CONTROL

A solid-source mass spectrometer data acquisition program for use with Hewlett-Packard computers

by
Brian D. Marshall

UNITED STATES GEOLOGICAL SURVEY
OPEN-FILE REPORT 89-160

Revision of Mar 27, 1989

Table of Contents

Introduction

CONTROL is a computer program which provides a user-friendly environment for operating a simple solid-source mass spectrometer. It was developed for use with modified NBS-type solid-source mass spectrometers. These mass spectrometers have 6-inch radius flight tubes and wre originally manufactured in 1968. CONTROL version 1.1 consists of the following program files: CONTROL, Subfile, CABDM, SRBDM, KBDM, RBBDM, and IRBDM.

Requirements

Hardware

CONTROL runs on a Hewlett-Packard series 200 (or 300) microcomputer with at least 512K of RAM and a dual floppy disk drive. Interfaces must be present for controlling the magnetic field and for controlling and reading a digital voltmeter. Naturally, the software will need to be modified for various hardware configurations. The source code described here requires an HPIB printer with graphics capability set to select code 701, a Solartron 7062 Systems Voltmeter with HPIB interface at select code 716, and a GPIO interface for controlling the magnet set to select code 12.

Software

CONTROL requires H-P BASIC version 3.0 (or later) and the following binary extensions must be loaded prior to running CONTROL: IO, HPIB, GPIO, DISC, MS, MAT, CLOCK, TRANS, GRAPH, GRAPHX.

Running a sample

Getting started

To get CONTROL into memory type "LOAD "CONTROL"" and press the Execute key. Then press the Run key. An opening screen will appear which shows the current time and date, a statement containing the version of CONTROL and a soft key menu at the bottom of the screen. In order to prepare the voltmeter for data acquisition press the soft key labelled `Setup DVM`. The voltmeter will be set to remote control, the local panel will be disabled, and the initialization will be performed. The voltmeter is then set to a 1 second integration time and started tracking.

The five upper soft keys start various data acquisition programs, four programs each dedicated to a single element (potassium, calcium, rubidium, or strontium) and a fifth program for general data acquisition with no mass discrimination correction or spike subtraction. The `Erase Disk` option allows the user to initialize a floppy disk placed in the left-hand disk drive. The `Shut Down` option simply clears the screen, sets the voltmeter to a standby mode and stops the program.

As an example, assume you wish to run a strontium sample. Pressing the `Strontium` key starts the strontium program which immediately asks for a sample name. Any characters are OK and there is a maximum limit of 35 characters for the name. Press Enter to continue.

At this point, the program presents a soft key menu with the options for type of sample being run, unspiked, spiked, or a spike itself. If the sample is spiked then the user will be presented with choices for spike ratios used to correct the data. Note that these choices must be entered ahead of time into the program itself. If you are analyzing a spike then the program will not correct for mass discrimination.

Setting the peak positions

Once the program progresses through the preliminaries, you will be presented with a screen showing the current peak and magnet setting (e.g. 88 = 3580) and a soft key menu as follows:

```
Step            UP            Set           Field
Cycle           DOWN          Scan          3580          Escape
```

These soft keys allow you to center the peaks using the computer in combination with a manual voltage offset device on the mass spectrometer. The `Step` key switches the magnetic field to the next isotope in sequence, returning to the first isotope upon completion of the sequence. The `UP` and `DOWN` keys change the magnetic field one digital step at a time. The knob will change the magnetic field faster if your keyboard has one. Once the peak of interest is centered, use the `Set` key to store the magnetic field value with the peak. The magnetic field is always displayed below the `Field` key; these keys perform no other purpose. The `Cycle` key steps through the peaks automatically waiting 5 seconds on each before switching. It can only be stopped by pressing the `Hold` key which will become active during the Cycle routine. Cycling the magnetic field before centering the peaks eliminates some of the hysteresis in the magnet, allowing for more accurate centering. It is emphasized that centering of the peaks is critical since the program will not adjust the relative peak positions during the run. The program centers the peaks by finding the largest peak, centering it, and then adjusting all other peaks equally. Of course, you may re-center the peaks manually any time during the run. The `Escape` key exits this segment and presents the user with a screen showing the current run parameters.

Run parameters

The Current Run Parameters screen looks similar to the following:

```
                        CURRENT RUN PARAMETERS

Total number of blocks= 30

Number of scans per block= 11

Integration times (seconds):
     88= 5      87= 7      86= 7      84= 7      85= 9

Delay times (seconds):
     Before 88= 5    Before 87= 6    Before 86= 4    Before 84= 4    Before 85= 6

Print means only.

Correct for 87Rb.
```

This screen always comes up after exiting the peak adjustment routine. If you answer yes to the question "Do you want to change any of these parameters?" then you will be presented with a series of questions asking for new values. Any yes/no question may be answered with a y or Y or n or N. When changing the integration and/or delay times enter all required values separated by commas. Any question may be skipped by pressing the Enter key without entering any value. If changes are made you will have the opportunity to review them before starting or resuming data acquisition.

After reviewing the current run parameters and answering no to the query, you will be presented with a soft key menu allowing you to start data acquisition, abort this analysis, or escape back to the peak adjustment routine. If you select `START` then the program enters its

main portion. If running a strontium or calcium the program checks to see if a disk with a name containing "DATA" is in the left-hand disk drive. You will be required to provide one if necessary. A data file containing the sample name and date is created on this disk; the name for the file is created from the current date and time as "MMYYDDHH" where MM is a two-letter code for the month and YY, DD, and HH are the year, day, and hour, respectively. As each block is finished the means are stored in this file as a precaution against a hardware or software failure. The use of these files will require another program and is left up to the user. The program then flashes `Printer ON LINE?` for five seconds allowing you to turn it on if it is not already on. Data acquisition will then begin.

Data Acquisition

During background measurements, the screen shows the current measured value and time of measurement. A soft key menu is displayed as follows:

```
Block                          Strontium      Field          Below
1 of 20                        Baselines      1815           85
```

All of these keys are for display purposes only; they will do nothing when pressed. They communicate to the user the current block number, the current field value and which baseline is being measured. During background and interference measurements as well as during peak centering the computer may not be interrupted easily by the user.

When the peak tops are being measured the screen shows a plot of the average peak values and displays the current measured value (baselines are subtracted) and time. The soft key menu looks something like this:

```
Block          Scan           Strontium      Field
1 of 20        1 of 11        88             3650           ABORT Block
```

It is similar to the menu displayed during measurement of the baselines, except that it provides a key (`ABORT Block`) which will exit the peak measurement routine immediately. If the user presses this key, the program will forget the data for that block (including the baselines) and return to the peak adjustment routine. Data acquisition may be resumed after making any desired adjustments to the beam or to the run parameters.

After a block of data is finished (immediately after the last baseline measurement), the computer will display a soft key menu as follows:

```
                               PAUSE
FINISH                         Change Params                  ABORT RUN!
```

The user will have only a few seconds to respond to this menu; the program will start a new block (assuming that the maximum number of blocks has not been reached) if no key is pressed. The `PAUSE` key simply pauses the program allowing the user to make minor adjustments (typically focussing) to the hardware which is out of the computer's control. Data acquisition can then be resumed; baselines will not be re-measured. The `Change Params` key also pauses the program, but unlike the `PAUSE` key, it returns the user to the peak adjustment routine so that the peak centers can be checked and any run parameters can be changed. When data acquisition is resumed, the baselines are re-measured. For this reason, it is a very good idea to choose `Change Params` when making hardware adjustments which affect the beam intensity. The `FINISH` key can be pressed if the user decides that enough data have been acquired; the program will quit normally, allowing final editing of the data. The `ABORT RUN!` key immediately quits the program without any final data output.

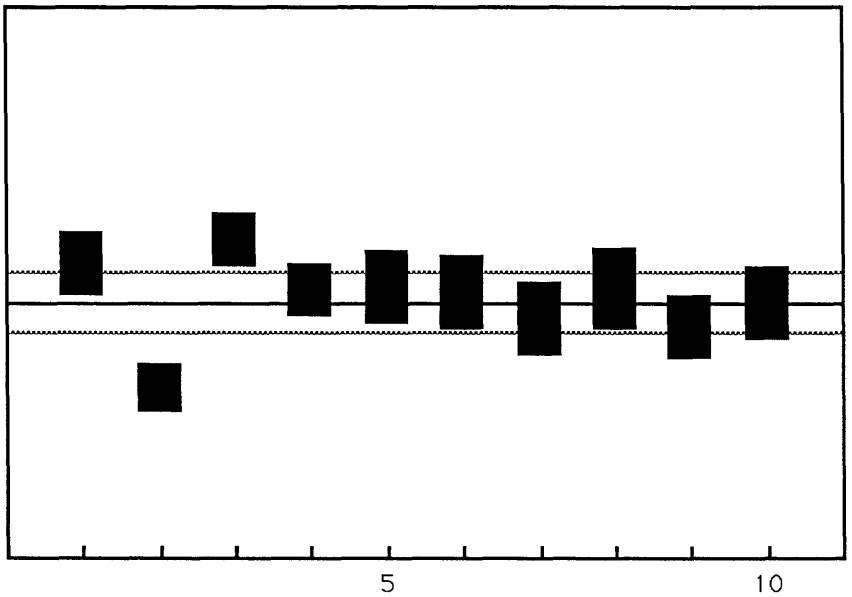

Figure 1. Final plot of an analysis showing the grand mean (solid line) and ±2σ errors of the mean (dotted lines). Each ratio is plotted as its 2σ error bar.

At the conclusion of an analysis the screen will display a plot (Figure 1) showing the means and 2σ errors for each block. This plot can be dumped to the printer if desired. At this point, the program will ask the user how many blocks to reject. A non-zero answer to this query will be followed by a series of questions asking the user which blocks to reject. The program will continue in this mode allowing the user to re-edit the data over and over (and printing the results) until the user enters "0" for the number of blocks to be rejected.

Using the General program option

The General program allows the user to measure any element but will not perform any mass discrimination correction or spike subtraction; it merely reports raw values. After prompting the user for a sample name, the General program asks the user for an element to be analyzed (version 1.1 supports U, Th, Pb, Re, Rb, K, Sr, and Ca). If the element selected has more than two isotopes the program will ask the user to choose the number of isotopes he/she wishes to measure. The user must then select the isotopes to be measured from a list presented on screen. Note that the reference isotope is the isotope that will be reported in the denominator of the ratios reported. For example, if one wishes to receive data as $^{40}Ca/^{42}Ca$ then ^{42}Ca should be chosen as the reference isotope. After choosing the desired isotopes, the General program operates the same as the other options.

Scanning the magnetic field

The `Scan` soft key is available whenever the user is in the peak adjustment routine. Selecting the `Scan` key brings up a new set of soft key definitions:

```
Start Scan                    LO field=1450  SCAN UP
LOG                           HI field=3210                Escape
```

When the scan routine is operating, the voltmeter integration time is set to 0.5 second and the readings are displayed on the screen. The backgrounds should be between 5.2 and 5.3 mV for proper scan display (5 mV is subtracted before plotting). The LO and HI field keys display the lowest and highest magnetic field values to be scanned. The initial values are determined from the current peak settings assuming that the user wishes to scan over the whole mass range being measured. These values may be changed by simply pressing the LO or HI field key and entering a new value. The SCAN UP key indicates that the scan will be from the lowest to the highest field value or up mass. To scan down mass simply press the SCAN UP key, which will then become SCAN DOWN. Note that this key indicates the current setting rather than the setting that will occur if pressed. The LOG key, which operates in a similar manner, indicates that the scan will be plotted with a logarithmic scale. Pressing the LOG key changes it to LINEAR and brings up two more soft key definitions, labelled Min=0V and Max=5V. These two keys refer to the minimum and maximum limits for the plot. They operate just like the LO and HI field keys; note that the values are in volts rather than millivolts.

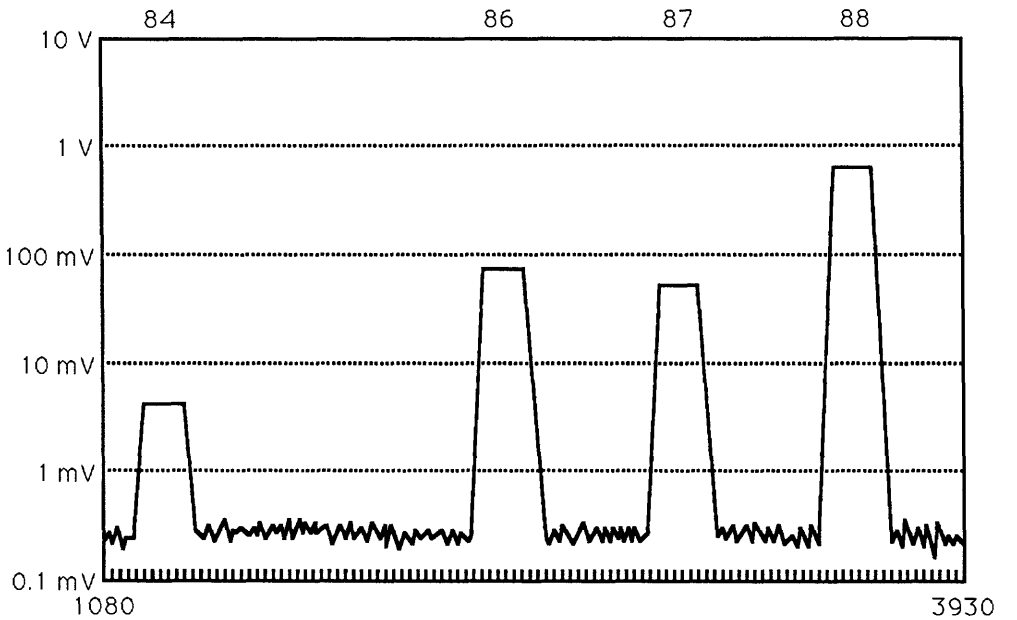

Figure 2. Simulation of plot showing magnetic field scan of a strontium mass spectrum. The mass numbers of the peaks are shown at top and the tick marks at the bottom are shown every 20 digital units.

Once these various options have been selected, pressing the Start Scan key switches the magnet to the first field value (LO or HI field depending on the scan direction) and waits four seconds for the magnet to stabilize before proceeding with the scan. Two soft keys labelled Slower and Faster are active during a scan; they lower or raise the scan speed by factors of two, limited by 1 and 32 as the lowest and highest number of digits per step. The default speed is 4 digits per step. The magnetic field value is displayed in a soft key label during the scan. The Escape key returns the user to the peak adjustment routine. If a scan is completed (the Escape key is not pressed) a soft key labelled Dump Graphics allows the scan to be printed on the printer, (Figure 2).

Appendices

Algorithms

Many users express an interest in learning about the methods used to acquire and reduce the data. This section explains in simple terms the various algorithms used in CONTROL. During data acquisition the backgrounds (measured on each side of each peak) and peak tops are integrated for a given length of time, typically 5 seconds. After every integration the data is edited for outliers. First, data outside 3σ are rejected on the basis of their extremely low probability. Then Chauvenet's criterion is used to reject any data with a probability of occurrence of less than 0.5 (Taylor, 1982). Since backgrounds are measured on both sides of a peak, these two measurements are averaged to obtain the background for a given peak. Backgrounds are only measured before and after each block of peak top measurements, so after a block of data (backgrounds, peak tops, backgrounds) is obtained the backgrounds are extrapolated linearly to the times of measurement of each peak and subtracted from the peak top values to obtain the net peak values. The net peak values are used to obtain ratios using Dodson's (1978) algorithm which corrects for first and second order fluctuations of the ion beam. These raw ratios are then checked for outliers by fitting the ratios to a straight line (in case of varying mass discrimination during a block) and editing them twice as described above for the integrations. The errors estimated for each ratio are used to estimate the internal error for the block. For calcium and strontium measurements, mass discrimination is corrected using an exponential law (Wasserburg *et al.*, 1981). This complicates the spike subtraction routine which must use an iterative calculation to estimate the mass discrimination and sample/spike ratio.

Source code description

Since the program files are not extensively commented, this section describes briefly the meaning of all variables and the operation of all subprograms. These subprograms contain the essential data acquisition, analysis and display routines; they are stored in a separate file called "Subfile" and are loaded into memory by the main programs. The element-specific subroutines such as the default field values, spike ratios, and mass discrimination correction are contained in the program file for that element.

Variables. The following variables are declared at the beginning of each module (KBDM, CABDM, RBBDM, SRBDM, and IRBDM). They are thus global variables; additional variables are allocated by the various subprograms as discussed below.

