ORNL/TM--9105

DE86 002903

Contract No. DE-AC05-840R21400

PHYSICS DIVISION

"LIFETIME": A COMPUTER PROGRAM FOR ANALYZING DOPPLER-SHIFT RECOIL-DISTANCE NUCLEAR LIFETIME DATA

John C. Wells[†]
Matthew P. Fewell
Noah R. Johnson

Date Published - October 1985

[†]Tennessee Technological University, Cookeville, TN 38505
and Oak Ridge National Laboratory

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY

ep

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	THE INPUT FILES	5
	A. Data File	5
	B. Parameter File	12
	C. Velocity Data File	13
III.	HOW TO RUN THE PROGRAM	15
	A. Interactive Mode	15
	B. Batch Mode	16
	C. The Commands	16
IV.	HOW TO USE THE PROGRAM	25
V.	ORGANIZATION OF THE PROGRAM	33
VI.	SUBROUTINES AND FUNCTIONS	39
	A. LIFETIME.FTN	39
	B. MINUIT.FTN	66
	C. Other Functions and Subroutines	78
VII.	COMMON VARIABLES AND DIMENSIONS OF ARRAYS	79
	A. COMMON Variables in LIFETIME	79
	B. COMMON Variables in MINUIT	80
	C. Dimensions of Arrays	82
VIII.	SAMPLE OUTPUT	83
	ACKNOWLEDGMENTS	90
	REFERENCES	91

LIST OF FIGURES

		Page
Figure 1.	Level scheme for the sample case.	26
Figure 2.	Flow chart for the program LIFETIME.	34-35
Figure 3.	Schematic spectrum showing typical shifted and unshifted photopeaks.	46
Figure 4.	Schematic cross section of a Ge(Li) detector.	56
Figure 5.	Calculation of MINOS errors of parameter X.	73

"LIFETIME": A COMPUTER PROGRAM FOR ANALYZING DOPPLER-SHIFT RECOIL-DISTANCE NUCLEAR LIFETIME DATA

John C. Wells[†]
Matthew P. Fewell[††]
Noah R. Johnson

I. INTRODUCTION

For many years, the Doppler-shift recoil-distance method has served as a very reliable method for the determination of lifetimes of excited nuclear states. It was the advent of high-resolution γ-ray detectors coupled with the development of accelerators offering a variety of energetic heavy-ion beams that enabled this technique to achieve its present prominence and sophistication. Lifetimes ranging from about one picosecond (ps) to several hundred ps can be determined in such recoil-distance measurements.

Until recently, most recoil-distance lifetime measurements were carried out following heavy-ion Coulomb excitation. The analysis of such data, especially for deformed nuclei where the lifetime of each successively-higher-spin member of a band is shorter than that of the state below it, is accomplished in a relatively simple manner. This is not the case, however, for the data collected following a compound-nucleus reaction. Here the problem is complicated by the fact that a given level often has appreciable side feeding from unseen transitions from the γ-ray continuum. Further, this side feeding frequently has

[†]Permanent address: Tennessee Technological University
Cookeville, TN 38505
[††]Permanent address: The Australian National University
Canberra ACT, 2600, Australia

long lifetimes associated with it which can cause ambiguities in the solutions of the Bateman equations describing the decay chain. These problems can be surmounted, however, and we describe here a computer program which handles these complexities, thus opening the opportunities to obtain very valuable model-free matrix elements for transitions connecting excited states in nuclei.

The program LIFETIME described in this report is designed to extract lifetimes of nuclear levels from Doppler-shift recoil-distance experiments by performing a least-squares fit to the experimental data (shifted and unshifted photopeak intensities and branching ratios). The starting point for our program was the program MASTER supplied to us from G.S.I., Darmstadt, by Hans Emling.[1] It incorporates the program MINUIT, a general-purpose program for function minimization and parameter error analysis developed at CERN by F. James and M. Roos.[2] Initial populations of levels and transition rates between levels are treated as variable parameters. In terms of these parameters, the population of each level as a function of time is determined by the Bateman equations, and the calculated values of the shifted and unshifted intensities are obtained.

Included in LIFETIME is the option of making a number of corrections to the data. These consist of corrections for position- and velocity-dependent changes in the solid angle subtended by the detector, for the effect of attenuation of alignment during flight, and for

γ rays emitted during the slowing-down time resulting in an energy distribution between the shifted and unshifted energies. Since these corrections depend on the parameters which are being varied, the program provides the option of making them either at every iteration, which can be very time-consuming, or just once with the initial values of the parameters.

