTR(8)=BLANK
      NSTR(5)=POINT
      RETURN
      END



      FUNCTION PINTR (XS,YS,XA,KORD)



      DIMENSION XS( 1),YS( 1)
      YP=0.0
      NPTS=KORD+1
      DO 10 J=1,NPTS
      YADD=YS(J)
      DO 11 K=1,NPTS
   11 IF((J-K).NE.0)YADD=(XA-XS(K))/(XS(J)-XS(K))*YADD
   10 YP=YP+YADD
      PINTR=YP
      RETURN
      END



      SUBROUTINE PLTPRO(NPLTS)



      LOGICAL*1 DATLAB(15,100),T(12),GLAB(20),XLAB(16)
      LOGICAL*1 YLAB(18)
      COMMON /POINTS/X(100),Y(100),IPRPLT(10),NPTS
      COMMON /LABELS/ISYMB(10),T,GLAB,XLAB,YLAB,DATLAB
      COMMON /XTRM/XMAX,XMIN,YMAX,YMIN,XSCALE,YSCALE
      XMAX=-1.E30      !INITIALIZE XMAX,XMIN,YMAX,YMIN
      XMIN=1.E30
      YMAX=-1.E30
```

```
      YMIN=1.E30
      DO 51 J=1,NPLTS
      CALL GETDAT(.FALSE.,J)  !READ DATA TO PLOT
      DO 51 I=1,NPTS
      XMAX=AMAX1(XMAX,X(I))     !CALCULATE MAXIMA AND MINIMA
      XMIN=AMIN1(XMIN,X(I))
      YMAX=AMAX1(YMAX,Y(I))
51    YMIN=AMIN1(YMIN,Y(I))
      CALL LABEL         !INITIALIZE CRT FOR PLOTTING
      CALL PLOT(1.0,0.75,-3) !SET ORIGIN FOR PLOTTING
      CALL PLOT(0.0,0.0,3)  !PLOT X AND Y AXIS
      CALL PLOT(0.0,5.0,2)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(5.0,0.0,2)
      XMON=XMIN
      XSCALE=1.0
      IF((XMAX-XMIN).NE.0.)
     1   CALL SCALE(XMAX,XMIN,5.0,XSCALE,XMON)
      XMIN=XMON
      YMON=YMIN
      YSCALE=1.0
      IF((YMAX-YMIN).NE.0.)
     1   CALL SCALE(XMAX,XMIN,5.0,YSCALE,YMON)
      YMIN=YMON
      DO 81 N=1,NPLTS
      CALL GETDAT(.TRUE.,N)   !READ DATA FROM DISK
      CALL CURVE       !PLOT DATA ON CRT
      DO 81 K=1,NPTS
81    CALL SYMBOL(X(K)-0.05,Y(K)-0.05,0.10,ISYMB(N),0.0,1)
      CALL AXIG       !LABEL AXIS
      RETURN
      END



      SUBROUTINE SCALE (AYMAX,AYMIN,AXISZE,CSCALE,CYMIN)



      ADY1=(AYMAX-AYMIN)/AXISZE
      TYMIN=AYMIN+0.05*ADY1
      TYMAX=AYMAX-0.05*ADY1
      ADY=(TYMAX-TYMIN)/AXISZE
      K1=0
    5 IF (ADY-80.00001)  6,6,90
    6 IF (ADY-10.00001) 10,10,20
    7 ADY=ADY*10.0
      K1=K1-1
```

```
      GO TO 6
   90 ADY=ADY/10.0
      K1=K1+1
      GO TO 5
   10 IF (ADY-8.000001)  7,7,11
   11 TDY=10.00005
      I=1
      GO TO 96
   20 IF (ADY-20.00001) 21,21,40
   21 TDY=20.00005
      I=2
      GO TO 96
   40 IF (ADY-40.00001)  41,41,50
   41 TDY=40.00005
      I=3
      GO TO 96
   50 IF (ADY-50.00001)  51,51,80
   51 TDY=50.00005
      I=4
      GO TO 96
   80 TDY=80.00005
      I=5
   96 TSF=TDY*10.0**K1
      TYMIN=AYMIN+0.05*TSF
      TYMAX=AYMAX-0.05*TSF
      TYMIN1=TYMIN-AMOD(TYMIN,TSF)
      IF(TYMIN.LT.0.0)TYMIN1=TYMIN1-TSF
      IF ((TYMAX-TYMIN1-AXISZE*TSF)    .LE.0.0)GO TO 102
      GO TO (21,41,51,80,101),I
  101 K1=K1+1
      GO TO 11
102   CYMIN=TYMIN1
      CSCALE=TSF
      RETURN
      END



      SUBROUTINE SYMBOL(XX,YY,HGT,BCD,TH,N)