Name	Type	Usage
A$	string[4]	="DATA" first 4 characters of disk label of data storage disk
Block$	string[8]	key label specifying current block and total blocks
Elem$	string[2]	abbreviated name of element
Element$	string[10]	full name of element
File$	string[8]	name for data storage file on disk
N$	string[6]	disk label of data storage disk
Sample$	string[35]	sample designation
Traw$	string array[8]	times of each ratio measurement (HH:MM:SS)
B	integer	loop counter
Block	integer	current block number
Blockmax	integer	total number of blocks to be acquired
Center	integer	peak number of peak to be used for centering

Check	integer(T/F)	true if any run parameters in Parameters sub changed
Delay	integer array	delay times in seconds for each peak
Edit1	integer	number of blocks to be edited
Factor	integer(T/F)	true if total printout desired
False	integer(T/F)	=0; boolean value for false
Halt	integer(T/F)	true if run is paused between blocks
I	integer	loop counter
Index1	integer	subscript of array to be plotted in Summ_plot sub
Index2	integer	subscript of array where deviation is stored
Interference	integer(T/F)	true if an interfering peak is to be corrected for
Intn	integer array	integration times for peaks in seconds
Iter	integer	number of iterations used in spike subtraction routine
Mass	integer array	mass numbers of peaks
Numpeak	integer	number of peaks to be measured
Numsel	integer	Number of isotopes to select from (General option only)
P_mass	integer array	possible mass numbers (General option only)
P_pot	integer array	possible peak field values (General option only)
Params	integer(T/F)	true if run paused and user requested run parameters
Pot	integer array	magnetic field values for peaks
Prog	integer	type of sample (program) to be run (spiked vs unspiked)
Rejected	integer array	number of values rejected from each ratio
Scan	integer	number of scans per block
Sel	integer	number of next isotope to measure (General option only)
Sp	integer	number corresponding to spike selected by user
Timbegin	integer	time at start of data acquisition (minutes)
Timfirst	integer	time at start of bock since start of run (minutes)
Timlast	integer	time at finish of block since start of run (minutes)
True	integer(T/F)	=1; boolean value for true
Alpha	real	approximation of spike to sample ratio
Amu	real array	atomic mass of each isotope
Avetime	real array	average time of measurement for 4 peaks used for a ratio
Baseline	real array	values of baselines before and after each block
Beta	real	approximation of normalization ratio
Disc	real	discrimination per mass unit
Ipeak	real array	value of peak used to correct for interference
Itime	real array	time of measurement of interference peak
Mean	real array	block to block average for each ratio
Meancorr	real array	discrimination corrected ratio means within block
Meanraw	real array	raw ratio means within block
Netpeak	real array	baseline corrected peak values
P_exp	real	incremental mass discrimination exponent
Peak	real array	peak values
Percent_dev	real array	percent deviations (1 sigma) for each ratio within block
Qpeak	real array	peak values from quick scan prior to centering peaks
Rcorr	real array	corrected ratios for all blocks
Rraw	real array	raw ratios for each block
Rspike	real array	spike ratios for selected spike
Rstore	real array	array used for storing data on disk
Scantime	real	estimate of time for each block of peak scans in minutes
Sigma	real array	block to block 2-sigma error estimates for each ratio

Sigma_int	real	block to block estimate of total internal error (2-sigma)
Sum_p	real	total mass discrimination exponent
Tcorr	real array	temporary corrected ratios used in spike subtraction routine
Timebase	real array	times of measurement of baselines
Timep	real array	times of measurement of peaks
X_gdu_max	real	horizontal dimension of screen in graphic units
Y_gdu_max	real	vertical dimension of screen in graphic units

Subprograms. The subprograms are listed here in alphabetical order as they are in the Subfile module. The names of subprograms are bold and the variable names are shown in italics. each subprogram is described in three parts: a sentence describing its purpose, another sentence specifying the context in which it is called, and a paragraph describing its operation.

Name: **Center_peaks**
Purpose: centers the peaks by adjusting their magnetic field values
Context: called by main program before measuring peaks
Description: This subprogram first calculates a value (*Side*) by which to adjust the magnetic field so that a position on the side of the peak is obtained. This value is calibrated empirically. It then loops through the peaks in order waiting 4 seconds between each jump. It stores the peak values (*Qpeak*) for use in scaling the plotting routines. This loop also reduces hysteresis effects. During the **Center_peaks** routine the integration time of the voltmeter is changed to 0.5 second. The main loop repeats until an acceptable center position on the centering peak is obtained. This is done by measuring values on the low side, the top, and the high side of the centering peak. A case statement then uses the ratio of the differences from the top to each side to determine how much to adjust the field closer to the center position. These amounts are calibrated empirically. The subprogram then adjusts the magnetic field values for each peak (*Pot*) for the total amount of adjustment calculated for the centering peak. After setting the magnetic field to the first peak position it returns control to the main program.

Name: **Clear_crt**
Purpose: clears both the alpha and graphics screens
Context: called by many other contexts

Name: **Dump**
Purpose: dumps graphics screen to the printer
Context: called by soft key from certain subprograms

Name: **Edit**
Purpose: edits blocks and calculates new means and sigmas after a run
Context: called by main program
Description: The main program, after asking the user how many blocks he/she wishes to edit from the run, calls **Edit** to accomplish the task. **Edit** queries the user as to which blocks he/she wants to edit and then calculates new weighted means. **Edit** then calls **Grandstats** to calculates new external 2-sigmas and the value for the internal 2-sigma for the remaining blocks. **Print_means** is called to print out the results. Note that **Edit** does not change the corrected ratios allowing the user to re-edit the data as many times as necessary, each time beginning with the whole data set.

Name: **Edit_routine**

Purpose: edits each integration for outliers
Context: called by **Measure_peaks**, **Measure_zeros**, and **Measure_inter**
Description: Peak and baseline values are obtained by integrating over a period of time, during which the voltmeter is sending readings to the computer. **Edit_routine** takes these values and returns the average after rejecting outliers. Outliers are rejected at two levels of significance. First, it is assumed that all acceptable values fall within 3 standard deviations of the mean value. Values outside 3 sigma are rejected. Second, the routine checks for values with a probability of occurrence of less than 0.5 (Chauvenet's criterion). In order to do this the program calculates a sigma multiplier (*Sigmult*) such that values outside *Sigmult* standard deviations will have a probability of occurrence of less than 0.5 and therefore can be rejected. *Sigmult* is calculated from a best-fit curve to data from probability tables. **Edit_routine** returns the mean after rejecting all outliers found. If the number of values either before or after rejection falls to 1, **Edit_routine** does not try to reject data but returns the mean.

Name: **Field_values**
Purpose: allows user to set magnetic field values, cycle and scan the magnet field
Context: called by main program
Description: **Field_values** is called at the beginning of a run to get the peak centers from the user. It can also be invoked during a run by aborting a block or by choosing to change the run parameters between blocks. This subprogram can cycle the magnet through the peak settings indefinitely and calls **Magnet_scan** to scan the magnetic field. The computer does not know where the peaks are initially, except that the approximate peak field values are retained as default values from the main program. The user must center the peaks manually using the computer before data acquisition. The program can be in this context indefinitely, during which time the magnetic field can be adjusted, cycled or set using soft keys. The knob will also change the magnetic field. When this subprogram is exited (via a soft key) the magnetic field values should be set for all peaks. Because an interference peak may be too small to easily center manually, its position is calculated based on the settings for the other peaks.

Name: **Final_print**
Purpose: prints the corrected ratios for all blocks at the end of a run
Context: called by the main program
Description: **Final_print** consists of a single loop through the number of blocks of data in order to print out the corrected ratios (*Rcorr*) for each block including the percent error and block number.

Name: **Grandstats**
Purpose: calculates block to block statistics
Context: called by main program and **Edit**
Description: **Grandstats** calculates the block to block or external statistics at the 2-sigma level. Individual blocks are weighted by the inverse square of their estimated percent error and weighted means are calculated. External errors are reported as twice the standard error of the mean.

Name: **Identify**
Purpose: identifies the element to be run and prompts the user for a sample name
Context: called by main program

Name: **Linear_regress**
Purpose: edits each ratio for outliers
Context: called by **Statistics**

Description: **Linear_regress** calculates the means and standard deviations for the raw ratios after rejecting outliers based on Chauvenet's criterion (see **Edit_routine**). It fits the raw ratios to a straight line as a function of their time of measurement because monotonically decreasing or increasing values for a ratio are frequently obtained due to changing mass discrimination during a block. The mean value is the same as would be calculated using a simpler routine, but the error will be more accurate (and lower) than would be calculated otherwise.

Name: **Magnet_scan**
Purpose: scans the magnetic field following users directions
Context: called by **Field_values**
Description: **Magnet_scan** allows the user to scan the magnetic field over a limited gauss range covering the peaks to be measured. The scans are output to the screen and can be dumped to the printer if desired. The user may select (via soft keys) the range of field values to be scanned, whether to scan up field or down field, to draw the scan in log or linear units and if linear then what scale to draw the plot at. The speed of the scan (in field units per integration time) can also be changed during the scan. The integration time of the voltmeter is changed to 0.5 second until the program leaves the **Magnet_scan** routine. The program remains in the **Magnet_scan** routine until the user escapes by pressing a soft key.

Name: **Measure_inter**
Purpose: measures the peak of an interfering element
Context: called by main program
Description: **Measure_inter** measures an isotope peak for an interfering element. It is only implemented for measuring the rubidium interference when measuring strontium samples. The interfering peak is measured after the baselines but before the peak tops. The interfering peak is measured only once between blocks as are the baselines (except for the first block or whenever the user pauses the program to change parameters). For a given block, the interfering peak value and its time of measurement are *Ipeak(1)* and *Itime(1)* for before the block and *Ipeak(2)* and *Itime(2)* after the block. Before each block **New_zeros** is called to set the baselines and interfering peak values to the values measured after the previous block.

Name: **Measure_peaks**
Purpose: measures the peak tops
Context: called by main program
Description: **Measure_peaks** measures the peak tops integrating each for the time specified by the user (stored in *Intn*). Standard graphics statements allow the peak values to be shown on the screen. Approximate baseline subtracted values are shown on the screen and plotted as averages. A soft key is enabled during acquisition of peak tops to allow the user to abort the block. The program then invokes the Showdefaults subroutine which shows the current run parameters and calls **Field_values** to allow the user to make any adjustments to the mass spectrometer as necessary.

Name: **Measure_zeros**
Purpose: measures the baselines on both sides of each peak including interfering peak
Context: called by main program
Description: **Measure_zeros** first calculates the magnetic field offset from the peak at which to measure baselines for the peaks. The value in *Del* is the offset in mass

units. On the NBS 6" mass spectrometers this value is about 0.3 mass units for potassium and calcium and nearly 0.5 mass units for rubidium and strontium. The offset in magnetic field units is calculated from the positions (field values) and masses of the first two peaks. Each baseline is integrated for twice the time as its associated peak after a delay equal to the delay before measuring the first peak. To minimize large jumps in the magnetic field, the baseline data is acquired either continuously down mass or up mass depending on the order of peak measurement. If an interfering peak is being measured, its baselines are always measured first.

Name: **New_zeros**
Purpose: sets values of baselines measured after peak tops to values for before peak tops
Context: called by main program
Description: **New_zeros** simply takes the values of the baselines (*Baseline*) and their times of measurement (*Timebase*) from after the last block and sets the values for before the next block equal to them. This allows the baselines to be measured only once between blocks. The interference peak is handled the same way.

Name: **Parameters**
Purpose: displays current run parameters and allows user to change them
Context: called by main program from Showdefaults subroutine
Description: **Parameters** allows the user to view and change the following: total number of blocks to be acquired, the number of scans per block, the peak integration times and delay times, and whether or not all raw ratios are to be printed out or just the means. If strontium is being analyzed, the user can decide whether or not rubidium is to be measured to correct for interference. If potassium or rubidium is being analyzed then the user may change the mass discrimination correction to be applied to the data.

Name: **Plot**
Purpose: sets up the screen to plot the peak tops during a block
Context: called by main program
Description: **Plot** uses standard graphics commands to draw axes and label them for the plot of peak intensities vs time since start of run. The maximum value for the graph is calculated from the maximum peak value obtained by the **Center_peaks** subprogram.

Name: **Print_means**
Purpose: prints block to block means and external errors
Context: called by main program (Printout3 and Summary subroutines) and **Edit**
Description: Once two blocks have been acquired, **Print_means** prints the weighted means and external 2-sigma errors of the means. It is also called after data acquisition to print the final means and is called again to print the means obtained after editing the data.

Name: **Print_out1**
Purpose: prints peak and baseline intensities, magnet field values, and raw ratios
Context: called by main program
Description: **Print_out1** prints the following information after each block: baseline intensities(mV) both before and after the block, initial and final peak intensities (mV), magnet field values, and if the user requested the raw ratios and times of measurement of each ratio.

Name: **Print_out2**
Purpose: prints means of raw ratios, percent deviations and number rejected
Context: called by main program

Description: **Print_out2** prints the following information after each block: means of the raw ratios, percent deviations (1-sigma) of the means, and the number of ratios that were rejected.

Name: **Ratios**
Purpose: calculates the net peak intensities, raw ratios and the times of measurement
Context: called by main program
Description: **Ratios** first calculates the rate of change of the baselines for each peak. Using this rate, the baselines are extrapolated linearly to the times of measurement of each peak and subtracted from the peak values to obtain the net peak values (*Netpeak*). Any interfering peak correction is made in a similar way and subtracted from the appropriate *Netpeak* value if significant. The rate of change of the net peaks is then calculated. Using this rate each two pairs of peak measurements is extrapolated linearly to their mean time of measurement (*Avetime*) and their ratio is obtained at this time. This method is used because it inherently corrects for second order rates of change of the peaks (Dodson, 1978). There will be one less ratio obtained than the number of scans measured.

Name: **Start_up**
Purpose: initializes variables and starts data acquisition
Context: called by main program
Description: **Start_up** asks the user to be sure the printer is on line and then prints the sample name and the date. *Timbegin* is initialized to the current time and *Block* is set to 1.

Name: **Statistics**
Purpose: calculates the mean and percent deviations of the raw ratios
Context: called by main program
Description: **Statistics** first normalizes the times of measurement by subtracting a constant time from each. Then it calls **Linear_regress** to obtain the regression of the raw ratios as a function of their time of measurement. It then calculates the standard deviation (1-sigma) of the means and returns the percent deviations as well as the mean values.

Name: **Summ_plot**
Purpose: plots the corrected ratio of interest for each block on the screen
Context: called by Summary subroutine of main program
Description: **Summ_plot** starts by obtaining the corrected ratios for the main ratio of interest and its associated error and storing these as well as the associated mean and error values in temporary variables. It then plots a graph on the screen with an unlabeled y-axis corresponding to the value of the corrected ratio and an x-axis with the block numbers labeled. The minimum value for the plot is calculated from the minimum value of the corrected ratio minus the maximum error. The maximum value for the plot is similarly obtained from the maximum ratio and the maximum error. The mean is plotted as a solid horizontal line and 2-sigma error of the mean is shown by two dashed horizontal lines. Soft keys allow the user to dump the graphic screen to the printer or continue. This plot is very helpful in examining the data during editing.

Name: **Weird_event**
Purpose: beeps when certain soft keys are pressed
Context: called by many other contexts
Description: **Weird_event** simply returns a beep sound when a soft key is pressed that does not do anything. Since some of the soft key labels are used to

communicate information to the operator, these keys are not supposed to trigger an event. The **Weird_event** subprogram provides a way to effectively disable a soft key but still allow its label to appear on the screen.

Source Code Listings. The complete listings of the source code for CONTROL 1.1 are shown below. These are shown primarily for documentation purposes. The program files are available on disk directly from the author at U.S. Geological Survey, MS 963, Denver, CO 80225.

CONTROL

```
10    REM CONTROL    Version 1.1              Brian D. Marshall  7/88
20    REM This program and all associated modules constitute CONTROL.
30    PRINTER IS CRT
40    DISP CHR$(128);
50    INTEGER True,False
60    DIM Name$[6]
70    False=0
80    True= NOT False
90    GCLEAR
100   OUTPUT KBD;CHR$(255)&CHR$(75);
110   PRINT USING "10(/)"
120   PRINT "USGS Computer Program 'CONTROL'. Version 1.100;"
130   PRINT "Technical Contact: Brian Marshall."
140   ON CYCLE 1,2 GOSUB Update
150   ON KEY 9 LABEL "Shut Down" GOTO Exit
160   ON KEY 0 LABEL "Potassium" GOTO Getk
170   ON KEY 1 LABEL "Calcium" GOTO Getca
180   ON KEY 2 LABEL "Rubidium" GOTO Getrb
190   ON KEY 4 LABEL "General" GOTO Getir
200   ON KEY 3 LABEL "Strontium" GOTO Getsr
210   ON KEY 6 LABEL "Setup DVM" GOSUB Getset
220   ON KEY 5 LABEL "Erase Disk" GOSUB Initdisk
230   WHILE True
240   END WHILE
250 Update:!
260   PRINT TABXY(1,1);DATE$(TIMEDATE)
270   PRINT
280   PRINT TIME$(TIMEDATE)
290   RETURN
300 Getk:!
310   LOAD "KBDM"
320 Getca:!
330   LOAD "CABDM"
340 Getrb:!
350   LOAD "RBBDM"
360 Getir:!
370   LOAD "IRBDM"
380 Getsr:!
390   LOAD "SRBDM"
400 Getset:!
410   DISP "Setting up digital voltmeter... please wait."
420   DISABLE
430   REMOTE 716
440   LOCAL LOCKOUT 7
450   OUTPUT 716;"INITIALISE"
460   WAIT 3
470   OUTPUT 716;"LITERALS OFF"
480   OUTPUT 716;"ITIME USER=1"
490   OUTPUT 716;"TRACK ON"
500   OUTPUT 716;"RESUME ON"
510   ENABLE
520   DISP CHR$(12);
530   RETURN
```

```
540 Initdisk:!
550   DISABLE
560   ON ERROR GOTO Done
570   DISP "Insert a disk to be ERASED into LEFT drive and press Continue."
580   PAUSE
590   DISP "Initializing disk...please wait."
600   INITIALIZE ":,700,0"
610   INPUT "Name (6 characters beginning with a letter) for this disk?",Name$
620   PRINT LABEL Name$ TO ":,700,0"
630 Done:!
640   OFF ERROR
650   ENABLE
660   DISP CHR$(12);
670   RETURN
680 Exit:!
690   OFF CYCLE
700   OUTPUT KBD;CHR$(255)&CHR$(75);
710   OFF KEY
720   OUTPUT 716;"TRACK OFF"
730   OUTPUT 716;"DISPLAY [                    ]"
740   END
```