Very often the side-feeding pattern to the levels of interest is not known. Thus, it becomes necessary to define additional levels from which feeding transitions occur. In this program, the initial population and transition rate for each modeled feeding level are treated as variable parameters. A further option, which is available, is to assume a rotational band to model the feeding pattern to a particular level (e.g., the highest level in a cascade). The transition rates λ_{jk} (in ps^{-1}) are defined in terms of a quadrupole moment Q (in e•b) and transition energies E_{jk} (in keV) for $K_j = K_k = 0$ as:

$$\lambda_{jk} = 1.2253 \times 10^{-15} E_{jk}^{5} Q^2 (J_j\ 2\ 0\ 0 \mid J_k\ 0).$$

Here, Q and the initial population of the highest level in the band are treated as variable parameters. Experimental values for transition energies may be used.

The possibility that the measured zero target-plunger distance is not the true zero is considered by subtracting a term D_0 from each flight distance. The quantity D_0 is treated as a variable parameter in the least-squares fit. Further, if there are two or three

transitions out of a given level and the branching ratios are known experimentally, these branching ratios may be entered as data and will be included in the least-squares fit.

If a photopeak is known to be a multiplet consisting of two or more transitions in the same nucleus and if the positions of these transitions in the level scheme are known, the program can be instructed to sum the calculated intensities of the transitions and to make a multi-component fit for the lifetimes.

Sometimes it is important in the lifetime analyses to utilize data from different experiments; e.g., from experiments involving different reactions to give the same final nucleus. The program LIFETIME has been written so that such data from two or more different experiments can be simultaneously included in the analysis if it can be assumed that the relative intensity of each transition is the same in all of the experiments. A normalization factor for each experiment is treated as a variable parameter.

The recoil velocity with solid-angle corrections is calculated from the shifted and unshifted energies if an energy calibration and data on the shifted and unshifted peak centroids are provided as input information.

The program LIFETIME is written in FORTRAN 77 and versions are available for the Perkin-Elmer 3230, VAX 11/780, and UNIVAC 1100/82 computers.

II. THE INPUT FILES

A. DATA FILE

Each line in the Data File begins with a data type identifier (which must begin in column 1) that identifies the data on the line. All lines are in free format. Allowable delimiters are blank, comma, semi-colon, =, <, and >. Zero values may be omitted if no non-zero values follow. Energy levels are referred to by assigning each one a level identifier, which is a character constant. Each level ID (or group of level ID's) must be enclosed in parentheses in order to be decoded as a character constant rather than as a number. Level ID's may be composed of up to four numbers, letters, or symbols, except for the above-mentioned delimiters.

Blank lines are ignored, and may be included to make the file more readable. Each data type identifier may be abbreviated by the first two letters. A description of each line follows.

NUCLEUS, ZR, AR

NUCLEUS: Data type.

ZR: Atomic number of recoil nucleus.

AR: Atomic mass of recoil nucleus in u.

This line should be first.

LEVEL, (LID(1), LID(2), ..., LID(N))

LEVEL: Data type.

LID(I): Level ID.

This line should be second, and must come before any other reference to Level ID's. The Level ID's must be ordered so that each succeeding level decays to a level or levels already listed. If all the Level ID's cannot be listed on one line, they may be continued on additional lines, each beginning with the data type identifier LEVEL. The maximum number of levels cannot exceed NL (see Sec. VII. C).

OPTIONS, ICOR, IPOS, IVEL, IALN, ISHP, IVOL, IPLT, NPTS, NPTG

OPTIONS: Data type.

ICOR = 0 Corrections are not made.

= 1 Corrections are made initially only.

= 2 Corrections are made every iteration.

IPOS = 0 Positional solid-angle corrections are not made.

= 1 Positional solid-angle corrections are made.

IVEL = 0 Velocity solid-angle corrections are not made.

= 1 Velocity solid-angle corrections are made.

IALN = 0 Alignment attenuation corrections are not made.

= 1 Alignment attenuation corrections are made.

ISHP = 0 Lineshape corrections are not made.

= 1 Lineshape corrections are made.

IVOL = 0 Detector surface is assumed to be black for all γ-ray energies.

= 1 Solid angle is calculated by integrating over the volume of the detector.

IPLT = 0 Line-printer plots are not made automatically.

= 1 Line-printer plots are made automatically, 6 lines per inch.

= 2 Line-printer plots are made automatically, 10 lines per inch.

NPTS: Number of points in numerical integration along flight path.
NPTG: Number of points in numerical integration over detector volume. Zero values may be omitted if no non-zero values follow. If NPTS or NPTG is specified as zero, its default value is 11. If this line is omitted, the default values are: OPTIONS, 1, 1, 1, 1, 1, 1, 0, 11, 11

ALIGNMENT, TAU2, TAU4
ALIGNMENT: Data type.
TAU2: Relaxation time for alignment attenuation coefficient α_2 in ps.
TAU4: Relaxation time for alignment attenuation coefficient α_4 in ps.
If this line is omitted, the default values are: ALIGNMENT, 30, 10.