      LOGICAL*1 BCD(1),SYM(702),CHRSTR(129),STR(2)
      DIMENSION ISYMB(2,128)
      DATA STR/"0,"0/
      DATA CHRSTR/"1,"2,"3,"4,"5,"6,"7,"10,"11,"12,"13,"14,
     +  "16,"17,"20,"21,"22,"23,"177,"24,"25,"15,"26,
     +   "27,"137,"15,"30,"31,"32,"33,"34,"35,"135,"133,"36,
```

```
     +  "37,"140,"141,"142,"143,"144,"145,"146,"147,
     +  "150,"151,"152,"153,"154,"155,"156,"157,"160,
     +  "173,"175,"134,"161,"162,"163,"164,"165,"166,
     +  "136,"167,"40,"101,"102,"103,"104,"105,"106,
     +  "107,"110,"111,"170,"56,"74,"50,"53,"174,"46,
     +  "112,"113,"114,"115,"116,"117,"120,"121,"122,
     +  "41,"44,"52,"51,"73,"171,"55,"57,"123,"124,
     +  "125,"126,"127,"130,"131,"132,"172,"54,"45,
     +  "176,"76,"77,"60,"61,"62,"63,"64,"65,"66,"67,
     +  "70,"71,"72,"43,"100,"47,"75,"42,"0/
        DATA ISYMB/1,08,8,12,19,06,24,07,30,07,36,07,42,07,
     +   48,06,53,08,60,07,66,14,24,13,79,06,84,04,475,06,
     +   123,02,87,02,656,01,499,03,89,08,494,05,655,01,
     +   103,08,95,08,502,02,657,01,504,02,554,08,567,06,
     +   573,03,576,06,576,13,589,09,598,09,607,09,616,06,
     +   630,12,633,12,622,10,645,07,658,09,667,07,458,14,
     +   674,12,686,09,695,08,653,01,654,01,520,05,506,14,
     +   111,06,115,06,525,04,121,04,123,04,132,02,529,07,
     +   481,05,536,10,536,13,549,05,562,05,489,05,486,05,
     +   652,01,137,09,146,12,159,08,152,07,125,07,125,06,
     +   159,10,169,06,175,06,181,11,198,05,111,03,204,04,
     +   98,05,208,02,210,10,221,05,226,07,122,03,233,05,
     +   236,04,243,12,258,07,246,12,258,10,268,07,286,12,
     +   298,08,306,04,192,12,310,03,298,02,135,02,315,12,
     +   284,04,220,06,329,03,238,05,132,05,332,05,337,07,
     +   458,17,198,06,344,14,358,02,118,03,271,13,246,09,
     +   360,05,365,08,373,13,390,08,398,09,379,11,405,05,
     +   313,16,410,12,192,11,422,11,433,16,449,04,103,05,
     +   449,09/
      DATA SYM/"042,"044,"004,"000,"100,"104,"044,"042,"044,
     +  "024,
     +   "003,"001,"020,"060,"101,"103,"064,"044,"042,"044,
     +   "001,"101,"044,"042,"044,"040,"042,"002,"102,"042,
     +   "004,"100,"042,"000,"104,"042,"044,"002,"040,"102,
     +   "044,"042,"040,"044,"002,"102,"044,"042,"000,"104,
     +   "004,"100,"042,"104,"004,"104,"000,"100,"000,"042,
     +   "004,"042,"104,"042,"040,"042,"104,"063,"023,"004,
     +   "023,"021,"000,"021,"061,"100,"061,"063,"042,"104,
     +   "004,"100,"000,"042,"044,"040,"042,"046,"047,"147,
     +   "360,"145,"045,"360,"043,"143,"360,"104,"110,"106,
     +   "046,"146,"044,"144,"360,"146,"046,"360,"063,"127,
     +   "150,"046,"144,"360,"143,"043,"360,"044,"146,"050,
     +   "151,"051,"042,"142,"151,"051,"046,"126,"046,"042,
     +   "142,"051,"142,"360,"042,"151,"042,"045,"145,"045,
     +   "050,"071,"131,"150,"142,"143,"145,"126,"046,"126,
     +   "147,"150,"131,"051,"042,"122,"143,"150,"150,"131,
     +   "071,"050,"043,"062,"122,"143,"145,"125,"042,"051,
     +   "046,"146,"151,"142,"062,"122,"102,"111,"071,"131,
     +   "146,"127,"067,"046,"044,"063,"123,"144,"360,"102,
     +   "110,"065,"066,"106,"105,"065,"360,"102,"062,"063,
```