Subfile

```
10    REM Subfile              Brian D. Marshall  4/88
20    SUB Center_peaks(Qpeak(*),INTEGER Numpeak,Interference,Mass(*),Pot(*))
30      OPTION BASE 1
40      INTEGER Delta,J,Sumdelta,K,Side,Center
50      REAL V(3)
60      DIM F$[4]
70      DISP CHR$(128);
80      OUTPUT 716;"ITIME USER=.5"
90      FOR J=1 TO Numpeak+Interference
100       OUTPUT 12 USING "#,W";Pot(J)
110       DISP "Quick scan: ";Mass(J);" = ";Pot(J)
120       WAIT 4
130       OUTPUT 716;"HISTORY CLEAR"
140       ENTER 716;Qpeak(J)
150     NEXT J
160     FOR J=1 TO Numpeak+Intererence
170       IF MAX(Qpeak(*))=Qpeak(J) THEN Center=J
180     NEXT J
190     OUTPUT 12 USING "#,W";Pot(Center)
200     DISP CHR$(129);
210     Side=PROUND(ABS(Pot(1)-Pot(2))/ABS(Mass(1)-
Mass(2))/450*(.684*Mass(Center)+12.6),0)
220     WAIT 4
230     Sumdelta=0
240     REPEAT
250       DISP "Centering mass ";Mass(Center);"    Field= ";Pot(Center);"    New Field=
";Pot(Center)+Sumdelta
260       FOR J=-1 TO 1
270         OUTPUT 12 USING "#,W";Pot(Center)+Sumdelta+J*Side
280         WAIT 3
290         OUTPUT 716;"HISTORY CLEAR"
300         ENTER 716;V(J+2)
310       NEXT J
320       SELECT (V(2)-V(1))/(V(2)-V(3))
330       CASE >4
340         Delta=PROUND(Side/40,0)*4
350       CASE <.25
360         Delta=PROUND(Side/40,0)*(-4)
370       CASE >2
380         Delta=PROUND(Side/40,0)*2
390       CASE <.5
400         Delta=PROUND(Side/40,0)*(-2)
```

```
410       CASE >1.2
420         Delta=PROUND(Side/40,0)
430       CASE <.83
440         Delta=PROUND(Side/40,0)*(-1)
450       CASE ELSE
460         Delta=0
470       END SELECT
480       Sumdelta=Sumdelta+Delta
490     UNTIL Delta=0
500     FOR K=1 TO Numpeak+Interference
510       Pot(K)=Pot(K)+Sumdelta
520     NEXT K
530     OUTPUT 12 USING "#,W";Pot(Center)
540     OUTPUT 716;"ITIME USER=1"
550     DISP CHR$(12);
560   SUBEND
570   SUB Clear_crt
580     OUTPUT KBD;CHR$(255)&CHR$(75);
590     OUTPUT KBD;CHR$(255)&CHR$(84);
600     GCLEAR
610   SUBEND
620   SUB Dump
630     DISABLE
640     DISP CHR$(128);"Dumping graphics...please wait."
650     DUMP GRAPHICS
660     ENABLE
670     PRINT
680     DISP CHR$(12);
690   SUBEND
700   SUB Edit(R(*),Mean(*),Sigma(*),Sigma_int,INTEGER Nedit,Nratio,Blockmax)
710     OPTION BASE 1
720     ALLOCATE W(Blockmax),Rtemp(7,Blockmax)
730     INTEGER N,Bnum,K,Nblock
740     FOR N=1 TO Blockmax
750       W(N)=N
760     NEXT N
770     FOR N=1 TO Nedit
780       INPUT "Number of next block to edit?",Bnum
790       PRINT Bnum;
800       W(Bnum)=0
810     NEXT N
820     PRINT "rejected."
830     Nblock=0
840     FOR N=1 TO Blockmax
850       IF W(N)=0 THEN Gone
860       Nblock=Nblock+1
870       FOR K=1 TO Nratio
880         Rtemp(K,Nblock)=R(K,N)
890       NEXT K
900       Rtemp(7,Nblock)=R(7,N)
910 Gone: !
920     NEXT N
930     CALL Grandstats(Rtemp(*),Mean(*),Sigma(*),Sigma_int,Nratio,Nblock)
940     CALL Print_means(Mean(*),Sigma(*),Sigma_int,Nblock,Nratio)
950   SUBEND
960   SUB Edit_routine(Values(*),Cmean,INTEGER Num,Rej)
970     INTEGER K,J
980     REAL Sigmult,Sumvalues,Sumd,Stdd,D
990     Rej=0
1000    GOSUB Mean_sd
1010    Sigmult=3
1020    GOSUB Edit
1030    GOSUB Mean_sd
1040    Sigmult=1.16+(Num-Rej)*.0988-(Num-Rej)^2*.00214
1050    GOSUB Edit
1060    GOSUB Mean_sd
```

```
1070    SUBEXIT
1080 Mean_sd:!
1090    Sumvalues=0
1100    FOR K=1 TO Num-Rej
1110      Sumvalues=Sumvalues+Values(K)
1120    NEXT K
1130    Cmean=Sumvalues/(Num-Rej)
1140    IF Num-Rej<2 THEN SUBEXIT
1150    Sumd=0
1160    FOR K=1 TO Num-Rej
1170      Sumd=Sumd+(Values(K)-Cmean)^2
1180    NEXT K
1190    Stdd=SQR(Sumd/(Num-Rej-1))
1200    RETURN
1210 Edit:!
1220    J=0
1230    FOR K=1 TO Num-Rej
1240      D=Values(K)-Cmean
1250      IF ABS(D)>Sigmult*Stdd THEN
1260        Rej=Rej+1
1270      ELSE
1280        J=J+1
1290        Values(J)=Values(K)
1300      END IF
1310    NEXT K
1320    RETURN
1330  SUBEND
1340  SUB Field_values(X_gdu_max,Y_gdu_max,INTEGER Numpeak,Interference,Mass(*),Pot(*))
1350    INTEGER Field,True,I
1360    REAL Dvm
1370    DIM F$[4]
1380    CALL Clear_crt
1390    True=-1
1400    OFF KEY
1410    DISP CHR$(128);
1420    ON KEY 9 LABEL "Escape",2 GOTO Done
1430    ON KEY 0 LABEL "Step",2 GOSUB Step
1440    ON KEY 1 LABEL "UP",2 GOSUB Inc
1450    ON KEY 6 LABEL "DOWN",2 GOSUB Dec
1460    ON KEY 2 LABEL "Set",2 GOSUB Set
1470    ON KEY 5 LABEL "Cycle",2 GOSUB Cycle
1480    ON KEY 3 LABEL "Field",2 CALL Weird_event
1490    ON KEY 7 LABEL "Scan",2 GOSUB Scan
1500    I=1
1510    OUTPUT 12 USING "#,W";Pot(I)
1520    DISP USING "3D,"" = "",4Z";Mass(I);Pot(I)
1530    Field=Pot(I)
1540    GOSUB Update_field
1550    ON KNOB .1,3 GOSUB Turn
1560    WHILE True
1570    END WHILE
1580 Step:!
1590    I=I MOD (Numpeak+Interference)+1
1600    OUTPUT 12 USING "#,W";Pot(I)
1610    DISP USING "3D,"" = "",4Z";Mass(I);Pot(I)
1620    Field=Pot(I)
1630    GOSUB Update_field
1640    RETURN
1650 Inc:!
1660    Field=Field+1
1670    OUTPUT 12 USING "#,W";Field
1680    GOSUB Update_field
1690    RETURN
1700 Dec:!
1710    Field=Field-1
1720    OUTPUT 12 USING "#,W";Field
```

```
1730    GOSUB Update_field
1740    RETURN
1750 Set:!
1760    Pot(I)=Field
1770    DISP USING "3D,"" = "",4Z";Mass(I);Pot(I)
1780    RETURN
1790 Turn:!
1800    Field=Field+KNOBX
1810    IF Field<0 THEN Field=0
1820    IF Field>4095 THEN Field=4095
1830    OUTPUT 12 USING "#,W";Field
1840    GOSUB Update_field
1850    RETURN
1860 Cycle:!
1870    OFF KEY
1880    OFF KNOB
1890    GOSUB Update_field
1900    ON KEY 3 LABEL "Field",2 CALL Weird_event
1910    ON KEY 5 LABEL "Hold",4 GOTO Stop
1920    ON CYCLE 5,3 GOSUB Step
1930    WHILE True
1940    END WHILE
1950 Stop:!
1960    ON KEY 5 LABEL "Cycle",2 GOSUB Cycle
1970    ON KEY 7 LABEL "Scan",2 GOSUB Scan
1980    ON KEY 2 LABEL "Set",2 GOSUB Set
1990    ON KEY 6 LABEL "DOWN",2 GOSUB Dec
2000    ON KEY 1 LABEL "UP",2 GOSUB Inc
2010    ON KEY 0 LABEL "Step",2 GOSUB Step
2020    ON KEY 9 LABEL "Escape",2 GOTO Done
2030    ON KNOB .1,3 GOSUB Turn
2040    OFF CYCLE
2050    RETURN
2060 Update_field:!
2070    F$=VAL$(Field)
2080    ON KEY 8 LABEL F$,2 CALL Weird_event
2090    RETURN
2100 Scan:!
2110    OFF KNOB
2120    CALL Magnet_scan(X_gdu_max,Y_gdu_max,Numpeak,Interference,Mass(*),Pot(*))
2130    OUTPUT 12 USING "#,W";Pot(I)
2140    DISP USING "3D,"" = "",4Z";Mass(I);Pot(I)
2150    ON KNOB .1,3 GOSUB Turn
2160    RETURN
2170 Done:!
2180    OFF KNOB
2190    OUTPUT 12 USING "#,W";Pot(1)
2200    IF Interference THEN Pot(5)=(Pot(4)+Pot(3))/2
2210  SUBEND
2220  SUB Final_print(S$,Rcorr(*),INTEGER Ncorr,Blockmax)
2230    INTEGER N,R
2240    PRINT USING "@"
2250    PRINT
2260    PRINT "U.S.G.S.  B.I.G.  #1 6"""
2270    PRINT DATE$(TIMEDATE)
2280    PRINT TIME$(TIMEDATE)
2290    PRINT TAB(20);S$
2300    PRINT
2310    FOR N=1 TO Blockmax
2320      FOR R=1 TO Ncorr
2330        PRINT USING "2X,3D.7D,#";Rcorr(R,N)
2340      NEXT R
2350      PRINT
2360      PRINT
2370    NEXT N
2380  SUBEND
```

```
2390  SUB Grandstats(R(*),Mean(*),Sigma(*),Sigma_int,INTEGER Nratio,Nblock)
2400    OPTION BASE 1
2410    IF Nblock=1 THEN SUBEXIT
2420    INTEGER N,K
2430    ALLOCATE Sumdev(Nratio),Weight(Nblock)
2440    REAL Sum_weight
2450    Sum_weight=0
2460    FOR N=1 TO Nblock
2470      Sum_weight=Sum_weight+R(7,N)
2480    NEXT N
2490    Sigma_int=1/SQR(Sum_weight)
2500    FOR N=1 TO Nblock
2510      Weight(N)=R(7,N)/Sum_weight
2520    NEXT N
2530    FOR K=1 TO Nratio
2540      Mean(K)=0
2550      FOR N=1 TO Nblock
2560        Mean(K)=Mean(K)+R(K,N)*Weight(N)
2570      NEXT N
2580      Sumdev(K)=0
2590      FOR N=1 TO Nblock
2600        Sumdev(K)=Sumdev(K)+(R(K,N)-Mean(K))^2
2610      NEXT N
2620      Sigma(K)=SQR(Sumdev(K)/(Nblock*(Nblock-1)))
2630    NEXT K
2640  SUBEND
2650  SUB Identify(Elem$,Sample$,X_gdu_max,Y_gdu_max)
2660    DISP CHR$(128);
2670    GINIT
2680    GRAPHICS ON
2690    CSIZE 50
2700    MOVE X_gdu_max/2,Y_gdu_max/2
2710    LORG 5
2720    LABEL Elem$
2730    LINPUT "Sample name (<=35 characters)?",Sample$
2740  SUBEND
2750  SUB Linear_regress(Times(*),Values(*),Cmean,Std_d,INTEGER Num,Rej)
2760    INTEGER K,J,N
2770    REAL Slope,Intcpt,Sigmult,Sumx,Sumx2,Sumy,Sumxy,Sumd,D
2780    Rej=0
2790    GOSUB Mean_sd_lr
2800    J=0
2810    Sigmult=1.16+Num*.0988-Num^2*.00214
2820    FOR K=1 TO Num
2830      D=Values(K)-Intcpt-Slope*Times(K)
2840      IF ABS(D)>Sigmult*Std_d THEN
2850        Rej=Rej+1
2860      ELSE
2870        J=J+1
2880        Values(J)=Values(K)
2890        Times(J)=Times(K)
2900      END IF
2910    NEXT K
2920    IF Rej=0 OR Num-Rej<2 THEN SUBEXIT
2930    GOSUB Mean_sd_lr
2940    SUBEXIT
2950  Mean_sd_lr:!
2960    Sumx=0
2970    Sumx2=0
2980    Sumy=0
2990    Sumxy=0
3000    N=Num-Rej
3010    FOR K=1 TO N
3020      Sumx=Sumx+Times(K)
3030      Sumx2=Sumx2+Times(K)^2
3040      Sumy=Sumy+Values(K)
```

```
3050      Sumxy=Sumxy+Values(K)*Times(K)
3060    NEXT K
3070    Cmean=Sumy/N
3080    Intcpt=(Sumx2*Sumy-Sumx*Sumxy)/(N*Sumx2-Sumx^2)
3090    Slope=(N*Sumxy-Sumx*Sumy)/(N*Sumx2-Sumx^2)
3100    Sumd=0
3110    FOR K=1 TO N
3120      Sumd=Sumd+(Values(K)-Intcpt-Slope*Times(K))^2
3130    NEXT K
3140    Std_d=SQR(Sumd/(N-2))
3150    RETURN
3160  SUBEND
3170  SUB Magnet_scan(X_gdu_max,Y_gdu_max,INTEGER Numpeak,Interference,Mass(*),Pot(*))
3180    INTEGER Offset,Begin,End,Speed,True,Field,Scanup,First,Last,Tem,Logscale
3190    INTEGER Start
3200    REAL Dvm,Dplot,Del,Upper,Lower,Temp,L
3210    DIM B$[14],E$[14],F$[4],L$[6],U$[10],L1$[10],S$[10]
3220    OFF KEY
3230    ON KEY 9 LABEL "Escape",3 GOTO Done_it
3240    True=1
3250    Del=.00368*Mass(1)+.153
3260    Offset=PROUND(ABS(Pot(1)-Pot(2))/ABS(Mass(1)-Mass(2))*Del+20,0)
3270    Begin=MIN(Pot(*))-Offset
3280    End=MAX(Pot(*))+Offset
3290    GOSUB Reduce
3300    Speed=4
3310    Scanup=True
3320    Logscale=True
3330    Upper=5
3340    Lower=0
3350    GOSUB Scale
3360    GOSUB Scan_dir
3370    ON KEY 0 LABEL "Start Scan",3 GOSUB New
3380    OUTPUT 716;"ITIME USER=.5"
3390    OUTPUT 716;"HISTORY CLEAR"
3400    WHILE True
3410      ENTER 716;Dvm
3420      DISP USING "5D.3D,X,""mV""";Dvm*1000
3430    END WHILE
3440  New: !
3450    OFF KEY
3460    First=End+1
3470    IF Scanup THEN First=Begin-1
3480    Last=Begin
3490    IF Scanup THEN Last=End
3500    OUTPUT 12 USING "#,W";First
3510    GINIT
3520    CALL Clear_crt
3530    GRAPHICS OFF
3540    VIEWPORT .12*X_gdu_max,.96*X_gdu_max,.3*Y_gdu_max,.95*Y_gdu_max
3550    FRAME
3560    IF Logscale THEN
3570      WINDOW Begin,End,-1,4
3580      AXES 20,0,Begin,-1
3590      LINE TYPE 4
3600      FOR L=0 TO 3
3610        MOVE Begin,L
3620        PLOT End,L,-1
3630      NEXT L
3640      LINE TYPE 1
3650      CLIP OFF
3660      CSIZE 4
3670      LORG 6
3680      MOVE Begin,-1.05
3690      LABEL USING "4Z";Begin
3700      MOVE End,-1.05
```

```
3710        LABEL USING "4Z";End
3720        LORG 8
3730        L=-1
3740        MOVE Begin,L
3750        LABEL USING "1Z.1D,X,""mV""";10^L
3760        FOR L=0 TO 2
3770          MOVE Begin,L
3780          LABEL USING "3D,X,""mV""";10^L
3790        NEXT L
3800        FOR L=3 TO 4
3810          MOVE Begin,L
3820          LABEL USING "2D,X,""V""";10^L/1000
3830        NEXT L
3840        LORG 4
3850        FOR L=1 TO Numpeak+Interference
3860          MOVE Pot(L),4.05
3870          LABEL Mass(L)
3880        NEXT L
3890      ELSE
3900        WINDOW Begin,End,Lower,Upper
3910        AXES 20,0,Begin,Lower
3920        CLIP OFF
3930        CSIZE 4
3940        LORG 6
3950        MOVE Begin,Lower
3960        LABEL USING "4Z";Begin
3970        MOVE End,Lower
3980        LABEL USING "4Z";End
3990        LORG 8
4000        FOR L=Lower TO Upper STEP Upper-Lower
4010          MOVE Begin,L
4020          LABEL USING "2D.2D,X,""V""";L
4030        NEXT L
4040        LORG 4
4050        FOR L=1 TO Numpeak
4060          MOVE Pot(L),Upper+Upper*.002
4070          LABEL Mass(L)
4080        NEXT L
4090      END IF
4100      CLIP ON
4110      GRAPHICS ON
4120      ON KEY 3 LABEL "Field",3 CALL Weird_event
4130      ON KEY 9 LABEL "Escape",4 GOTO Done_it
4140      DISP "Waiting for magnet to stabilize..."
4150      WAIT 4
4160      OUTPUT 716;"HISTORY CLEAR"
4170      ENTER 716;Dvm
4180      DISP USING "5D.3D,X,""mV""";Dvm*1000
4190      Dplot=Dvm*1000-5
4200      IF Dplot<=0 THEN Dplot=.01
4210      IF Logscale THEN
4220        MOVE First,LGT(Dplot)
4230      ELSE
4240        MOVE First,Dplot/1000
4250      END IF
4260      Start=First+(2*Scanup-1)
4270      OUTPUT 716;"HISTORY CLEAR"
4280      ON KEY 1 LABEL "Slower",4 GOTO Slower
4290      ON KEY 6 LABEL "Faster",4 GOTO Faster
4300 Do_it:!
4310      FOR Field=Start TO Last STEP Speed
4320        OUTPUT 12 USING "#,W";Field
4330        F$=VAL$(Field)
4340        ON KEY 8 LABEL F$,3 CALL Weird_event
4350        ENTER 716;Dvm
4360        DISP USING "5D.3D,X,""mV""";Dvm*1000
```

```
4370      Dplot=Dvm*1000-5
4380      IF Dplot<=0 THEN Dplot=.01
4390      IF Logscale THEN
4400        PLOT Field,LGT(Dplot),-1
4410      ELSE
4420        PLOT Field,Dplot/1000,-1
4430      END IF
4440    NEXT Field
4450    OFF KEY
4460    ON KEY 0 LABEL "Start Scan",3 GOSUB New
4470    ON KEY 2 LABEL B$,3 GOSUB Newbegin
4480    ON KEY 7 LABEL E$,3 GOSUB Newend
4490    ON KEY 5 LABEL L$,3 GOSUB Newscale
4500    IF NOT Logscale THEN
4510      ON KEY 6 LABEL U$,3 GOSUB Newmax
4520      ON KEY 1 LABEL L1$,3 GOSUB Newmin
4530    END IF
4540    ON KEY 4 LABEL "Dump Graphics",3 CALL Dump
4550    ON KEY 9 LABEL "Escape",3 GOTO Done_it
4560    ON KEY 3 LABEL S$,3 GOSUB New_scan_dir
4570    RETURN
4580 Slower:!
4590    Speed=Speed/2
4600    IF ABS(Speed)<1 THEN Speed=1
4610    IF Scanup AND Speed<0 THEN Speed=-Speed
4620    IF NOT Scanup AND Speed>0 THEN Speed=-Speed
4630    Start=Field
4640    OUTPUT 716;"HISTORY CLEAR"
4650    GOTO Do_it
4660 Faster:!
4670    Speed=Speed*2
4680    IF ABS(Speed)>32 THEN Speed=32
4690    IF Scanup AND Speed<0 THEN Speed=-Speed
4700    IF NOT Scanup AND Speed>0 THEN Speed=-Speed
4710    Start=Field
4720    OUTPUT 716;"HISTORY CLEAR"
4730    GOTO Do_it
4740 Reduce:!
4750    IF Begin<0 THEN Begin=0
4760    IF End>4090 THEN End=4090
4770    Begin=PROUND(Begin,1)
4780    B$="LO field="&VAL$(Begin)
4790    End=PROUND(End,1)
4800    E$="HI field="&VAL$(End)
4810    ON KEY 2 LABEL B$,3 GOSUB Newbegin
4820    ON KEY 7 LABEL E$,3 GOSUB Newend
4830    RETURN
4840 Newbegin:!
4850    INPUT "New lower field value",Tem
4860    IF Tem<End THEN Begin=Tem
4870    GOSUB Reduce
4880    RETURN
4890 Newend:!
4900    INPUT "New upper field value",Tem
4910    IF Tem>Begin THEN End=Tem
4920    GOSUB Reduce
4930    RETURN
4940 Scale:!
4950    L$="LINEAR"
4960    IF Logscale THEN L$="LOG"
4970    ON KEY 5 LABEL L$,3 GOSUB Newscale
4980    IF NOT Logscale THEN
4990      U$="Max="&VAL$(Upper)&"V"
5000      L1$="Min="&VAL$(Lower)&"V"
5010      ON KEY 6 LABEL U$,3 GOSUB Newmax
5020      ON KEY 1 LABEL L1$,3 GOSUB Newmin
```

```
5030     ELSE
5040       OFF KEY 6
5050       OFF KEY 1
5060     END IF
5070     RETURN
5080 Newscale:!
5090     Logscale= NOT Logscale
5100     GOSUB Scale
5110     RETURN
5120 Newmin:!
5130     INPUT "New lower limit for plot (in volts)",Temp
5140     IF Temp<Upper THEN Lower=Temp
5150     GOSUB Scale
5160     RETURN
5170 Newmax:!
5180     INPUT "New upper limit for plot (in volts)",Temp
5190     IF Temp>Lower THEN Upper=Temp
5200     GOSUB Scale
5210     RETURN
5220 New_scan_dir:!
5230     Scanup= NOT Scanup
5240     Speed=-Speed
5250     GOSUB Scan_dir
5260     RETURN
5270 Scan_dir:!
5280     S$="SCAN DOWN"
5290     IF Scanup THEN S$="SCAN UP"
5300     ON KEY 3 LABEL S$,3 GOSUB New_scan_dir
5310     RETURN
5320 Done_it:!
5330     OUTPUT 716;"ITIME USER=1"
5340   SUBEND
5350   SUB Measure_inter(Block$,Baseline(*),Ipeak(*),Itime(*),INTEGER
Delay(*),Mass(*),Intn(*),Pot(*))
5360     OPTION BASE 1
5370     DIM K$[5],M$[3],F$[4]
5380     INTEGER T,P,Rej
5390     REAL Electrometer(60),Time(60),Sumtime,Cmean
5400     OFF KEY
5410     ON KEY 5 LABEL Block$ CALL Weird_event
5420     ON KEY 0 LABEL "Block" CALL Weird_event
5430     ON KEY 2 LABEL "Rubidium" CALL Weird_event
5440     ON KEY 3 LABEL "Field" CALL Weird_event
5450     P=5
5460     K$=VAL$(Mass(P))&"RB"
5470     M$=VAL$(Mass(P))
5480     F$=VAL$(Pot(P))
5490     OUTPUT 12 USING "#,W";Pot(P)
5500     ON KEY 7 LABEL M$ CALL Weird_event
5510     ON KEY 8 LABEL F$ CALL Weird_event
5520     DISP "Delay in progress, ";Rej;" readings were rejected last integration."
5530     WAIT Delay(P)
5540     Sumtime=0
5550     OUTPUT 716;"HISTORY CLEAR"
5560     FOR T=1 TO Intn(P)
5570       ENTER 716;Electrometer(T)
5580       Time(T)=TIMEDATE
5590       DISP USING "5A,""= "",5D.3D,"" mV at "",8A";K$;(Electrometer(T)-
Baseline(P,2))*1000;TIME$(Time(T))
5600       Sumtime=Sumtime+Time(T)
5610     NEXT T
5620     OUTPUT 12 USING "#,W";Pot(1)
5630     CALL Edit_routine(Electrometer(*),Cmean,(Intn(P)),Rej)
5640     Ipeak(2)=Cmean
5650     Itime(2)=Sumtime/Intn(P)
5660   SUBEND
```

```
5670  SUB Measure_peaks(Element$,Elem$,Block$,Baseline(*),Peak(*),Timep(*),INTEGER
Scan,Numpeak,Mass(*),Pot(*),Delay(*),Intn(*),Timbegin,Block,Params)
5680    OPTION BASE 1
5690    INTEGER S,P,Rej,T,True,False
5700    REAL Sumtime,Electrometer(60),Time(60),Cmean
5710    DIM K$[5],Scan$[8],M$[3],F$[4]
5720    OFF KEY
5730    False=0
5740    True= NOT False
5750    Rej=0
5760    DISP CHR$(129);
5770    ON KEY 5 LABEL Block$ CALL Weird_event
5780    ON KEY 1 LABEL "Scan" CALL Weird_event
5790    ON KEY 0 LABEL "Block" CALL Weird_event
5800    ON KEY 9 LABEL "ABORT Block" GOTO Block_abort
5810    ON KEY 2 LABEL Element$ CALL Weird_event
5820    ON KEY 3 LABEL "Field" CALL Weird_event
5830    FOR S=1 TO Scan
5840      FOR P=1 TO Numpeak
5850        K$=VAL$(Mass(P))&Elem$
5860        M$=VAL$(Mass(P))
5870        Scan$=VAL$(S)&" of "&VAL$(Scan)
5880        F$=VAL$(Pot(P))
5890        OUTPUT 12 USING "#,W";Pot(P)
5900        ON KEY 6 LABEL Scan$ CALL Weird_event
5910        ON KEY 7 LABEL M$ CALL Weird_event
5920        ON KEY 8 LABEL F$ CALL Weird_event
5930        DISP "Delay in progress, ";Rej;" readings were rejected last integration."
5940        WAIT Delay(P)
5950        Sumtime=0
5960        OUTPUT 716;"HISTORY CLEAR"
5970        FOR T=1 TO Intn(P)
5980          ENTER 716;Electrometer(T)
5990          Time(T)=TIMEDATE
6000          DISP USING "5A,""= "",5D.3D,"" mV at "",8A";K$;(Electrometer(T)-
Baseline(P,1))*1000;TIME$(Time(T))
6010          Sumtime=Sumtime+Time(T)
6020        NEXT T
6030        CALL Edit_routine(Electrometer(*),Cmean,(Intn(P)),Rej)
6040        Peak(P,S)=Cmean
6050        Timep(P,S)=Sumtime/Intn(P)
6060        IF S=1 THEN
6070          MOVE ((Timep(P,S) MOD 86400)/60-Timbegin)-(Intn(P)/2)/60,(Peak(P,S)-
Baseline(P,1))
6080          GOTO First_scan
6090        END IF
6100        MOVE ((Timep(P,S-1) MOD 86400)/60-Timbegin)+(Intn(P)/2)/60,(Peak(P,S-1)-
Baseline(P,1))
6110        LINE TYPE 4
6120        PLOT ((Timep(P,S) MOD 86400)/60-Timbegin)-(Intn(P)/2)/60,(Peak(P,S)-
Baseline(P,1)),-1
6130 First_scan:!
6140        LINE TYPE 1
6150        IDRAW Intn(P)/60,0
6160      NEXT P
6170    NEXT S
6180    SUBEXIT
6190 Block_abort: !
6200    Params=True
6210    Block=Block-1
6220  SUBEND
6230  SUB Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),INTEGER
Numpeak,Interference,Mass(*),Delay(*),Intn(*),Pot(*))
6240    OPTION BASE 1
6250    INTEGER P,S,Rej,T,B_o,First,Last,Offset,Field
6260    REAL Sumtime,Electrometer(120),Time(120),Zero(5,2),Timezero(5,2),Cmean
```

```
6270    REAL Del
6280    DIM M$[3],K$[5],F$[4]
6290    OFF KEY
6300    DISP CHR$(129);
6310    ON KEY 0 LABEL "Block" CALL Weird_event
6320    ON KEY 5 LABEL Block$ CALL Weird_event
6330    ON KEY 7 LABEL "Baselines" CALL Weird_event
6340    ON KEY 2 LABEL Element$ CALL Weird_event
6350    ON KEY 3 LABEL "Field" CALL Weird_event
6360    Del=.00368*Mass(1)+.153
6370    Offset=PROUND(ABS(Pot(1)-Pot(2))/ABS(Mass(1)-Mass(2))*Del,0)
6380    IF Mass(1)<=Mass(2) THEN
6390      First=2
6400      Last=1
6410    ELSE
6420      First=1
6430      Last=2
6440    END IF
6450    FOR P=Numpeak+Interference TO 1 STEP -1
6460      FOR S=First TO Last STEP Last-First
6470        B_o=(S-1)*Offset+(S-2)*Offset
6480        M$=VAL$(Mass(P))
6490        IF S=2 THEN
6500          K$="Above"
6510        ELSE
6520          K$="Below"
6530        END IF
6540        Field=Pot(P)+B_o
6550        F$=VAL$(Field)
6560        OUTPUT 12 USING "#,W";Field
6570        ON KEY 4 LABEL K$ CALL Weird_event
6580        ON KEY 9 LABEL M$ CALL Weird_event
6590        ON KEY 8 LABEL F$ CALL Weird_event
6600        DISP "Delay in progress, ";Rej;" readings were rejected last integration."
6610        WAIT Delay(1)
6620        Sumtime=0
6630        OUTPUT 716;"HISTORY CLEAR"
6640        FOR T=1 TO Intn(P)*2
6650          ENTER 716;Electrometer(T)
6660          Time(T)=TIMEDATE
6670          DISP USING """Baseline= "",5D.3D,"" mV at
"",8A";Electrometer(T)*1000;TIME$(Time(T))
6680          Sumtime=Sumtime+Time(T)
6690        NEXT T
6700        CALL Edit_routine(Electrometer(*),Cmean,Intn(P)*2,Rej)
6710        Zero(P,S)=Cmean
6720        Timezero(P,S)=Sumtime/(Intn(P)*2)
6730      NEXT S
6740    NEXT P
6750    IF NOT Interference THEN OUTPUT 12 USING "#,W";Pot(1)
6760    FOR P=1 TO Numpeak+Interference
6770      Baseline(P,2)=(Zero(P,1)+Zero(P,2))/2
6780      Timebase(P,2)=(Timezero(P,1)+Timezero(P,2))/2
6790    NEXT P
6800  SUBEND
6810  SUB New_zeros(Baseline(*),Timebase(*),INTEGER Numpeak,Interference,OPTIONAL
Ipeak(*),Itime(*))
6820    INTEGER P
6830    IF Interference THEN
6840      Ipeak(1)=Ipeak(2)
6850      Itime(1)=Itime(2)
6860    END IF
6870    FOR P=1 TO Numpeak+Interference
6880      Baseline(P,1)=Baseline(P,2)
6890      Timebase(P,1)=Timebase(P,2)
6900    NEXT P
```

```
6910  SUBEND
6920  SUB Parameters(Elem$,INTEGER
Check,Numpeak,Interference,Blockmax,Scan,Factor,Mass(*),Intn(*),Delay(*),OPTIONAL Disc)
6930    INTEGER N,True,False
6940    DIM Ans1$[3],Ans2$[3],Ans3$[3]
6950    CALL Clear_crt
6960    OFF KEY
6970    False=0
6980    True= NOT False
6990    DISP CHR$(128);
7000    OUTPUT 1;"                          CURRENT RUN PARAMETERS"
7010    OUTPUT 1
7020    OUTPUT 1;"Total number of blocks=";Blockmax
7030    OUTPUT 1
7040    OUTPUT 1;"Number of scans per block=";Scan
7050    OUTPUT 1
7060    OUTPUT 1;"Integration times (seconds):"
7070    OUTPUT 1;"      ";
7080    FOR N=1 TO Numpeak+Interference
7090      OUTPUT 1;"  ";Mass(N);"=";Intn(N);
7100    NEXT N
7110    OUTPUT 1 USING "/"
7120    OUTPUT 1;"Delay times (seconds):"
7130    OUTPUT 1;"      ";
7140    FOR N=1 TO Numpeak+Interference
7150      OUTPUT 1;"  Before";Mass(N);"=";Delay(N);
7160    NEXT N
7170    OUTPUT 1 USING "/"
7180    IF Factor THEN OUTPUT 1;"Complete data output."
7190    IF NOT Factor THEN OUTPUT 1;"Print means only."
7200    OUTPUT 1
7210    IF Interference AND Elem$="SR" AND SIZE(Mass,1)=5 THEN OUTPUT 1;"Correct for
87Rb."
7220    IF NOT Interference AND Elem$="SR" AND SIZE(Mass,1)=5 THEN OUTPUT 1;"No Rb
correction."
7230    IF NPAR>10 THEN OUTPUT 1;"Mass discrimination (per amu) =";Disc
7240    INPUT "Do you want to change any of these parameters?",Ans1$
7250    IF UPC$(Ans1$)="N" OR UPC$(Ans1$)="NO" THEN
7260      Check=True
7270      SUBEXIT
7280    END IF
7290    INPUT "Number of blocks (<=50)",Blockmax
7300    INPUT "Number of scans per block (<=25)",Scan
7310    DISP "Integration times in sec (";
7320    FOR N=1 TO Numpeak+Interference
7330      DISP Mass(N);",";
7340    NEXT N
7350    DISP CHR$(8);")";
7360    INPUT Intn(*)
7370    DISP "Delay times in sec (";
7380    FOR N=1 TO Numpeak+Interference
7390      DISP Mass(N);",";
7400    NEXT N
7410    DISP CHR$(8);")";
7420    INPUT Delay(*)
7430    INPUT "Complete data output (Yes or No)",Ans2$
7440    IF UPC$(Ans2$)="Y" OR UPC$(Ans2$)="YES" THEN
7450      Factor=True
7460    ELSE
7470      Factor=False
7480    END IF
7490    IF Elem$="SR" AND SIZE(Mass,1)=5 THEN
7500      INPUT "Correct for 87Rb interference",Ans3$
7510      IF UPC$(Ans3$)="N" OR UPC$(Ans3$)="NO" THEN
7520        Interference=False
7530      ELSE
```

```
7540        Interference=True
7550      END IF
7560    END IF
7570    IF NPAR>10 THEN
7580      INPUT "Mass discrimination (per amu)",Disc
7590    END IF
7600    Check=False
7610  SUBEND
7620  SUB Plot(Qpeak(*),X_gdu_max,Y_gdu_max,INTEGER Timfirst,Timlast)
7630    REAL Maxplot,L
7640    SELECT MAX(Qpeak(*))*1.3
7650    CASE <.1
7660      Maxplot=.1
7670    CASE <1
7680      Maxplot=1
7690    CASE ELSE
7700      Maxplot=10
7710    END SELECT
7720    GINIT
7730    CALL Clear_crt
7740    GRAPHICS OFF
7750    VIEWPORT .15*X_gdu_max,.99*X_gdu_max,.25*Y_gdu_max,.9*Y_gdu_max
7760    FRAME
7770    WINDOW Timfirst,Timlast,0,Maxplot
7780    AXES .1,Maxplot/50,Timfirst,Maxplot,10,5,4
7790    CLIP OFF
7800    DEG
7810    LDIR 90
7820    MOVE Timfirst-(Timlast-Timfirst)*.12,Maxplot/2
7830    LORG 4
7840    LABEL "INTENSITY  (V)"
7850    LDIR 0
7860    MOVE Timfirst+(Timlast-Timfirst)/2,Maxplot+Maxplot*.04
7870    LABEL "Time since start of run (minutes)"
7880    CSIZE 4
7890    FOR L=Timfirst TO Timlast
7900      MOVE L,Maxplot+Maxplot*.002
7910      LABEL L
7920    NEXT L
7930    LORG 8
7940    FOR L=0 TO Maxplot STEP Maxplot/10
7950      MOVE Timfirst-.05,L
7960      LABEL L
7970    NEXT L
7980    CLIP ON
7990    GRAPHICS ON
8000  SUBEND
8010  SUB Print_means(Mean(*),Sigma(*),Sigma_int,INTEGER Nblock,Nratio)
8020    INTEGER R
8030    IF Nblock=1 THEN
8040      PRINT USING "6(/)"
8050      SUBEXIT
8060    END IF
8070    PRINT
8080    PRINT "Weighted means and external 2-sigmas for ";Nblock;" blocks:"
8090    FOR R=1 TO Nratio
8100      PRINT USING "4X,3D.7D,#";Mean(R)
8110    NEXT R
8120    PRINT
8130    FOR R=1 TO Nratio
8140      PRINT USING "4X,3D.7D,#";2*Sigma(R)
8150    NEXT R
8160    PRINT USING "4X,3D.6D";2*Sigma_int
8170    PRINT USING "6(/)"
8180  SUBEND
8190  SUB Print_out1(Traw$(*),Baseline(*),Netpeak(*),Rraw(*),INTEGER
```

```
Factor,Numpeak,Interference,Block,Scan,Mass(*),Pot(*))
8200    OPTION BASE 1
8210    INTEGER N,K
8220    DIM M$(4)[7]
8230    PRINT "Block ";Block;" finished at ";TIME$(TIMEDATE)
8240    FOR N=1 TO Numpeak+Interference
8250      PRINT TAB(32+(N-1)*10);Mass(N);
8260    NEXT N
8270    PRINT
8280    PRINT "Baselines before this block:";
8290    FOR N=1 TO Numpeak+Interference
8300      PRINT USING "X,5D.3D,#";Baseline(N,1)*1000
8310    NEXT N
8320    PRINT
8330    PRINT "Baselines after this block: ";
8340    FOR N=1 TO Numpeak+Interference
8350      PRINT USING "X,5D.3D,#";Baseline(N,2)*1000
8360    NEXT N
8370    PRINT
8380    PRINT "Initial peak intensities:   ";
8390    FOR N=1 TO Numpeak+Interference
8400      PRINT USING "X,5D.3D,#";Netpeak(N,1)*1000
8410    NEXT N
8420    PRINT
8430    PRINT "Final peak intensities:     ";
8440    FOR N=1 TO Numpeak+Interference
8450      PRINT USING "X,5D.3D,#";Netpeak(N,Scan)*1000
8460    NEXT N
8470    PRINT
8480    PRINT "Magnet field values:        ";
8490    FOR N=1 TO Numpeak+Interference
8500      PRINT USING "2X,4Z,4X,#";Pot(N)
8510    NEXT N
8520    PRINT
8530    PRINT
8540    IF NOT Factor THEN SUBEXIT
8550    FOR N=2 TO Numpeak
8560      M$(N)=VAL$(Mass(N))&"/"&VAL$(Mass(1))
8570    NEXT N
8580    FOR N=2 TO Numpeak
8590      PRINT USING "4X,7A,5X,""Time"",#";M$(N)
8600    NEXT N
8610    PRINT
8620    FOR K=1 TO Scan-1
8630      FOR N=2 TO Numpeak
8640        PRINT USING "3D.6D,2X,8A,#";Rraw(N,K);Traw$(N,K)
8650      NEXT N
8660      PRINT
8670    NEXT K
8680  SUBEND
8690  SUB Print_out2(Meanraw(*),Percent_dev(*),INTEGER Numpeak,Rejected(*),Mass(*))
8700    OPTION BASE 1
8710    DIM M$(4)[7]
8720    INTEGER N
8730    FOR N=2 TO Numpeak
8740      M$(N)=VAL$(Mass(N))&"/"&VAL$(Mass(1))
8750    NEXT N
8760    FOR N=2 TO Numpeak
8770      PRINT USING "4X,7A,4X,""% Dev"",#";M$(N)
8780    NEXT N
8790    PRINT
8800    FOR N=2 TO Numpeak
8810      PRINT USING "3D.7D,3D.5D,#";Meanraw(N);Percent_dev(N)
8820    NEXT N
8830    PRINT
8840    IF MAX(Rejected(*))=0 THEN
```

```
8850      PRINT "No ratios were rejected."
8860    ELSE
8870      PRINT "Rejected:";
8880      FOR N=2 TO Numpeak
8890        PRINT TAB(15+(N-2)*20);Rejected(N);
8900      NEXT N
8910      PRINT
8920    END IF
8930  SUBEND
8940  SUB
Ratios(Traw$(*),Baseline(*),Timebase(*),Timep(*),Peak(*),Netpeak(*),Rraw(*),Avetime(*),INT
EGER Numpeak,Scan,Interference,OPTIONAL Ipeak(*),Itime(*))
8950    OPTION BASE 1
8960    INTEGER S,P,Numb
8970    REAL Zero_rate(5),Base(5,25),Peak_rate(5,25),Irate,Ibase
8980    FOR P=1 TO Numpeak+Interference
8990      Zero_rate(P)=(Baseline(P,2)-Baseline(P,1))/(Timebase(P,2)-Timebase(P,1))
9000    NEXT P
9010    FOR P=1 TO Numpeak
9020      FOR S=1 TO Scan
9030        Base(P,S)=Baseline(P,1)+Zero_rate(P)*(Timep(P,S)-Timebase(P,1))
9040        Netpeak(P,S)=Peak(P,S)-Base(P,S)
9050      NEXT S
9060    NEXT P
9070    IF Interference THEN
9080      FOR S=1 TO 2
9090        Base(5,S)=Baseline(5,1)+Zero_rate(5)*(Itime(S)-Timebase(5,1))
9100      NEXT S
9110      Netpeak(5,1)=Ipeak(1)-Base(5,1)
9120      Netpeak(5,Scan)=Ipeak(2)-Base(5,2)
9130      Irate=(Netpeak(5,Scan)-Netpeak(5,1))/(Itime(2)-Itime(1))
9140      FOR S=1 TO Scan
9150        Ibase=Netpeak(5,1)+Irate*(Timep(2,S)-Itime(1))
9160        IF Ibase>0 THEN Netpeak(2,S)=Netpeak(2,S)-Ibase*.387
9170      NEXT S
9180    END IF
9190    Numb=Scan-1
9200    FOR P=1 TO Numpeak
9210      FOR S=1 TO Numb
9220        Peak_rate(P,S)=(Netpeak(P,S)-Netpeak(P,S+1))/(Timep(P,S)-Timep(P,S+1))
9230      NEXT S
9240    NEXT P
9250    FOR P=2 TO Numpeak
9260      FOR S=1 TO Numb
9270        Avetime(P,S)=(Timep(P,S)+Timep(P,S+1)+Timep(1,S)+Timep(1,S+1))/4
9280        Rraw(P,S)=(Netpeak(P,S)+Peak_rate(P,S)*(Avetime(P,S)-
Timep(P,S)))/(Netpeak(1,S)+Peak_rate(1,S)*(Avetime(P,S)-Timep(1,S)))
9290        Traw$(P,S)=TIME$(Avetime(P,S))
9300      NEXT S
9310    NEXT P
9320  SUBEND
9330  SUB Start_up(Block$,Sample$,INTEGER Block,Blockmax,Timbegin)
9340    CALL Clear_crt
9350    DISP CHR$(128);
9360    DISP TAB(20);CHR$(131);"Printer ON LINE?"
9370    WAIT 5
9380    DISP CHR$(128);CHR$(12);
9390    PRINT TAB(20);Sample$;TAB(60);DATE$(TIMEDATE)
9400    PRINT
9410    Timbegin=INT((TIMEDATE MOD 86400)/60)
9420    Block=1
9430    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
9440  SUBEND
9450  SUB Statistics(Avetime(*),Rraw(*),Meanraw(*),Percent_dev(*),INTEGER
Numpeak,Scan,Rejected(*))
9460    OPTION BASE 1
```

```
9470    INTEGER P,K,Numb,Rej
9480    REAL Regress_time,Ratio(24),Tim(24),Cmean,Std_d,Std_dev
9490    Numb=Scan-1
9500    Regress_time=TIMEDATE
9510    FOR P=2 TO Numpeak
9520      FOR K=1 TO Numb
9530        Ratio(K)=Rraw(P,K)
9540        Tim(K)=Avetime(P,K)-Regress_time
9550      NEXT K
9560      CALL Linear_regress(Tim(*),Ratio(*),Cmean,Std_d,(Numb),Rej)
9570      Rejected(P)=Rej
9580      Meanraw(P)=Cmean
9590      Std_dev=Std_d/SQR(Numb-Rejected(P))
9600      Percent_dev(P)=Std_dev/Meanraw(P)*100
9610    NEXT P
9620  SUBEND
9630  SUB Summ_plot(Rcorr(*),Mean(*),Sigma(*),X_gdu_max,Y_gdu_max,INTEGER
Blockmax,Index1,Index2)
9640    OPTION BASE 1
9650    INTEGER I,L,True
9660    REAL Minplot,Maxplot,Plot_mean,Plot_sdev
9670    ALLOCATE Plot_corr(Blockmax),Plot_sigma(Blockmax)
9680    True=1
9690    FOR I=1 TO Blockmax
9700      Plot_corr(I)=Rcorr(Index1,I)
9710      Plot_sigma(I)=Rcorr(Index2,I)*Plot_corr(I)*2/100
9720    NEXT I
9730    Plot_mean=Mean(Index1)
9740    Plot_sdev=Sigma(Index1)
9750    GINIT
9760    CALL Clear_crt
9770    GRAPHICS OFF
9780    VIEWPORT .01*X_gdu_max,.99*X_gdu_max,.3*Y_gdu_max,.99*Y_gdu_max
9790    FRAME
9800    Minplot=MIN(Plot_corr(*))-MAX(Plot_sigma(*))
9810    Maxplot=MAX(Plot_corr(*))+MAX(Plot_sigma(*))
9820    WINDOW 0,Blockmax+1,Minplot,Maxplot
9830    AXES 1,0,0,Minplot,1,1,4
9840    CLIP OFF
9850    CSIZE 4
9860    LORG 6
9870    FOR L=5 TO Blockmax+1 STEP 5
9880      MOVE L,Minplot
9890      LABEL L
9900    NEXT L
9910    CLIP ON
9920    FOR I=1 TO Blockmax
9930      MOVE I-.3,Plot_corr(I)-Plot_sigma(I)
9940      RECTANGLE .6,2*Plot_sigma(I),FILL,EDGE
9950    NEXT I
9960    FOR I=-1 TO 1
9970      MOVE 0,Plot_mean+I*2*Plot_sdev
9980      LINE TYPE ABS(I)*3+1
9990      IDRAW Blockmax+1,0
10000   NEXT I
10010   GRAPHICS ON
10020   ON KEY 4 LABEL "Dump Graphics" GOTO Dump
10030   ON KEY 0 LABEL "Continue" GOTO Done
10040   WHILE True
10050   END WHILE
10060 Dump:!
10070   CALL Dump
10080 Done:!
10090 SUBEND
10100 SUB Weird_event
10110   BEEP
```

```
10120 SUBEND
```