BAND, (LBND), (LFST), AJB, NSTATES, EGB(1), ..., EGB(NSTATES)
BAND: Data type.
(LBND): Level ID of level being fed by band.
(LFST): Level ID of first level in band.
AJB: Spin of first level in band.
NSTATES: Number of levels in band.
EGB(N): Transition energies, in keV, starting with the transition to level LBND and proceeding up the band. If zero, these will be determined internally with EGB(1) = 500 keV.

Level ID's must be enclosed in parentheses. This line is omitted if there are no feeder bands. If there are more than one band, then there must be one BAND line for each. The maximum number of bands cannot exceed NB, and, for each band, NSTATES < NS (see Sec. VII. C).

BRANCH, (IBR), (JBR(1)), BR(1), BRR(1), (JBR(2)), BR(2), BRR(2), ...

BRANCH: Data type.

(IBR): Level ID of initial level.

(JBR(K)): Level ID of final level.

BR(K): Branching ratio.

BRR(K): Error in branching ratio.

Level ID's must be enclosed in parentheses. Zero values may be omitted if no non-zero values follow. A maximum of four branching ratios for each initial level is permitted. This line is omitted if there are no branching ratios or repeated if there are more than one level for which branching ratios are available. The total number of branching ratios cannot exceed NR (see Sec. VII. C).

EXPERIMENT, L

EXPERIMENT: Data type.

L: Experiment or data set number.

Experiments must be numbered sequentially. All data following this line and before the next EXPERIMENT line are assigned to experiment L. If there is only one experiment, this line may be omitted. The number of experiments cannot exceed NE (see Sec. VII. C).

DETECTOR, TDD, DL, DRO, DRI

DETECTOR: Data type.

TDD: Target-detector distance in cm.

DL: Detector length in cm.

DRO: Detector outer radius in cm.

DRI: Radius of inactive core of detector in cm.

If these values are the same for all experiments, this line need be included for the first experiment only.

DISTANCE, DIST(1), DIST(2), ..., DIST(NDIST)

DISTANCE: Data type.

DIST(I): Recoil flight distance in μm.

If NDIST distances cannot be listed on one line, they may be continued on additional lines with the same data type identifier. NDIST cannot exceed ND (see Sec. VII. C).

VELOCITY, BETA, DBETA, ZS, AS, RHOST, FOIL, BETA1, DBETA1, FWHM, STIME

VELOCITY: Data type.

BETA: Recoil velocity, v/c (Default value is 0.01).

DBETA: Error in recoil velocity.

ZS: Atomic number of stopper material, may be set to zero for Pb.

AS: Atomic mass of stopper material in u, may be set to zero for ^{208}Pb.

RHOST: Density of stopper material in g/cm^3, may be set to zero for Pb or Au.

FOIL: Thickness of slowing foil in mg/cm^2. Complete stopping is indicated by setting FOIL = 0.

BETA1: Slowed velocity, v/c, for retarding foils.

DBETA1: Error in slowed velocity.

FWHM: Full width at half maximum in keV of an unshifted peak at 500 keV (Default value is 1.5 keV).

STIME: Stopping time or slowing down time. (If STIME = 0.0, it will be calculated by the program.)

Zero values may be omitted if no non-zero values follow.

TRANSITION, (LNI, LNF), EGAM, AJI, AJF, IFIT, ALF

TRANSITION: Data type.

(LNI): Level ID of initial level.

(LNF): Level ID of final level.

EGAM: Transition energy in keV.

AJI: Spin of initial level.

AJF: Spin of final level.

IFIT = 0 Do not include this transition.

= 1 Include only unshifted data.

= 2 Include only shifted data.

= 3 Include both shifted and unshifted data.

AFL: Internal conversion coefficient (If zero, it will be calculated internally if internal table exists; see page 42.)

Level ID's must be enclosed in parentheses.

If this γ ray is a multiplet (i.e. more than one transition with energies too close together to be resolved), then this line must be followed immediately by one or more MULTIPLET lines for the second, third, etc. member of the multiplet.

MULTIPLET, (LNI, LNF), AJI, AJF

MULTIPLET: Data type.

The others are the same as for the TRANSITION line. Level ID's must be enclosed in parentheses. The total number of data curves cannot exceed NC, and the number of transitions, counting each member of any multiplet separately, cannot exceed NT (see Sec. VII. C).

The TRANSITION (or MULTIPLET) line must be followed immediately by a set of INTENSITY lines, one for each flight distance, in the same order as in the DISTANCE line.

INTENSITY, PS, DPS, PU, DPU

INTENSITY: Data type.