```
+    "103,"102,"061,"151,"130,"123,"142,"041,"052,"142,
+    "067,"070,"111,"130,"045,"044,"063,"103,"144,"051,
+    "043,"062,"122,"143,"151,"051,"042,"045,"151,"360,
+    "107,"142,"042,"051,"105,"151,"142,"051,"042,"106,
+    "142,"151,"107,"151,"360,"150,"131,"071,"050,"043,
+    "062,"122,"143,"150,"360,"104,"142,"042,"051,"131,
+    "150,"147,"126,"046,"126,"145,"142,"111,"104,"360,
+    "062,"103,"122,"062,"360,"104,"106,"126,"147,"150,
+    "131,"071,"050,"051,"151,"111,"102,"360,"043,"123,
+    "144,"145,"126,"066,"047,"070,"150,"045,"145,"105,
+    "143,"047,"105,"043,"147,"051,"070,"063,"042,"045,
+    "165,"164,"126,"147,"150,"131,"071,"050,"047,"066,
+    "126,"145,"143,"122,"062,"043,"045,"066,"051,"102,
+    "151,"051,"107,"102,"107,"151,"051,"151,"042,"142,
+    "360,"066,"126,"070,"050,"051,"071,"070,"360,"151,
+    "042,"360,"123,"143,"142,"122,"123,"025,"165,"070,
+    "111,"102,"062,"122,"050,"071,"131,"150,"146,"044,
+    "042,"142,"050,"071,"131,"150,"147,"126,"066,"126,
+    "145,"143,"122,"062,"043,"050,"071,"131,"150,"051,
+    "044,"144,"124,"131,"122,"102,"142,"043,"062,"122,
+    "143,"145,"126,"046,"051,"151,"150,"103,"102,"043,
+    "062,"122,"143,"150,"131,"071,"050,"046,"065,"125,
+    "146,"044,"144,"124,"123,"127,"126,"146,"046,"066,
+    "067,"063,"146,"127,"107,"066,"065,"104,"124,"145,
+    "147,"130,"070,"047,"044,"063,"123,"144,"127,"111,
+    "131,"127,"360,"067,"051,"071,"067,"147,"127,"106,
+    "067,"047,"026,"025,"044,"064,"105,"106,"105,"124,
+    "144,"165,"166,"147,"062,"130,"162,"046,"206,"062,
+    "043,"064,"102,"112,"172,"102,"064,"124,"102,"111,
+    "067,"127,"111,"025,"165,"126,"124,"165,"042,"106,
+    "142,"021,"201,"032,"212,"025,"145,"360,"063,"064,
+    "104,"103,"063,"360,"066,"067,"107,"106,"066,"161,
+    "041,"126,"052,"172,"022,"107,"162,"022,"050,"071,
+    "110,"102,"110,"131,"150,"102,"104,"064,"124,"104,
+    "110,"070,"130,"110,"111,"106,"066,"126,"025,"066,
+    "064,"025,"165,"022,"041,"061,"102,"111,"132,"152,
+    "171,"043,"147,"360,"047,"143,"044,"124,"145,"146,
+    "127,"047,"046,"102,"146,"024,"045,"065,"123,"143,
+    "164,"360,"166,"145,"125,"067,"047,"026,"041,"061,
+    "102,"105,"126,"107,"112,"073,"053,"141,"121,"102,
+    "105,"066,"107,"112,"133,"153,"042,"067,"064,"103,
+    "123,"144,"147,"144,"163,"063,"067,"047,"147,"127,
+    "123,"046,"066,"064,"103,"124,"126,"146,"360,"102,
+    "110,"360,"067,"046,"044,"063,"123,"144,"146,"127,
+    "067,"360,"025,"165,"043,"146,"360,"046,"066,"123,
+    "143,"360,"361,"362,"363,"364,"365,"047,"044,"063,
+    "104,"107,"104,"123,"144,"147,"050,"067,"105,"143,
+    "360,"042,"105,"130,"110,"067,"106,"126,"145,"144,
+    "123,"103,"064,"065,"106,"147,"107,"066,"064,"103,
+    "143,"360,"145,"045,"046,"067,"106,"064,"106,"127,
```

```
     +           "146,"122/
      THETA=TH/57.2958
      HINC=HGT/7.
      CINC=COS(THETA)*HINC
      SINC=SIN(THETA)*HINC
      X=XX
      Y=YY
      DO 20 I=1,N
      IC=3
      STR(1)=BCD(I)
      K=INDEX(CHRSTR,STR)
      IF(K.EQ.0) GO TO 10
      L=ISYMB(1,K)
      M=ISYMB(2,K)
      DO 10 IK=1,M
      CALL CHRGEN(SYM(L-1+IK),IX,IY)
      F=1.          !NORMAL SCALING FACTOR
      IF(K.LE.15) F=7./4.
      IF(IX.NE.15) GO TO 5
      IC=3
      GO TO 10
5     CALL PLOT(X+CINC*F*IX-SINC*F*IY,
     1  Y+CINC*F*IY+SINC*F*IX,IC)
      IC=2
10    CONTINUE
      X=X+7.*CINC
      Y=Y+7.*SINC
20    CALL PLOT(X,Y,3)
      RETURN
      END
```