KBDM

```
10    REM KBDM                  Brian D. Marshall 7/88
20    OPTION BASE 1
30    PRINTER IS PRT
40    CONTROL 1,4;0             !Turns off DISPLAY FUNCTIONS
50    LOADSUB ALL FROM "Subfile"
60    DIM
A$[4],Block$[8],Elem$[2],Element$[10],File$[6],N$[6],Sample$[35],Traw$(2,24)[8],Fil$[1]
70    INTEGER Block,Blockmax,Check,Delay(2),Edit1,Factor,False,Halt
80    INTEGER Index1,Index2,Interference,Intn(2),Mass(2),Numpeak,Params,Pot(2)
90    INTEGER Prog,Rejected(2),Scan,Timbegin,Timfirst,Timlast,True
100   REAL Avetime(2,24),Baseline(2,2),Disc,Mean(2)
110   REAL Meanraw(2),Netpeak(2,24),Peak(2,24),Percent_dev(2)
120   REAL Qpeak(2),Rcorr(7,50),Rraw(2,24),Rspike,Rstore(4),Scantime
130   REAL Sigma(2),Sigma_int,Tcorr(2),Timebase(2,2),Timep(2,24)
140   REAL X_gdu_max,Y_gdu_max
150   CALL Clear_crt
160   Y_gdu_max=100*MAX(1,1/RATIO)
170   X_gdu_max=100*MAX(1,RATIO)
180   False=0
190   True= NOT False
200   Element$="Potassium"
210   Elem$="K"
220   CALL Identify(Elem$,Sample$,X_gdu_max,Y_gdu_max)
230   INPUT "Triple filament as perchlorate (T) or single filament as chloride (S)?",Fil$
240   ON KEY 0 LABEL "Sample" GOTO Unspiked
250   ON KEY 2 LABEL "Spiked sample" GOTO Spiked
260   ON KEY 4 LABEL "Spike" GOTO Spike
270   WHILE True
280   END WHILE
290 Unspiked:!
300   Prog=1
310   GOTO Initialize
320 Spiked:!
330   Prog=3
340   GOTO Initialize
350 Spike:!
360   Prog=2
370 Initialize:!
380   OFF KEY
390   GRAPHICS OFF
400   IF Prog=3 THEN GOSUB Spikeratios
410   GOSUB Defaults
420   ON KEY 2 LABEL "START" GOTO Main
430   ON KEY 9 LABEL "Escape" GOTO Initialize
440   ON KEY 5 LABEL "ABORT RUN!" GOTO Escape
450   WHILE True
460   END WHILE
470 Massdisc:!
480   Rcorr(1,Block)=1                                    !39S/41T
490   Rcorr(2,Block)=Meanraw(2)*(1+2*Disc)                !41/39
500   Rcorr(3,Block)=Percent_dev(2)
510   Rcorr(4,Block)=Block
520   Rcorr(7,Block)=1/Rcorr(3,Block)^2
530   RETURN
540 Spikesub:!
550   Tcorr(2)=Meanraw(2)*(1+2*Disc)
560   Rcorr(1,Block)=(1-Tcorr(2)*Rspike)/(Tcorr(2)-.072168)
570   Rcorr(2,Block)=(Tcorr(2)*Rspike-1)/Rcorr(1,Block)+Tcorr(2)
580   Rcorr(3,Block)=Percent_dev(2)
590   Rcorr(4,Block)=Block
600   Rcorr(7,Block)=1/Rcorr(3,Block)^2
610   RETURN
620 Printout3: !
```

```
630 Format3: IMAGE 2(4X,3D.7D)
640   IF Prog=3 THEN
650     PRINT
660     PRINT "Spike ratio: ";DROUND(Rspike,6)
670   END IF
680   PRINT
690   PRINT "Mass discrimination corrected (=";Disc;"per a.m.u.)."
700   PRINT
710   PRINT USING "2(9X,7A)";"39S/41T";" 41/39 "
720   PRINT USING Format3;Rcorr(1,Block);Rcorr(2,Block)
730   CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Block),2)
740   RETURN
750 Defaults: !
760   Blockmax=10
770   Intn(1)=5
780   Intn(2)=5
790   Delay(1)=5
800   Delay(2)=5
810   Pot(1)=1250
820   Pot(2)=2170
830   Mass(1)=39
840   Mass(2)=41
850   Scan=11
860   Factor=False
870   Interference=False
880   Numpeak=2
890   IF UPC$(Fil$)="T" THEN
900     Disc=.003
910   ELSE
920     Disc=.012
930   END IF
940 Showdefaults: !
950   CALL Field_values(X_gdu_max,Y_gdu_max,Numpeak,Interference,Mass(*),Pot(*))
960   CALL
Parameters(Elem$,Check,Numpeak,Interference,Blockmax,Scan,Factor,Mass(*),Intn(*),Delay(*),
Disc)
970   IF Check THEN RETURN
980   IF NOT Check THEN GOTO Showdefaults
990 Spikeratios: !
1000  Rspike=.02015    !39/41
1010  RETURN
1020  REM Key_handlers
1030 Pause:!
1040  OFF KEY
1050  Halt=True
1060  RETURN
1070 Last_block:!
1080  OFF KEY
1090  Blockmax=Block
1100  RETURN
1110 Escape:!
1120  OFF KEY
1130  CALL Clear_crt
1140  DISP CHR$(131);"PROGRAM ABORTED"
1150  WAIT 5
1160  GOTO Lastline
1170 New_params:!
1180  OFF KEY
1190  Params=True
1200  RETURN
1210 Summary:!
1220   Index1=(Prog<>3)+1
1230   Index2=3
1240  CALL Final_print(Sample$,Rcorr(*),4,Blockmax)
1250  CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Blockmax),2)
1260  CALL Summ_plot(Rcorr(*),Mean(*),Sigma(*),X_gdu_max,Y_gdu_max,Blockmax,Index1,Index2)
```

```
1270  DISP CHR$(128)
1280  LOOP
1290    Edit1=0
1300    INPUT "Number of blocks to be edited",Edit1
1310  EXIT IF Edit1=0
1320    CALL Edit(Rcorr(*),Mean(*),Sigma(*),Sigma_int,(Edit1),2,Blockmax)
1330  END LOOP
1340  RETURN
1350 Main:!
1360  OFF KEY
1370  CALL Start_up(Block$,Sample$,Block,Blockmax,Timbegin)
1380  CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
1390  CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
1400  WHILE Block<=Blockmax
1410    OFF KEY
1420    Halt=False
1430    Params=False
1440    CALL Center_peaks(Qpeak(*),Numpeak,Interference,Mass(*),Pot(*))
1450    Timfirst=INT((TIMEDATE MOD 86400)/60)-Timbegin
1460    Scantime=(SUM(Delay)+SUM(Intn))/60
1470    Timlast=Timfirst+PROUND(Scantime*(Scan+4),0)
1480    CALL Plot(Qpeak(*),X_gdu_max,Y_gdu_max,Timfirst,Timlast)
1490    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
1500    CALL
Measure_peaks(Element$,Elem$,Block$,Baseline(*),Peak(*),Timep(*),Scan,Numpeak,Mass(*),Pot(
*),Delay(*),Intn(*),Timbegin,Block,Params)
1510    IF Params THEN GOTO Hold
1520    CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
1530    ON KEY 9 LABEL "ABORT RUN!" GOTO Escape
1540    ON KEY 5 LABEL "FINISH" GOSUB Last_block
1550    ON KEY 2 LABEL "PAUSE" GOSUB Pause
1560    ON KEY 7 LABEL "Change Params" GOSUB New_params
1570    BEEP 880,2
1580    DISP CHR$(128);"Calculating results"
1590    CALL
Ratios(Traw$(*),Baseline(*),Timebase(*),Timep(*),Peak(*),Netpeak(*),Rraw(*),Avetime(*),Num
peak,Scan,Interference)
1600    CALL
Print_out1(Traw$(*),Baseline(*),Netpeak(*),Rraw(*),Factor,Numpeak,Interference,Block,Scan,
Mass(*),Pot(*))
1610    CALL
Statistics(Avetime(*),Rraw(*),Meanraw(*),Percent_dev(*),Numpeak,Scan,Rejected(*))
1620    CALL Print_out2(Meanraw(*),Percent_dev(*),Numpeak,Rejected(*),Mass(*))
1630    IF Prog=3 THEN
1640      GOSUB Spikesub
1650    ELSE
1660      GOSUB Massdisc
1670    END IF
1680    CALL Grandstats(Rcorr(*),Mean(*),Sigma(*),Sigma_int,2,(Block))
1690    GOSUB Printout3
1700 Hold:!
1710    IF Halt THEN
1720      DISP CHR$(128);"Press CONTINUE to resume data acquisition."
1730      PAUSE
1740    END IF
1750    IF Params THEN
1760      CALL Clear_crt
1770      GOSUB Showdefaults
1780      DISP CHR$(128);"Press CONTINUE to resume data acquisition."
1790      PAUSE
1800    END IF
1810    Block=Block+1
```

```
1820     Block$=VAL$(Block)&" of "&VAL$(Blockmax)
1830     IF Params AND Block<=Blockmax THEN CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
1840     CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
1850   END WHILE
1860   OFF KEY
1870   GOSUB Summary
1880 Lastline:!
1890   CALL Clear_crt
1900   PRINT USING "@"
1910   LOAD "CONTROL"
1920   END
```