PS: Shifted peak intensity.

DPS: Error in shifted peak intensity.

PU: Unshifted peak intensity.

DPU: Error in unshifted peak intensity.

There will normally be NDIST lines, where NDIST is the number of flight distances. However, lines with all zeros may be omitted if no lines with non-zero entries follow. If DPS = 0, then the shifted data point is omitted from the fit. If DPU = 0, then the unshifted data point is omitted.

END

The last line in the Data File must be END.

B. PARAMETER FILE

The first line in the Parameter File is a TITLE, up to 70 characters, which will be reproduced on the output. Each line after the first begins with a parameter type identifier (which must begin in column 1) that identifies the parameter information on the line. All lines are in free format. Allowable delimiters are blank, comma, semicolon, =, <, and >. Zero values may be omitted if no non-zero values follow. Level ID's must be enclosed in parentheses. Each parameter type identifier may be abbreviated by the first letter. There must be one line for each parameter, as follows:

TYPE, IDENT, PARA, STEP, ALO, AHI

TYPE = TRANS - transition rate.

POP - initial population.

QUAD - band quadrupole moment.

DZERO - distance zero offset.

NORM - data set normalization factor.

END - end of parameter file, must be last.

IDENT = (LNI, LNF), initial and final level ID's of transition rate.

(LNI), level ID of initial population.

(LBND), level ID of level being fed by band.

L, experiment number for zero distance or normalization.

PARA = starting value for the parameter. Default value is zero for all parameter types except NORM, for which it is one.

STEP = initial step size for minimization search. If STEP = 0, the parameter will be permanently fixed.

ALO = lower bound on parameter.

AHI = upper bound on parameter.

If both ALO = 0 and AHI = 0, the parameter is unbounded. The last line must have TYPE = END. The maximum number of parameters is NP, and of variable parameters is NV (see Sec. VII. C).

C. VELOCITY DATA FILE

The format for the Velocity Data File is the same as for the Data File (see Sec. II. A). A description of each line follows.

EXPERIMENT, L, AA, AM, AB

EXPERIMENT: Data type.

L: Experiment or data set number.

AA, AM, AB: Coefficients in the energy calibration

E = AA + AM*C + AB*C**2, where C is the channel number and E is the energy in keV.

Experiment numbers must agree with those in the Data File. All data following this line and before the next experiment line are assigned to experiment L. Calibration coefficient default values are AA = 0.0, AM = 1.0, AB = 0.0.

TRANSITION, (LNI, LNF)

TRANSITION: Data type.

(LNI): Level ID of initial level.

(LNF): Level ID of final level

Level ID's must be enclosed in parentheses.

CENTROID, CS, DCS, CU, DCU, CL, DCL

CENTROID: Data type.

CS: Centroid channel number of fully shifted peak.

DCS: Error in fully shifted centroid.

CU: Centroid channel number of unshifted peak.

DCU: Error in unshifted centroid.

CL: Centroid channel number of less shifted peak.

DCL: Error in less shifted centroid.

The TRANSITION line must be followed immediately by a set of CENTROID lines, one for each flight distance, in the same order as in the DISTANCE line for experiment L which appears in the Data File. There will normally be NDIST lines, where NDIST is the number of flight distances. However, lines with all zeros may be omitted if no lines with non-zero entries follow. Any centroid entered as zero (or negative) is omitted from the calculation. Any error entered as zero (or negative) is set by default to 0.05.

END

The last line in the Velocity Data File must be END.

III. HOW TO RUN THE PROGRAM

For definiteness, this section refers to the version on the Perkin-Elmer computer. The other versions are essentially similar.

A. TO RUN THE PROGRAM IN INTERACTIVE MODE FROM THE TERMINAL:

Set up the input files with a single file name and different extensions:

Parameter file: FILENAME.XXX

Data file: FILENAME.YYY

Velocity file: FILENAME.ZZZ

(For example: YB160.P01, YB160.D01, YB160.V01)

At the terminal, type LIFE FILENAME,XXX,YYY,ZZZ. If no velocity calculations are intended, the last extension, ZZZ, may be omitted.

After reading the input and performing the initial calculations, the program will ask: WHAT NEXT? At this point, one may enter from the terminal one of 24 possible commands, which are described in Sec. III. C. Following some of the commands, numerical arguments may be specified (on the same line, separated by blanks or commas). Zero arguments may be omitted if no non-zero values follow. Some commands will ask for additional information which should be entered as requested. As each stage in the program is completed, the message WHAT NEXT? will appear on the terminal. At that time the user may enter the next desired command.

B. TO RUN THE PROGRAM IN BATCH MODE:

Set up two additional files: FILENAME.JOB and FILENAME.CMD. The JOB file, FILENAME.JOB, has three lines:

```
S User id, acct no, password
LIFE FILENAME,XXX,YYY,[ZZZ],FILENAME.CMD,NULL:
SIGNOF
```

The CMD file, FILENAME.CMD, contains the commands to be executed, one per line. At the terminal, type SUBMIT FILENAME.JOB. The resulting output will be saved in FILENAME.PRT.

C. THE COMMANDS:

ALTER

This gives the user the opportunity to change the corrections being made, the data curves and branching ratios being included in the fit, and the highest and lowest levels included in the calculations. In interactive mode, the instructions for making these changes will appear on the terminal. In batch mode, four lines must follow the command ALTER in the CMD file (or three lines if there are no branching ratios). On the first line, specify the new values for ICOR, IPOS, IVEL, IALN, ISHP, and IVOL. On the second line, enter the data curve number and value of IFIT for each data curve to be changed (e.g. 1,1,2,1,3,1 if data curves 1, 2, and 3 are to have only unshifted data fitted). Data curves are numbered in the order in which they appear in the data file. Enter 0,0 if there are no changes. On the third line, enter the branching ratio number and the number 1 (include in

fit) or 0 (do not include) for each branching ratio to be changed. Enter 0,0 if there are no changes. Omit this line if there are no branching ratios. On the fourth line, enter the new LID's for LMIN, LMAX in parentheses.

CALL

The value of χ^2 (FCN) is calculated and displayed on the terminal.

CORRECT

The corrections to the data are calculated using the current parameters. This command may be used to periodically update these corrections when using option ICOR = 1.

END

The program is terminated with a full printout of the final results. If IPLT > 0, printer plots are produced also.

ERROR, arg1

The value of arg1 becomes the value of the variable UP, a confidence interval used by the subroutines SIMPLEX, MIGRAD, and MINOS. For example, as FCN is a chi-squared function, if the user desires a 68% confidence interval, then UP = 1, which is the default value.

EXIT

The program is terminated with no final printout or plots.

FIX

Parameters are removed from the variable list, and their values fixed at the values they have at the time the command is executed. No more than NF parameters may be fixed at one time (see Sec. VII. C). - interactive mode, the instructions for specifying the parameters to be fixed will appear on the terminal. In batch mode, on the line in the CMD file following the command FIX, specify the parameter numbers of parameters to be fixed, separated by blanks or commas. A range of numbers may be designated with a dash (e.g. 1,3,5-8 means that parameters 1, 3, 5, 6, 7, and 8 will be fixed). Parameters are numbered in the order in which they appear in the parameter file.

HESSE

The covariance matrix is calculated. If a covariance matrix already exists, it is replaced by the new one.

MATOUT

MATOUT prints the covariance matrix, and calculates and prints the individual correlation coefficients, which are the normalized off-diagonal elements of the covariance matrix.

MIGRAD arg1, arg2, arg3

The program performs a minimization using the MIGRAD technique, which is described in Section VI. B. The command may have three arguments, which control the convergence.

arg1: NFCNMX - Maximum number of calls to FCN. The minimization will be stopped after this number of calls even if convergence has not been attained. If NFCNMX=0, there is no maximum.

arg2: EPSI - Tolerance on the minimum FCN value. The program has the ability to predict, from the covariance matrix and the first derivatives of FCN, how far it is vertically from the true minimum. When this predicted distance SIGMA becomes smaller than EPSI for two consecutive iterations, this convergence criterion is satisfied. The default value = 0.1 UP (see ERROR).

arg3: VTEST - Tolerance on the stability of the error matrix. The program calculates the average fractional change in the diagonal elements of the covariance matrix from one iteration to the next. When this quantity is smaller than VTEST for two consecutive iterations, this convergence criterion is satisfied. The default value is 0.04.

Convergence is attained when both the EPSI and VTEST criteria are satisfied simultaneously, or when SIGMA (defined above) becomes less than 10^{-5} EPSI.

MINOS, arg1, arg2, arg3

A MINOS error analysis is performed on specified parameters. The arguments are the same as for the command MIGRAD: NFCNMX, EPSI, and VTEST. (However, here VTEST has a default value of 0.20.) In interactive mode, the program will ask you to specify the parameter numbers of the parameters for which a MINOS analysis is desired. In batch mode, this must be done on the line in the CMD file following the MINOS

command. A range of parameter numbers may be designated with a dash (e.g. 1-3 means parameters 1, 2, and 3 will be analyzed). Parameters are numbered in the order in which they appear in the parameter file.