CABDM

```
10     REM CABDM                  Brian D. Marshall 7/88
20     OPTION BASE 1
30     PRINTER IS PRT
40     CONTROL 1,4;0              !Turns off DISPLAY FUNCTIONS.
50     LOADSUB ALL FROM "Subfile"
60     DIM
A$[4],Block$[8],Elem$[2],Element$[10],File$[6],N$[6],Sample$[35],Traw$(4,24)[8],Today$[11]
,Now$[8],Mon$[3]
70     INTEGER B,Block,Blockmax,Check,Delay(4),Edit1,Factor,False,Halt,I
80     INTEGER Index1,Index2,Interference,Intn(4),Iter,Mass(4),Numpeak,Params
90     INTEGER Pot(4),Prog,Rejected(4),Scan,Sp,Timbegin,Timfirst,Timlast,True
100    REAL Alpha,Amu(4),Avetime(4,24),Baseline(4,2),Beta,Disc,Mean(4)
110    REAL Meancorr(4),Meanraw(4),Netpeak(4,24),P_exp,Peak(4,24),Percent_dev(4)
120    REAL Qpeak(4),Rcorr(7,50),Rraw(4,24),Rspike(3),Rstore(6),Scantime
130    REAL Sigma(4),Sigma_int,Sum_p,Tcorr(3),Timebase(4,2),Timep(4,24)
140    REAL X_gdu_max,Y_gdu_max
150    CALL Clear_crt
160    Y_gdu_max=100*MAX(1,1/RATIO)
170    X_gdu_max=100*MAX(1,RATIO)
180    False=0
190    True= NOT False
200    Elem$="CA"
210    Element$="Calcium"
220    OFF KEY
230    CALL Identify(Elem$,Sample$,X_gdu_max,Y_gdu_max)
250    ON KEY 0 LABEL "Sample" GOTO Unspiked
260    ON KEY 2 LABEL "Spiked sample" GOTO Spiked
270    ON KEY 4 LABEL "Spike" GOTO Spike
280    WHILE True
290    END WHILE
300 Unspiked:!
310    Prog=1
320    GOTO Initialize
330 Spiked:!
340    Prog=3
350    GOTO Initialize
360 Spike:!
370    Prog=2
380 Initialize:!
390    OFF KEY
400    GRAPHICS OFF
410    IF Prog=3 THEN GOSUB Spikeratios
420    GOSUB Defaults
430    ON KEY 2 LABEL "START" GOTO Main
440    ON KEY 9 LABEL "Escape" GOTO Initialize
450    ON KEY 5 LABEL "ABORT RUN!" GOTO Escape
460    WHILE True
470    END WHILE
480 Massdisc:!
490    Sum_p=LOG(.31221/(Meanraw(2)/Meanraw(4)))/LOG(Amu(2)/Amu(4))
```

```
500   IF Prog=2 THEN Sum_p=0                    !No correction for spike run
510   Rcorr(1,Block)=1                                                !42S/43T
520   Rcorr(2,Block)=1/(Meanraw(2))*(Amu(1)/Amu(2))^Sum_p             !40/42
530   Rcorr(3,Block)=1/(Meanraw(4))*(Amu(1)/Amu(4))^Sum_p             !40/44
540   Rcorr(4,Block)=(Meanraw(3)/Meanraw(2))*(Amu(3)/Amu(2))^Sum_p     !43/42
550   Rcorr(5,Block)=SQR(2*Percent_dev(2)^2+Percent_dev(4)^2)
560   Rcorr(6,Block)=Block
570   Rcorr(7,Block)=1/Rcorr(5,Block)^2
580   RETURN
590 Spikesub:!
600   FOR I=2 TO 4
610     Meancorr(I)=Meanraw(I)
620   NEXT I
630   Sum_p=0
640   Iter=0
650   REPEAT
660     Alpha=(Meancorr(3)/Meancorr(2)-.207745)/(1-Meancorr(3)/Meancorr(2)*Rspike(1))
670     Beta=1/(Meancorr(4)/Meancorr(2)+Alpha*Rspike(1)*(Meancorr(4)/Meancorr(2)-
Rspike(3)/Rspike(2)))
680     P_exp=LOG(.31221/Beta)/LOG(Amu(2)/Amu(4))
690     Tcorr(1)=1/Alpha
700     Tcorr(2)=1/(Meancorr(2))*(Amu(1)/Amu(2))^P_exp
710     Tcorr(3)=1/(Meancorr(4))*(Amu(1)/Amu(4))^P_exp
720     Sum_p=Sum_p+P_exp
730     Iter=Iter+1
740     Meancorr(2)=1/Tcorr(2)
750     Meancorr(3)=Meancorr(3)*(Amu(3)/Amu(1))^P_exp
760     Meancorr(4)=1/Tcorr(3)
770   UNTIL ABS(P_exp)<1.E-4
780   DISP "Number of iterations was ";Iter
790   Rcorr(1,Block)=Tcorr(1)
800   Rcorr(2,Block)=Tcorr(2)+Alpha*Rspike(1)*(Tcorr(2)-1/Rspike(2))
810   Rcorr(3,Block)=Rcorr(2,Block)*.31221
820   Rcorr(4,Block)=.207745
830   Rcorr(5,Block)=SQR(2*Percent_dev(2)^2+Percent_dev(4)^2)
840   Rcorr(6,Block)=Block
850   Rcorr(7,Block)=1/Rcorr(5,Block)^2
860   RETURN
870 Printout3: !
880   Disc=((Amu(2)/Amu(4))^Sum_p-1)/2
890 Format3: IMAGE 5(4X,3D.7D)
900   IF Prog=3 THEN
910     PRINT
920     PRINT "Spike ratios:
";DROUND(Rspike(1),6);",";DROUND(Rspike(2),6);",";DROUND(Rspike(3),6)
930     PRINT "Number of iterations was ";Iter
940   END IF
950   PRINT
960   PRINT "Mass discrimination corrected (p=";DROUND(Sum_p,6);",
alpha=";DROUND(Disc,6);")."
970   PRINT
980   PRINT USING "5(8X,7A)";"42S/43T";" 40/42 ";" 40/44 ";" 43/42 ";"% Error"
990   PRINT USING
Format3;Rcorr(1,Block);Rcorr(2,Block);Rcorr(3,Block);Rcorr(4,Block);Rcorr(5,Block)
1000  CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Block),4)
1010  RETURN
1020 Defaults:!
1030  Blockmax=30
1040  Intn(1)=5
1050  Intn(2)=9
1060  Intn(3)=7
1070  Intn(4)=7
1080  Delay(1)=5
1090  Delay(2)=6
1100  Delay(3)=4
1110  Delay(4)=4
```

```
1120  Pot(1)=1700
1130  Pot(2)=2600
1140  Pot(3)=3040
1150  Pot(4)=3475
1160  Mass(1)=40
1170  Mass(2)=42
1180  Mass(3)=43
1190  Mass(4)=44
1200  Amu(1)=39.962591
1210  Amu(2)=41.958622
1220  Amu(3)=42.958770
1230  Amu(4)=43.955485
1240  Scan=11
1250  Factor=False
1260  Interference=False
1270  Numpeak=4
1280 Showdefaults:!
1290  CALL Field_values(X_gdu_max,Y_gdu_max,Numpeak,Interference,Mass(*),Pot(*))
1300  CALL
Parameters(Elem$,Check,Numpeak,Interference,Blockmax,Scan,Factor,Mass(*),Intn(*),Delay(*))
1310  IF Check THEN RETURN
1320  IF NOT Check THEN GOTO Showdefaults
1330 Spikeratios:!
1340  CALL Clear_crt
1350  OUTPUT 1
1360  OUTPUT 1
1370  OUTPUT 1;"1=Ca-Sr Mixed Spike 1 (salt)"
1380  OUTPUT 1;"2=Ca-Sr Mixed Spike 2 (mica)"
1390  OUTPUT 1;"3=Ca43-2(X100)"
1400  OUTPUT 1;"4=Ca43-2"
1410  INPUT "Which spike (type a number then ENTER)?",Sp
1420  IF Sp=1 THEN
1430    Rspike(1)=.013606              !42/43
1440    Rspike(2)=.055219              !42/40
1450    Rspike(3)=.34887               !44/40
1460  END IF
1470  IF Sp=2 THEN
1480    Rspike(1)=.013016
1490    Rspike(2)=.08087
1500    Rspike(3)=.52441
1510  END IF
1520  IF Sp=3 THEN
1530    Rspike(1)=.013135
1540    Rspike(2)=.080413
1550    Rspike(3)=.51197
1560  END IF
1570  IF Sp=4 THEN
1580    Rspike(1)=.013063
1590    Rspike(2)=.082955
1600    Rspike(3)=.53219
1610  END IF
1620  IF Sp<>1 AND Sp<>2 AND Sp<>3 AND Sp<>4 THEN Spikeratios
1630  RETURN
1640  REM Key_handlers
1650 Pause:!
1660  OFF KEY
1670  Halt=True
1680  RETURN
1690 Last_block:!
1700  OFF KEY
1710  Blockmax=Block
1720  RETURN
1730 Escape:!
1740  OFF KEY
1750  CALL Clear_crt
1760  DISP CHR$(131);"PROGRAM ABORTED"
```

```
1770  WAIT 5
1780  GOTO Lastline
1790 New_params: !
1800  OFF KEY
1810  Params=True
1820  RETURN
1830 Summary:!
1840  Index1=2
1850  Index2=5
1860  CALL Final_print(Sample$,Rcorr(*),6,Blockmax)
1870  CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Blockmax),4)
1880  CALL Summ_plot(Rcorr(*),Mean(*),Sigma(*),X_gdu_max,Y_gdu_max,Blockmax,Index1,Index2)
1890  DISP CHR$(128)
1900  LOOP
1910    Edit1=0
1920    INPUT "Number of blocks to be edited",Edit1
1930  EXIT IF Edit1=0
1940    CALL Edit(Rcorr(*),Mean(*),Sigma(*),Sigma_int,(Edit1),4,Blockmax)
1950  END LOOP
1960  RETURN
1970 Main:!
1980  OFF KEY
1990  MASS STORAGE IS ":,700,0"
2000  READ LABEL N$ FROM ":,700,0"
2010  A$="DATA"
2020  IF POS(UPC$(N$),A$)=0 THEN
2030    DISP "Insert a data disk in the left drive and press Continue."
2040    PAUSE
2050  END IF
2060  OFF KEY
2061  Today$=DATE$(TIMEDATE)
2062  Now$=TIME$(TIMEDATE)
2063  SELECT Today$[4,6]
2064    CASE "Mar","Jul"
2065      Mon$=Today$[4,4]&Today$[6,6]
2066    CASE ELSE
2067      Mon$=Today$[4,5]
2068  END SELECT
2069  File$=Mon$&Today$[10]&Today$[1,2]&Now$[1,2]
2070  CREATE BDAT File$,52,48
2080  ASSIGN @Data TO File$              !I/O path for disc file
2090  OUTPUT @Data,1;DATE$(TIMEDATE)
2100  OUTPUT @Data,2;Sample$
2110  CALL Start_up(Block$,Sample$,Block,Blockmax,Timbegin)
2120  CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2130  CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
2140  WHILE Block<=Blockmax                          !main program loop
2150    OFF KEY
2160    Halt=False
2170    Params=False
2180    CALL Center_peaks(Qpeak(*),Numpeak,Interference,Mass(*),Pot(*))
2190    Timfirst=INT((TIMEDATE MOD 86400)/60)-Timbegin
2200    Scantime=(SUM(Delay)+SUM(Intn))/60
2210    Timlast=Timfirst+PROUND(Scantime*(Scan+4),0)
2220    CALL Plot(Qpeak(*),X_gdu_max,Y_gdu_max,Timfirst,Timlast)
2230    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
2240    CALL
Measure_peaks(Element$,Elem$,Block$,Baseline(*),Peak(*),Timep(*),Scan,Numpeak,Mass(*),Pot(
*),Delay(*),Intn(*),Timbegin,Block,Params)
2250    IF Params THEN GOTO Hold
2260    CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2270    ON KEY 9 LABEL "ABORT RUN!" GOTO Escape
```

```
2280     ON KEY 5 LABEL "FINISH" GOSUB Last_block
2290     ON KEY 2 LABEL "PAUSE" GOSUB Pause
2300     ON KEY 7 LABEL "Change Params" GOSUB New_params
2310     BEEP 880,2
2320     DISP CHR$(128);"Calculating results"
2330     CALL
Ratios(Traw$(*),Baseline(*),Timebase(*),Timep(*),Peak(*),Netpeak(*),Rraw(*),Avetime(*),Num
peak,Scan,Interference)
2340     CALL
Print_out1(Traw$(*),Baseline(*),Netpeak(*),Rraw(*),Factor,Numpeak,Interference,Block,Scan,
Mass(*),Pot(*))
2350     CALL
Statistics(Avetime(*),Rraw(*),Meanraw(*),Percent_dev(*),Numpeak,Scan,Rejected(*))
2360     CALL Print_out2(Meanraw(*),Percent_dev(*),Numpeak,Rejected(*),Mass(*))
2370     IF Prog=3 THEN
2380       GOSUB Spikesub
2390     ELSE
2400       GOSUB Massdisc
2410     END IF
2420     FOR B=1 TO 6
2430       Rstore(B)=Rcorr(B,Block)
2440     NEXT B
2450     OUTPUT @Data,Block+2;Rstore(*)
2460     CALL Grandstats(Rcorr(*),Mean(*),Sigma(*),Sigma_int,4,(Block))
2470     GOSUB Printout3
2480 Hold:!
2490     IF Halt THEN
2500       DISP CHR$(128);"Press CONTINUE to resume data acquisition."
2510       PAUSE
2520     END IF
2530     IF Params THEN
2540       CALL Clear_crt
2550       GOSUB Showdefaults
2560       DISP CHR$(128);"Press CONTINUE to resume data acquisition."
2570       PAUSE
2580     END IF
2590     Block=Block+1
2600     Block$=VAL$(Block)&" of "&VAL$(Blockmax)
2610     IF Params AND Block<=Blockmax THEN CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2620     CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
2630   END WHILE
2640   OFF KEY
2650   GOSUB Summary
2660 Lastline: MASS STORAGE IS ":,700,1"
2670   CALL Clear_crt
2680   PRINT USING "@"
2690   LOAD "CONTROL"
2700   END
```