MINPLOT, arg1, arg2, arg3, arg4, arg5

The MINOS subroutine calculates and produces a printer plot of FCN as a function of a selected parameter at ten equally spaced values of the parameter on each side of its best value. At each step, FCN is reminimized by varying all the other parameters.

arg1: parameter number of parameter to be plotted.

arg2: range of plot on each side of best value (If zero, MINOS will determine range.)

arg3: NFCNMX

arg4: EPSI

arg5: VTEST

These last three have the same default values as for the MINOS command.

OUTPUT, arg1

If arg1 = 1, the values of the parameters are printed in full whenever SEEK, SIMPLEX, or MIGRAD is executed. Also, a table of populations of levels as a function of time is included in the final output. If arg1 = 0, these are not printed. (The default value is 0.)

PAGE

The printout begins a new page at the point this command is executed.

PLOT, arg1

A printer plot is printed immediately without terminating the program. If arg1 = 1, the plot is adjusted for 6 lines per inch; if arg1 = 2, for 10 lines per inch. If no argument is given, the default value is IPLT from the input data file. (The line printer of the Perkin-Elmer computer must be set manually for 6 or 10 lines per inch before executing this command).

PRINT

The final output is printed immediately without terminating the program. It is useful if one wants an intermediate output to determine what to do next.

RELEASE

Fixed parameters are made variable again. In interactive mode instructions for specifying the parameters to be released will appear on the terminal. RELEASE can operate only on parameters which were originally variable and were fixed by a previous FIX command. If a covariance matrix exists at the time of the execution of a RELEASE command, only the diagonal element for the restored parameter is estimated. In batch mode, on the line in the CMD file following the command RELEASE, specify the parameter numbers of the parameters to be released, separated by blanks or commas. A range of numbers may be designated with a dash (e.g. 1, 3, 5-8 means that parameters 1, 3, 5, 6, 7, and 8 will be released).

RESTORE, arg1

This command is similar to RELEASE, except that either all previously fixed parameters are returned to variable status, or only the last parameter fixed becomes variable.

arg1 = 0: All previously fixed parameters are released.

arg2 = 1: The last parameter fixed is released.

SAVE

SAVE causes the current values of the parameters to be stored in a file in a format such that the file can be used as the Input Parameter File when the program is run again. In interactive mode, after the command SAVE is entered, the program asks for the name of the file in which to store the parameters. If this file already exists, the program asks if the user wishes to overwrite it. The user is given the opportunity to provide a new title to replace the title on line 1 of the file. In batch mode, the line in the CMD file following the command SAVE must have the name of the file in which the parameters are to be saved, and the next line must contain the title. If the file already exists, it will be overwritten automatically.

SEEK, arg1

SEEK causes a Monte Carlo search for the minimum of FCN. The argument is the number of FCN calls to be made during the search. The default value is 10 NPAR, where NPAR is the number of variable parameters. For each FCN call from SEEK, all variable parameters are chosen randomly according to uniform distributions centered at the best

parameter values with widths equal to the previous step sizes. At the end of SEEK, the parameters are those corresponding to the lowest value of FCN.

SEND PROGRESS

SEND PROGRESS can be executed at any time. It interrupts the program, and, if one of the fitting routines (SIMPLEX, MIGRAD, OR SEEK) is in progress, it writes on the terminal the time of day, FCN value, number of calls to FCN, and the current value and percent change of each parameter. Control is then returned to the program instruction following the last one executed before the interrupt. It is useful for checking on the progress of a fit which may otherwise go on for hours before convergence or maximum calls to FCN is reached. (SEND PROGRESS may be abbreviated SEND P).

SEND STOP

SEND STOP may be executed at any time a fit (SIMPLEX, MIGRAD, or SEEK) or MINOS is in progress. It causes the fitting routine to terminate smoothly at the end of the current iteration. The program then asks "WHAT NEXT?" and accepts a new command from the terminal. Since the fitting routine did not converge, the parameters are probably not at optimum values, but will have values corresponding to the smallest value of FCN obtained so far.

SIMPLEX arg1, arg2

The program performs a minimization using the SIMPLEX technique, which is described in Section VI. B. The command may have two arguments, which control convergence.

arg1: NFCNMX - Maximum number of calls to FCN. The minimization will be stopped after this number of calls, even if convergence is not attained. If NFCNMX = 0, there is no maximum.

arg2: EPSI - Tolerance on the range of FCN values. Default value = 0.1 UP (see ERROR).

Convergence is attained when the function values at the NPAR + 1 simplex points differ by less than EPSI.

VELCOMP

The program computes the recoil velocity (and the slowed velocity if appropriate) from the shifted and unshifted (and less shifted) photopeak centroids in the Velocity Data File (see Section II. C). The current values of BETA (and BETA1) are replaced by the newly calculated values.