RBBDM

```
10     REM RBBDM                  Brian D. Marshall  7/88
20     OPTION BASE 1
30     PRINTER IS PRT
40     CONTROL 1,4;0             !Turns off DISPLAY FUNCTIONS
50     LOADSUB ALL FROM "Subfile"
60     DIM
A$[4],Block$[8],Elem$[2],Element$[10],File$[6],N$[6],Sample$[35],Traw$(2,24)[8],Fil$[1]
70     INTEGER Block,Blockmax,Check,Delay(2),Edit1,Factor,False,Halt
80     INTEGER Index1,Index2,Interference,Intn(2),Mass(2),Numpeak,Params,Pot(2)
90     INTEGER Prog,Rejected(2),Scan,Timbegin,Timfirst,Timlast,True
100    REAL Avetime(2,24),Baseline(2,2),Disc,Mean(2)
110    REAL Meanraw(2),Netpeak(2,24),Peak(2,24),Percent_dev(2)
120    REAL Qpeak(2),Rcorr(7,50),Rraw(2,24),Rspike,Rstore(4),Scantime
```

```
130   REAL Sigma(2),Sigma_int,Tcorr(2),Timebase(2,2),Timep(2,24)
140   REAL X_gdu_max,Y_gdu_max
150   CALL Clear_crt
160   False=0
170   True= NOT False
180   Element$="Rubidium"
190   Elem$="RB"
200   Y_gdu_max=100*MAX(1,1/RATIO)
210   X_gdu_max=100*MAX(1,RATIO)
220   CALL Identify(Elem$,Sample$,X_gdu_max,Y_gdu_max)
230   INPUT "Triple filament as perchlorate (T) or single filament as chloride (S)?",Fil$
240   ON KEY 0 LABEL "Sample" GOTO Unspiked
250   ON KEY 2 LABEL "Spiked sample" GOTO Spiked
260   ON KEY 4 LABEL "Spike" GOTO Spike
270   WHILE True
280   END WHILE
290 Unspiked:!
300   Prog=1
310   GOTO Initialize
320 Spiked:!
330   Prog=3
340   GOTO Initialize
350 Spike:!
360   Prog=2
370 Initialize:!
380   OFF KEY
390   GRAPHICS OFF
400   IF Prog=3 THEN GOSUB Spikeratios
410   GOSUB Defaults
420   ON KEY 2 LABEL "START" GOTO Main
430   ON KEY 9 LABEL "Escape" GOTO Initialize
440   ON KEY 5 LABEL "ABORT RUN!" GOTO Escape
450   WHILE True
460   END WHILE
470 Massdisc:!
480   Rcorr(1,Block)=1                                        !85S/87T
490   Rcorr(2,Block)=Meanraw(2)*(1+2*Disc)                    !87/85
500   Rcorr(3,Block)=Percent_dev(2)
510   Rcorr(4,Block)=Block
520   Rcorr(7,Block)=1/Rcorr(3,Block)^2
530   RETURN
540 Spikesub:!
550   Tcorr(2)=Meanraw(2)*(1+2*Disc)
560   Rcorr(1,Block)=(1-Tcorr(2)*Rspike)/(Tcorr(2)-.38571)
570   Rcorr(2,Block)=(Tcorr(2)*Rspike-1)/Rcorr(1,Block)+Tcorr(2)
580   Rcorr(3,Block)=Percent_dev(2)
590   Rcorr(4,Block)=Block
600   Rcorr(7,Block)=1/Rcorr(3,Block)^2
610   RETURN
620 Printout3: !
630 Format3: IMAGE 2(4X,3D.7D)
640   IF Prog=3 THEN
650     PRINT
660     PRINT "Spike ratio: ";DROUND(Rspike,6)
670   END IF
680   PRINT
690   PRINT "Mass discrimination corrected (=";Disc;"per a.m.u.)."
700   PRINT
710   PRINT USING "2(9X,7A)";"85S/87T";" 87/85 "
720   PRINT USING Format3;Rcorr(1,Block);Rcorr(2,Block)
730   CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Block),2)
740   RETURN
750 Defaults: !
760   Blockmax=10
770   Intn(1)=5
780   Intn(2)=5
```

```
790    Delay(1)=5
800    Delay(2)=5
810    Pot(1)=2040
820    Pot(2)=2735
830    Mass(1)=85
840    Mass(2)=87
850    Scan=11
860    Factor=False
870    Interference=False
880    Numpeak=2
890    IF UPC$(Fil$)="T" THEN
900      Disc=.0017
910    ELSE
920      Disc=.002
930    END IF
940 Showdefaults: !
950    CALL Field_values(X_gdu_max,Y_gdu_max,Numpeak,Interference,Mass(*),Pot(*))
960    CALL
Parameters(Elem$,Check,Numpeak,Interference,Blockmax,Scan,Factor,Mass(*),Intn(*),Delay(*),
Disc)
970    IF Check THEN RETURN
980    IF NOT Check THEN GOTO Showdefaults
990 Spikeratios: !
1000   Rspike=.00833     !85/87
1010   RETURN
1020   REM Key_handlers
1030 Pause:!
1040   OFF KEY
1050   Halt=True
1060   RETURN
1070 Last_block:!
1080   OFF KEY
1090   Blockmax=Block
1100   RETURN
1110 Escape:!
1120   OFF KEY
1130   CALL Clear_crt
1140   DISP CHR$(131);"PROGRAM ABORTED"
1150   WAIT 5
1160   GOTO Lastline
1170 New_params:!
1180   OFF KEY
1190   Params=True
1200   RETURN
1210 Summary:!
1220   Index1=(Prog<>3)+1
1230   Index2=3
1240   CALL Final_print(Sample$,Rcorr(*),4,(Blockmax))
1250   CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Blockmax),2)
1260   CALL Summ_plot(Rcorr(*),Mean(*),Sigma(*),X_gdu_max,Y_gdu_max,Blockmax,Index1,Index2)
1270   DISP CHR$(128)
1280   LOOP
1290     Edit1=0
1300     INPUT "Number of blocks to be edited",Edit1
1310   EXIT IF Edit1=0
1320     CALL Edit(Rcorr(*),Mean(*),Sigma(*),Sigma_int,(Edit1),2,Blockmax)
1330   END LOOP
1340   RETURN
1350 Main:!
1360   OFF KEY
1370   CALL Start_up(Block$,Sample$,Block,Blockmax,Timbegin)
1380   CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
1390   CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
1400   WHILE Block<=Blockmax
```

```
1410    OFF KEY
1420    Halt=False
1430    Params=False
1440    CALL Center_peaks(Qpeak(*),Numpeak,Interference,Mass(*),Pot(*))
1450    Timfirst=INT((TIMEDATE MOD 86400)/60)-Timbegin
1460    Scantime=(SUM(Delay)+SUM(Intn))/60
1470    Timlast=Timfirst+PROUND(Scantime*(Scan+4),0)
1480    CALL Plot(Qpeak(*),X_gdu_max,Y_gdu_max,Timfirst,Timlast)
1490    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
1500    CALL
Measure_peaks(Element$,Elem$,Block$,Baseline(*),Peak(*),Timep(*),Scan,Numpeak,Mass(*),Pot(
*),Delay(*),Intn(*),Timbegin,Block,Params)
1510    IF Params THEN GOTO Hold
1520    CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
1530    ON KEY 9 LABEL "ABORT RUN!" GOTO Escape
1540    ON KEY 5 LABEL "FINISH" GOSUB Last_block
1550    ON KEY 2 LABEL "PAUSE" GOSUB Pause
1560    ON KEY 7 LABEL "Change Params" GOSUB New_params
1570    BEEP 880,2
1580    DISP CHR$(128);"Calculating results"
1590    CALL
Ratios(Traw$(*),Baseline(*),Timebase(*),Timep(*),Peak(*),Netpeak(*),Rraw(*),Avetime(*),Num
peak,Scan,Interference)
1600    CALL
Print_out1(Traw$(*),Baseline(*),Netpeak(*),Rraw(*),Factor,Numpeak,Interference,Block,Scan,
Mass(*),Pot(*))
1610    CALL
Statistics(Avetime(*),Rraw(*),Meanraw(*),Percent_dev(*),Numpeak,Scan,Rejected(*))
1620    CALL Print_out2(Meanraw(*),Percent_dev(*),Numpeak,Rejected(*),Mass(*))
1630    IF Prog=3 THEN
1640      GOSUB Spikesub
1650    ELSE
1660      GOSUB Massdisc
1670    END IF
1680    CALL Grandstats(Rcorr(*),Mean(*),Sigma(*),Sigma_int,2,(Block))
1690    GOSUB Printout3
1700 Hold:!
1710    IF Halt THEN
1720      DISP CHR$(128);"Press CONTINUE to resume data acquisition."
1730      PAUSE
1740    END IF
1750    IF Params THEN
1760      CALL Clear_crt
1770      GOSUB Showdefaults
1780      DISP CHR$(128);"Press CONTINUE to resume data acquisition."
1790      PAUSE
1800    END IF
1810    Block=Block+1
1820    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
1830    IF Params AND Block<=Blockmax THEN CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
1840    CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
1850  END WHILE
1860  OFF KEY
1870  GOSUB Summary
1880 Lastline:!
1890  CALL Clear_crt
1900  PRINT USING "@"
1910  LOAD "CONTROL"
1920  END
```