IV. HOW TO USE THE PROGRAM

The first step is to construct an energy level diagram which shows all the transitions for which you wish to include recoil-distance data. To this, add additional pseudo-levels to model the feeding pattern to each real level. Give each level an ID, which may consist of up to four letters, numbers, or symbols (except blank, comma, semi-colon, parentheses, <, >, =). Figure 1 shows a sample level scheme. Levels L1 - L4 are real levels, and transitions from these levels are those for which recoil-distance data exists. Feeding to level L2 is modeled by parallel transitions from feeder levels F2 and F2'. Feeding to level L3 is modeled by the two-step cascade from levels F3 and F3'. Feeding to level L4 is modeled by a rotational band with a constant quadrupole moment.

The next step is to decide which of the levels' initial populations will be fixed at zero and to assign starting values to the initial populations of the remaining levels. In the sample case of Fig. 1, we might decide to fix the initial populations of levels L1 - L4 and B1 - B3 at zero and to assign 5 units to level F2, F2', F3, and F3', and 20 units to level B4. We must also assign initial values to the transition rates for each of the transitions, and these should all be different. We might assign $0.01 ps^{-1}$ to the transition L2 → L1, $0.10\ ps^{-1}$ to L3 → L2, $0.18\ ps^{-1}$ to L3 → L1, $0.20\ ps^{-1}$ to L4 → L3, $0.25\ ps^{-1}$ to F2 → L2, $0.35\ ps^{-1}$ to F2' → L2, $0.50\ ps^{-1}$ to F3 → L3, and $1.0\ ps^{-1}$ to F3' → F3. We may choose a starting value of 2e·b for the quadrupole moment of the rotational band.

ORNL-DWG 85-7224

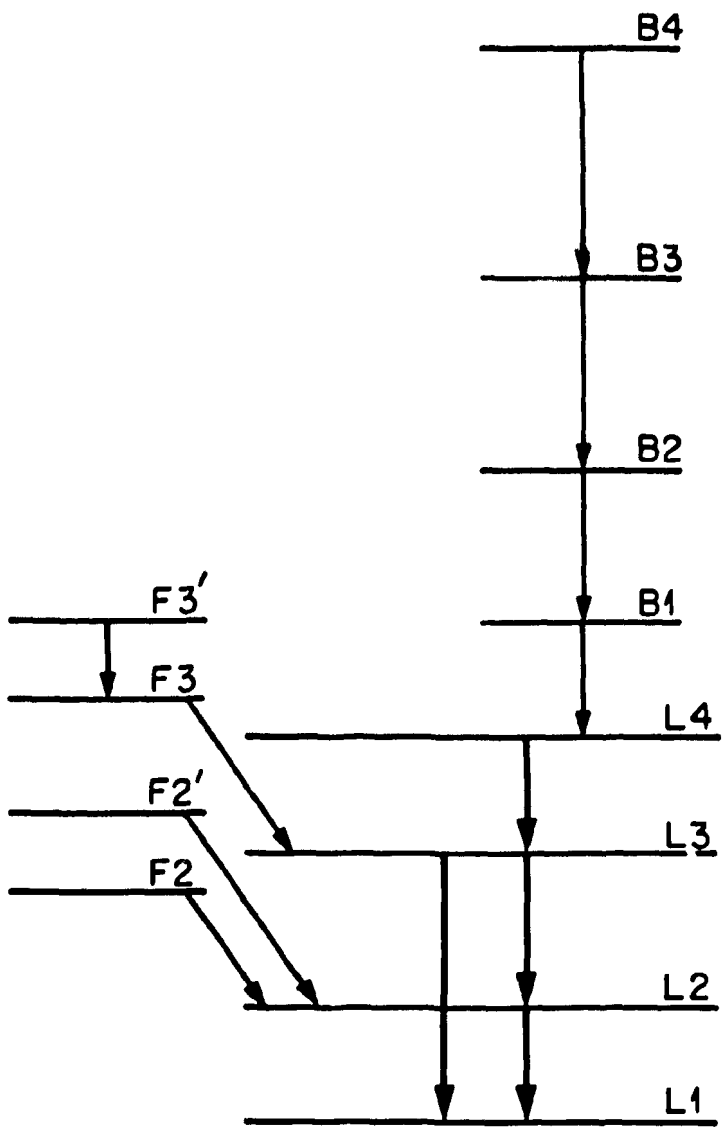


Fig. 1. Level Scheme for the sample case.