SRBDM

```
10    REM SRBDM                Brian D. Marshall  7/88
```

```
20    OPTION BASE 1
30    PRINTER IS PRT
40    CONTROL 1,4;0              !Turns off DISPLAY FUNCTIONS.
50    LOADSUB ALL FROM "Subfile"
60    DIM
A$[4],Block$[8],Elem$[2],Element$[10],File$[6],N$[6],Sample$[35],Traw$(4,24)[8],Today$[11]
,Now$[8],Mon$[3]
70    INTEGER B,Block,Blockmax,Check,Delay(5),Edit1,Factor,False,Halt,I
80    INTEGER Index1,Index2,Interference,Intn(5),Iter,Mass(5),Numpeak,Params
90    INTEGER Pot(5),Prog,Rejected(4),Scan,Sp,Timbegin,Timfirst,Timlast,True
100   REAL Alpha,Amu(4),Avetime(5,24),Baseline(5,2),Beta,Disc,Ipeak(2),Itime(2)
110   REAL Mean(4),Meancorr(4),Meanraw(4),Netpeak(5,24),P_exp,Peak(5,24)
120   REAL Percent_dev(4),Qpeak(5),Rcorr(7,50),Rraw(5,24),Rspike(3),Rstore(6)
130   REAL Scantime,Sigma(4),Sigma_int,Sum_p,Tcorr(3),Timebase(5,2),Timep(5,24)
140   REAL X_gdu_max,Y_gdu_max
150   CALL Clear_crt
160   OFF KEY
170   DISP CHR$(128)
180   Y_gdu_max=100*MAX(1,1/RATIO)
190   X_gdu_max=100*MAX(1,RATIO)
200   False=0
210   True= NOT False
220   Elem$="SR"
230   Element$="Strontium"
240   CALL Identify(Elem$,Sample$,X_gdu_max,Y_gdu_max)
260   ON KEY 0 LABEL "Sample" GOTO Unspiked
270   ON KEY 2 LABEL "Spiked sample" GOTO Spiked
280   ON KEY 4 LABEL "Spike" GOTO Spike
290   WHILE True
300   END WHILE
310 Unspiked:!
320   Prog=1
330   GOTO Initialize
340 Spiked:!
350   Prog=3
360   GOTO Initialize
370 Spike:!
380   Prog=2
390 Initialize:!
400   OFF KEY
410   GRAPHICS OFF
420   IF Prog=3 THEN GOSUB Spikeratios
430   GOSUB Defaults
440   ON KEY 2 LABEL "START" GOTO Main
450   ON KEY 9 LABEL "Escape" GOTO Initialize
460   ON KEY 5 LABEL "ABORT RUN!" GOTO Escape
470   WHILE True
480   END WHILE
490 Massdisc:!
500   Sum_p=LOG(.1194/Meanraw(3))/LOG(Amu(3)/Amu(1))
510   IF Prog=2 THEN Sum_p=0              !No correction for spike run
520   Rcorr(1,Block)=1                                                    !86S/84T
530   Rcorr(2,Block)=(Meanraw(2)/Meanraw(3))*(Amu(2)/Amu(3))^Sum_p        !87/86
540   Rcorr(3,Block)=Meanraw(2)*(Amu(2)/Amu(1))^Sum_p                     !87/88
550   Rcorr(4,Block)=(Meanraw(4)/Meanraw(3))*(Amu(4)/Amu(3))^Sum_p        !84/86
560   Rcorr(5,Block)=SQR(2*Percent_dev(3)^2+Percent_dev(2)^2)
570   Rcorr(6,Block)=Block
580   Rcorr(7,Block)=1/Rcorr(5,Block)^2
590   RETURN
600 Spikesub: !
610   FOR I=2 TO 4
620     Meancorr(I)=Meanraw(I)
630   NEXT I
640   Sum_p=0
650   Iter=0
660   REPEAT
```

```
670       Alpha=(Meancorr(4)/Meancorr(3)-.056572)/(1-Meancorr(4)/Meancorr(3)*Rspike(1))
680       Beta=1/(1/Meancorr(3)+Alpha*Rspike(1)*(1/Meancorr(3)-1/Rspike(2)))
690       P_exp=LOG(.1194/Beta)/LOG(Amu(3)/Amu(1))
700       Tcorr(1)=1/Alpha
710       Tcorr(2)=(Meancorr(2)/Meancorr(3))*(Amu(2)/Amu(3))^P_exp
720       Tcorr(3)=Meancorr(2)*(Amu(2)/Amu(1))^P_exp
730       Sum_p=Sum_p+P_exp
740       Iter=Iter+1
750       Meancorr(4)=Meancorr(4)*(Amu(4)/Amu(1))^P_exp
760       Meancorr(3)=Tcorr(3)/Tcorr(2)
770       Meancorr(2)=Tcorr(3)
780     UNTIL ABS(P_exp)<1.E-5
790     DISP "Number of iterations was ";Iter
800     Rcorr(1,Block)=Tcorr(1)
810     Rcorr(2,Block)=Tcorr(2)+Alpha*Rspike(1)*(Tcorr(2)-Rspike(3)/Rspike(2))
820     Rcorr(3,Block)=Rcorr(2,Block)*.1194
830     Rcorr(4,Block)=.056572
840     Rcorr(5,Block)=SQR(2*Percent_dev(3)^2+Percent_dev(4)^2)
850     Rcorr(6,Block)=Block
860     Rcorr(7,Block)=1/Rcorr(5,Block)^2
870     RETURN
880   Printout3: !
890     Disc=((Amu(3)/Amu(1))^Sum_p-1)/2
900   Format3: IMAGE 5(4X,3D.7D)
910     IF Prog=3 THEN
920       PRINT
930       PRINT "Spike ratios:
";DROUND(Rspike(1),6);",";DROUND(Rspike(2),6);",";DROUND(Rspike(3),6)
940       PRINT "Number of iterations was ";Iter
950     END IF
960     PRINT
970     IF Interference THEN PRINT "87Rb subtracted; 87Rb/87Sr=
";DROUND(Netpeak(5,Scan)*.387/Netpeak(2,Scan),6)
980     PRINT "Mass discrimination corrected (p=";DROUND(Sum_p,6);",
alpha=";DROUND(Disc,6);")."
990     PRINT
1000    PRINT USING "5(8X,7A)";"86S/84T";" 87/86 ";" 87/88 ";" 84/86 ";"% Error"
1010    PRINT USING
Format3;Rcorr(1,Block);Rcorr(2,Block);Rcorr(3,Block);Rcorr(4,Block);Rcorr(5,Block)
1020    CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Block),4)
1030    RETURN
1040  Defaults:!
1050    Blockmax=30
1060    Intn(1)=5
1070    Intn(2)=7
1080    Intn(3)=7
1090    Intn(4)=7
1100    Intn(5)=9
1110    Delay(1)=5
1120    Delay(2)=6
1130    Delay(3)=4
1140    Delay(4)=4
1150    Delay(5)=6
1160    Pot(1)=3025
1170    Pot(2)=2680
1180    Pot(3)=2330
1190    Pot(4)=1630
1200    Pot(5)=1980
1210    Mass(1)=88
1220    Mass(2)=87
1230    Mass(3)=86
1240    Mass(4)=84
1250    Mass(5)=85
1260    Amu(1)=87.905625
1270    Amu(2)=86.908890
1280    Amu(3)=85.909273
```

```
1290  Amu(4)=83.913428
1300  Scan=11
1310  Factor=False
1320  Numpeak=4
1330  Interference=True
1340 Showdefaults:!
1350  CALL Field_values(X_gdu_max,Y_gdu_max,Numpeak,Interference,Mass(*),Pot(*))
1360  CALL
Parameters(Elem$,Check,Numpeak,Interference,Blockmax,Scan,Factor,Mass(*),Intn(*),Delay(*))
1370  IF Check THEN RETURN
1380  IF NOT Check THEN GOTO Showdefaults
1390 Spikeratios:!
1400  CALL Clear_crt
1410  OUTPUT 1
1420  OUTPUT 1
1430  OUTPUT 1;"1=Sr-Ca Mixed Spike 1 (salt)"
1440  OUTPUT 1;"2=Sr-Ca Mixed Spike 2 (mica)"
1450  OUTPUT 1;"3=Sr KF-2"
1460  INPUT "Which spike (type a number then ENTER)?",Sp
1470  IF Sp=1 THEN
1480    Rspike(1)=.003402            !86/84
1490    Rspike(2)=.50446             !86/88
1500    Rspike(3)=.13505             !87/88
1510  END IF
1520  IF Sp=2 THEN
1530    Rspike(1)=.076066
1540    Rspike(2)=5.6316
1550    Rspike(3)=.20517
1560  END IF
1570  IF Sp=3 THEN
1580    Rspike(1)=.0021282
1590    Rspike(2)=.30997
1600    Rspike(3)=.12975
1610  END IF
1620  IF Sp<>1 AND Sp<>2 AND Sp<>3 THEN Spikeratios
1630  RETURN
1640  REM Key_handlers
1650 Pause:!
1660  OFF KEY
1670  Halt=True
1680  RETURN
1690 Last_block:!
1700  OFF KEY
1710  Blockmax=Block
1720  RETURN
1730 Escape:!
1740  OFF KEY
1750  CALL Clear_crt
1760  DISP CHR$(131);"PROGRAM ABORTED"
1770  WAIT 5
1780  GOTO Lastline
1790 New_params:!
1800  OFF KEY
1810  Params=True
1820  RETURN
1830 Summary:!
1840  Index1=2
1850  Index2=5
1860  CALL Final_print(Sample$,Rcorr(*),6,Blockmax)
1870  CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Blockmax),4)
1880  CALL Summ_plot(Rcorr(*),Mean(*),Sigma(*),X_gdu_max,Y_gdu_max,Blockmax,Index1,Index2)
1890  DISP CHR$(128)
1900  LOOP
1910    Edit1=0
1920    INPUT "Number of blocks to be edited",Edit1
1930  EXIT IF Edit1=0
```

```
1940    CALL Edit(Rcorr(*),Mean(*),Sigma(*),Sigma_int,(Edit1),4,Blockmax)
1950  END LOOP
1960  RETURN
1970 Main:!
1980  OFF KEY
1990  MASS STORAGE IS ":,700,0"
2000  READ LABEL N$ FROM ":,700,0"
2010  A$="DATA"
2020  IF POS(UPC$(N$),A$)=0 THEN
2030    DISP "Insert a data disk in the left drive and press Continue."
2040    PAUSE
2050  END IF
2051  Today$=DATE$(TIMEDATE)
2052  Now$=TIME$(TIMEDATE)
2053  SELECT Today$[4,6]
2054    CASE "Mar","Jul"
2055      Mon$=Today$[4,4]&Today$[6,6]
2056    CASE ELSE
2057      Mon$=Today$[4,5]
2058  END SELECT
2059  File$=Mon$&Today$[10]&Today$[1,2]&Now$[1,2]
2060  CREATE BDAT File$,52,48
2070  ASSIGN @Data TO File$              !I/O path for disc file
2080  OUTPUT @Data,1;DATE$(TIMEDATE)
2090  OUTPUT @Data,2;Sample$
2100  CALL Start_up(Block$,Sample$,Block,Blockmax,Timbegin)
2110  CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2120  IF Interference THEN CALL
Measure_inter(Block$,Baseline(*),Ipeak(*),Itime(*),Delay(*),Mass(*),Intn(*),Pot(*))
2130  CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference,Ipeak(*),Itime(*))
2140  WHILE Block<=Blockmax                          !main program loop
2150    OFF KEY
2160    Halt=False
2170    Params=False
2180    CALL Center_peaks(Qpeak(*),Numpeak,Interference,Mass(*),Pot(*))
2190    Timfirst=INT((TIMEDATE MOD 86400)/60)-Timbegin
2200    Scantime=(SUM(Delay)+SUM(Intn))/60-(Delay(5)+Intn(5))/60
2210    Timlast=Timfirst+PROUND(Scantime*(Scan+4),0)
2220    CALL Plot(Qpeak(*),X_gdu_max,Y_gdu_max,Timfirst,Timlast)
2230    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
2240    CALL
Measure_peaks(Element$,Elem$,Block$,Baseline(*),Peak(*),Timep(*),Scan,Numpeak,Mass(*),Pot(
*),Delay(*),Intn(*),Timbegin,Block,Params)
2250    IF Params THEN GOTO Hold
2260    CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2270    IF Interference THEN CALL
Measure_inter(Block$,Baseline(*),Ipeak(*),Itime(*),Delay(*),Mass(*),Intn(*),Pot(*))
2280    ON KEY 9 LABEL "ABORT RUN!" GOTO Escape
2290    ON KEY 5 LABEL "FINISH" GOSUB Last_block
2300    ON KEY 2 LABEL "PAUSE" GOSUB Pause
2310    ON KEY 7 LABEL "Change Params" GOSUB New_params
2320    BEEP 880,2
2330    DISP CHR$(128);"Calculating results"
2340    CALL
Ratios(Traw$(*),Baseline(*),Timebase(*),Timep(*),Peak(*),Netpeak(*),Rraw(*),Avetime(*),Num
peak,Scan,Interference,Ipeak(*),Itime(*))
2350    CALL
Print_out1(Traw$(*),Baseline(*),Netpeak(*),Rraw(*),Factor,Numpeak,Interference,Block,Scan,
Mass(*),Pot(*))
2360    CALL
Statistics(Avetime(*),Rraw(*),Meanraw(*),Percent_dev(*),Numpeak,Scan,Rejected(*))
2370    CALL Print_out2(Meanraw(*),Percent_dev(*),Numpeak,Rejected(*),Mass(*))
```

```
2380     IF Prog=3 THEN
2390       GOSUB Spikesub
2400     ELSE
2410       GOSUB Massdisc
2420     END IF
2430     FOR B=1 TO 6
2440       Rstore(B)=Rcorr(B,Block)
2450     NEXT B
2460     OUTPUT @Data,Block+2;Rstore(*)
2470     CALL Grandstats(Rcorr(*),Mean(*),Sigma(*),Sigma_int,4,(Block))
2480     GOSUB Printout3
2490 Hold:!
2500     IF Halt THEN
2510       DISP CHR$(128);"Press CONTINUE to resume data acquisition."
2520       PAUSE
2530     END IF
2540     IF Params THEN
2550       CALL Clear_crt
2560       GOSUB Showdefaults
2570       DISP CHR$(128);"Press CONTINUE to resume data acquisition."
2580       PAUSE
2590     END IF
2600     Block=Block+1
2610     Block$=VAL$(Block)&" of "&VAL$(Blockmax)
2620     IF Params AND Block<=Blockmax THEN
2630       CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2640       IF Interference THEN CALL
Measure_inter(Block$,Baseline(*),Ipeak(*),Itime(*),Delay(*),Mass(*),Intn(*),Pot(*))
2650     END IF
2660     CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference,Ipeak(*),Itime(*))
2670   END WHILE
2680   OFF KEY
2690   GOSUB Summary
2700 Lastline: MASS STORAGE IS ":,700,1"
2710   CALL Clear_crt
2720   PRINT USING "@"
2730   LOAD "CONTROL"
2740   END
```

IRBDM

```
10     REM IRBDM                Brian D. Marshall  7/88
20     OPTION BASE 1
30     PRINTER IS CRT
40     CONTROL 1,4;0            !Turns off DISPLAY FUNCTIONS
50     LOADSUB ALL FROM "Subfile"
60     DIM A$[4],Block$[8],Elem$[2],Element$[10],File$[6],N$[6],Sample$[35],Traw$(5,24)[8]
70     INTEGER Block,Blockmax,Check,Delay(5),Edit1,Factor,False,Halt
80     INTEGER Index1,Index2,Interference,Intn(5),Iter,Mass(5),Numpeak,Params
90     INTEGER Pot(5),Rejected(5),Scan,Timbegin,Timfirst,Timlast,True
100    INTEGER P_mass(5),P_pot(5),Sel,Numsel
110    REAL Avetime(5,24),Baseline(5,2),Mean(5)
120    REAL Meanraw(5),Netpeak(5,24),Peak(5,24),Percent_dev(5)
130    REAL Qpeak(5),Rcorr(7,50),Rraw(5,24),Scantime
140    REAL Sigma(5),Sigma_int,Timebase(5,2),Timep(5,24)
150    REAL X_gdu_max,Y_gdu_max
160    CALL Clear_crt
170    Y_gdu_max=100*MAX(1,1/RATIO)
180    X_gdu_max=100*MAX(1,RATIO)
190    False=0
200    True= NOT False
210    Elem$="IR"
220    CALL Identify(Elem$,Sample$,X_gdu_max,Y_gdu_max)
230 Element:!
```

```
240   OFF KEY
250   INPUT "Element to be analyzed (U, TH, PB, RE, RB, K, SR, CA)?",El$
260   Elem$=UPC$(El$)
270   SELECT Elem$
280   CASE ="U"
290     Element$="Uranium"
300     P_mass(1)=233
310     P_mass(2)=234
320     P_mass(3)=235
330     P_mass(4)=236
340     P_mass(5)=238
350     P_pot(1)=1000
360     P_pot(2)=1500
370     P_pot(3)=2000
380     P_pot(4)=2500
390     P_pot(5)=3000
400     Numsel=5
410     GOSUB Select
420   CASE ="TH"
430     Element$="Thorium"
440     Numpeak=2
450     REDIM Intn(2),Delay(2),Mass(2),Pot(2)
460     Mass(1)=230
470     Mass(2)=232
480     Pot(1)=700
490     Pot(2)=1350
500   CASE ="RB"
510     Element$="Rubidium"
520     Numpeak=2
530     REDIM Intn(2),Delay(2),Mass(2),Pot(2)
540     Mass(1)=85
550     Mass(2)=87
560     Pot(1)=1952
570     Pot(2)=2655
580   CASE ="K"
590     Element$="Potassium"
600     P_mass(1)=39
610     P_mass(2)=40
620     P_mass(3)=41
630     P_pot(1)=900
640     P_pot(2)=1350
650     P_pot(3)=1860
660     Numsel=3
670     GOSUB Select
680   CASE ="RE"
690     Element$="Rhenium"
700     Numpeak=2
710     REDIM Intn(2),Delay(2),Mass(2),Pot(2)
720     Mass(1)=185
730     Mass(2)=187
740     Pot(1)=1000
750     Pot(2)=2000
760   CASE ="PB"
770     Element$="Lead"
780     P_mass(1)=204
790     P_mass(2)=205
800     P_mass(3)=206
810     P_mass(4)=207
820     P_mass(5)=208
830     P_pot(1)=1000
840     P_pot(2)=1500
850     P_pot(3)=2000
860     P_pot(4)=2500
870     P_pot(5)=3000
880     Numsel=5
890     GOSUB Select
```

```
900   CASE "SR"
910     Element$="Strontium"
920     P_mass(1)=84
930     P_mass(2)=86
940     P_mass(3)=87
950     P_mass(4)=88
960     P_pot(1)=1545
970     P_pot(2)=2260
980     P_pot(3)=2610
990     P_pot(4)=2960
1000    Numsel=4
1010    GOSUB Select
1020  CASE "CA"
1030    Element$="Calcium"
1040    P_mass(1)=40
1050    P_mass(2)=42
1060    P_mass(3)=43
1070    P_mass(4)=44
1080    P_mass(5)=48
1090    P_pot(1)=1700
1100    P_pot(2)=2600
1110    P_pot(3)=3040
1120    P_pot(4)=3475
1130    P_pot(5)=4000
1140    Numsel=5
1150    GOSUB Select
1160  CASE ELSE
1170    GOTO Element
1180  END SELECT
1190  CALL Clear_crt
1200  PRINT
1210  PRINT "Selected ratios:"
1220  FOR R=2 TO Numpeak
1230    PRINT
1240    PRINT Mass(R);"/";Mass(1)
1250  NEXT R
1260  ON KEY 2 LABEL "OK" GOTO Initialize
1270  ON KEY 9 LABEL "Re-select" GOTO Element
1280  WHILE True
1290  END WHILE
1300 Select:!
1310  CALL Clear_crt
1320  INPUT "Number of isotopes to be measured?",Numpeak
1330  IF Numpeak<2 OR Numpeak>5 THEN GOTO Element
1340  REDIM Intn(Numpeak),Delay(Numpeak),Pot(Numpeak),Mass(Numpeak)
1350  PRINT
1360  PRINT
1370  FOR I=1 TO Numsel
1380    PRINT I;"=";P_mass(I)
1390  NEXT I
1400  INPUT "Select reference isotope (type a number then ENTER)?",Sel
1410  IF Sel<1 OR Sel>Numsel THEN GOTO Element
1420  PRINT TABXY(10,Sel+2);"Peak  1 (reference)";
1430  Mass(1)=P_mass(Sel)
1440  Pot(1)=P_pot(Sel)
1450  FOR N=2 TO Numpeak
1460    INPUT "Select next isotope to be measured (type a number then ENTER)?",Sel
1470    IF Sel<1 OR Sel>Numsel THEN GOTO Element
1480    PRINT TABXY(10,Sel+2);"Peak ";N;
1490    Mass(N)=P_mass(Sel)
1500    Pot(N)=P_pot(Sel)
1510  NEXT N
1520  RETURN
1530 Initialize:!
1540  PRINTER IS PRT
1550  OFF KEY
```

```
1560  GRAPHICS OFF
1570  GOSUB Defaults
1580  ON KEY 2 LABEL "START" GOTO Main
1590  ON KEY 9 LABEL "Escape" GOTO Initialize
1600  ON KEY 5 LABEL "ABORT RUN!" GOTO Escape
1610  WHILE True
1620  END WHILE
1630 Massdisc:!
1640  Rcorr(1,Block)=Meanraw(2)
1650  Rcorr(2,Block)=Meanraw(3)
1660  Rcorr(3,Block)=Meanraw(4)
1670  Rcorr(4,Block)=Meanraw(5)
1680  Rcorr(5,Block)=Percent_dev(2)
1690  Rcorr(6,Block)=Block
1700  Rcorr(7,Block)=1/Rcorr(5,Block)^2
1710  RETURN
1720 Printout3:!
1730  CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Block),Numpeak-1)
1740  RETURN
1750 Defaults:!
1760  Blockmax=10
1770  FOR N=1 TO Numpeak
1780    Intn(N)=5
1790    Delay(N)=5
1800  NEXT N
1810  Scan=11
1820  Factor=False
1830  Interference=False
1840 Showdefaults:!
1850  CALL Field_values(X_gdu_max,Y_gdu_max,Numpeak,Interference,Mass(*),Pot(*))
1860  CALL
Parameters(Elem$,Check,Numpeak,Interference,Blockmax,Scan,Factor,Mass(*),Intn(*),Delay(*))
1870  IF Check THEN RETURN
1880  IF NOT Check THEN GOTO Showdefaults
1890  REM Key_handlers
1900 Pause:!
1910  OFF KEY
1920  Halt=True
1930  RETURN
1940 Last_block:!
1950  OFF KEY
1960  Blockmax=Block
1970  RETURN
1980 Escape:!
1990  OFF KEY
2000  CALL Clear_crt
2010  DISP CHR$(131);"PROGRAM ABORTED"
2020  WAIT 5
2030  GOTO Lastline
2040 New_params:!
2050  OFF KEY
2060  Params=True
2070  RETURN
2080 Summary:!
2090  Index1=1
2100  Index2=5
2110  CALL Final_print(Sample$,Rcorr(*),6,Blockmax)
2120  CALL Print_means(Mean(*),Sigma(*),Sigma_int,(Blockmax),Numpeak-1)
2130  CALL Summ_plot(Rcorr(*),Mean(*),Sigma(*),X_gdu_max,Y_gdu_max,Blockmax,Index1,Index2)
2140  DISP CHR$(128)
2150  LOOP
2160    Edit1=0
2170    INPUT "Number of blocks to be edited",Edit1
2180  EXIT IF Edit1=0
2190    CALL Edit(Rcorr(*),Mean(*),Sigma(*),Sigma_int,(Edit1),Numpeak-1,Blockmax)
2200  END LOOP
```

```
2210  RETURN
2220 Main:!
2230  OFF KEY
2240  PRINTER IS PRT
2250  CALL Start_up(Block$,Sample$,Block,Blockmax,Timbegin)
2260  CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2270  CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
2280  WHILE Block<=Blockmax
2290    OFF KEY
2300    Halt=False
2310    Params=False
2320    CALL Center_peaks(Qpeak(*),Numpeak,Interference,Mass(*),Pot(*))
2330    Timfirst=INT((TIMEDATE MOD 86400)/60)-Timbegin
2340    Scantime=(SUM(Delay)+SUM(Intn))/60
2350    Timlast=Timfirst+PROUND(Scantime*(Scan+4),0)
2360    CALL Plot(Qpeak(*),X_gdu_max,Y_gdu_max,Timfirst,Timlast)
2370    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
2380    CALL
Measure_peaks(Element$,Elem$,Block$,Baseline(*),Peak(*),Timep(*),Scan,Numpeak,Mass(*),Pot(
*),Delay(*),Intn(*),Timbegin,Block,Params)
2390    IF Params THEN GOTO Hold
2400    CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2410    ON KEY 9 LABEL "ABORT RUN!" GOTO Escape
2420    ON KEY 5 LABEL "FINISH" GOSUB Last_block
2430    ON KEY 2 LABEL "PAUSE" GOSUB Pause
2440    ON KEY 7 LABEL "Change Params" GOSUB New_params
2450    BEEP 880,2
2460    DISP CHR$(128);"Calculating results"
2470    CALL
Ratios(Traw$(*),Baseline(*),Timebase(*),Timep(*),Peak(*),Netpeak(*),Rraw(*),Avetime(*),Num
peak,Scan,Interference)
2480    CALL
Print_out1(Traw$(*),Baseline(*),Netpeak(*),Rraw(*),Factor,Numpeak,Interference,Block,Scan,
Mass(*),Pot(*))
2490    CALL
Statistics(Avetime(*),Rraw(*),Meanraw(*),Percent_dev(*),Numpeak,Scan,Rejected(*))
2500    CALL Print_out2(Meanraw(*),Percent_dev(*),Numpeak,Rejected(*),Mass(*))
2510    GOSUB Massdisc
2520    CALL Grandstats(Rcorr(*),Mean(*),Sigma(*),Sigma_int,Numpeak-1,(Block))
2530    GOSUB Printout3
2540 Hold:!
2550    IF Halt THEN
2560      DISP CHR$(128);"Press CONTINUE to resume data acquisition."
2570      PAUSE
2580    END IF
2590    IF Params THEN
2600      CALL Clear_crt
2610      GOSUB Showdefaults
2620      DISP CHR$(128);"Press CONTINUE to resume data acquisition."
2630      PAUSE
2640    END IF
2650    Block=Block+1
2660    Block$=VAL$(Block)&" of "&VAL$(Blockmax)
2670    IF Params AND Block<=Blockmax THEN CALL
Measure_zeros(Block$,Element$,Baseline(*),Timebase(*),Numpeak,Interference,Mass(*),Delay(*
),Intn(*),Pot(*))
2680    CALL New_zeros(Baseline(*),Timebase(*),Numpeak,Interference)
2690  END WHILE
2700  OFF KEY
2710  GOSUB Summary
2720 Lastline:!
2730  CALL Clear_crt
```

```
2740  PRINT USING "@"
2750  LOAD "CONTROL"
2760  END
```

References

Dodson M.H. (1978) A linear method for second-degree interpolation in cyclical data collection. *J. Phys. E: Sci. Instrum.* **11**, 296.

Taylor J.R. (1982) *An introduction to error analysis: the study of uncertainties in physical measurements.*University Science Books, Mill Valley, 270 p.

Wasserburg G.J., Jacobsen S.B., DePaolo D.J., McCulloch M.T. and Wen T. (1981) Precise determination of Sm/Nd ratios, Sm and Nd isotopic abundances in standard solutions. *Geochim. Cosmochim. Acta* **45**, 2311-2323.