These starting values are entered in the Parameter File (see Sec. II. B) which would look like this:

```
CASE 1:  INITIAL ESTIMATES
POP        (F2)          5.0       1.0        0.0        100.0
POP        (F2')         5.0       1.0        0.0        100.0
POP        (F3)          5.0       1.0        0.0        100.0
POP        (F3')         5.0       1.0        0.0        100.0
POP        (B4)         20.0       1.0        0.0        100.0
TRANS      (L2,L1)       0.01      0.005      0.0        500.0
TRANS      (L3,L2)       0.10      0.05       0.0        500.0
TRANS      (L3,L1)       0.18      0.05       0.0        500.0
TRANS      (L4,L3)       0.20      0.05       0.0        500.0
TRANS      (F2,L2)       0.25      0.05       0.0        500.0
TRANS      (F2',L2)      0.35      0.05       0.0        500.0
TRANS      (F3,L3)       0.50      0.10       0.0        500.0
TRANS      (F3',F3)      1.00      0.20       0.0        500.0
QUAD       (L4)          2.0       0.2
DZERO       1            0.0       0.1       -1.0          1.0
NORM        2            1.0       0.1
END
```

For each parameter, the first letter in the key word identifies the parameter. For initial populations (POP), the next symbols are the level ID, and the numbers are the value of the parameter, the step size with which that parameter is varied initially, the lower limit, and the upper limit. The parameter is required to remain between these limits. For the transition rates (TRANS), the entries give the initial and final level ID's, the value, the step size and the lower and upper limits. For the rotational band quadrupole moment (QUAD), the symbols identify the level ID of the level being fed, followed by the value of the quadrupole moment and the step size. If, as in this case, both the lower and upper limits are zero or blank, then the parameter has, by default, no limits. If the step size is zero, or if the step size and both limits are blank, then the parameter is fixed at the starting value throughout the minimimization. The parameter

DZERO is a distance offset in μm subtracted from each flight distance to correct for the possibility that the measured zero distance is not the true zero. The parameter NORM is the normalization factor for data set 2 to make it compatible with data set 1, which has a normalization factor of 1.0 by default.

One should then construct the Data File (see Sec. II. A). For our example (Fig. 1), assuming two experiments, it might look like this:

```
NUCLEUS      70, 160.0
LEVEL        L1, L2, L3, L4, F2, F2', F3, F3', B1, B2, B3, B4
BAND         (L4), (B1), 6.0, 4, 500, 600, 700, 800
BRANCH       (L3), (L1), 0.6, 0.1, (L2), 0.4, 0.1

EXPERIMENT   1
DETECTOR     5.0, 2.0, 1.2, 0.2
DISTANCE     10.0, 20.0, 50.0, 100.0
VELOCITY     0.02, 0.002, 82, 208.0
TRANSITION   (L2,L1), 300.0, 1.0, 0.0, 3
MULTIPLET    (L4,L3), 4.0, 2.0
INTENSITY    0.09, 1.00, 33.98, 1.00
INTENSITY    0.27, 1.00, 33.44, 1.00
INTENSITY    1.81, 1.00, 30.81, 1.00
INTENSITY    7.11, 1.00, 24.46, 1.00
TRANSITION   (L3,L2), 400.0, 2.0, 1.0, 3
INTENSITY    0.09, 1.00, 6.48, 1.00
INTENSITY    0.40, 1.00, 6.21, 1.00
INTENSITY    1.80, 1.00, 5.12, 1.00
INTENSITY    3.69, 1.00, 3.63, 1.00
TRANSITION   (L3,L1), 700.0, 2.0, 0.0, 3
INTENSITY    0.23, 1.00, 16.34, 1.00
INTENSITY    1.00, 1.00, 15.34, 1.00
INTENSITY    4.35, 1.00, 11.91, 1.00
INTENSITY    8.55, 1.00,  7.73, 1.00

EXPERIMENT   2
DISTANCE     15.0, 30.0, 60.0
VELOCITY     0.06, 0.006, 79, 197.0, 0.0, 4.5, 0.04, 0.004
TRANSITION   (L3,L2), 400.0, 2.0, 1.0, 3
INTENSITY    0.02, 1.00, 8.03, 1.00
INTENSITY    0.10, 1.00, 7.93, 1.00
INTENSITY    0.43, 1.00, 7.55, 1.00
END
```

In this file, the first line identifies the recoil nucleus as ^{160}Yb. Line 2 gives a label for each level. Line 3 indicates that there is a rotational band feeding level L4, that the first level of the band is B1, whose spin is 6, and that there are 4 levels in the band with transition energies 500, 600, 700, and 800 keV. Line 4 gives the branching ratios from level L3 to level L1 (0.6 ± 0.1) and to level L2 (0.4 ± 0.1). The OPTIONS and ALIGNMENT lines are omitted because the default values are desired. Blank lines are ignored. Line 5 designates the data following as experiment or data set 1. Line 6 gives the target-detector distance and detector dimensions in cm. Line 7 gives the flight distances in μm. Line 8 gives the recoil velocity (v/c) and error, identifies the stopper material as ^{208}Pb, and specifies that the recoils are stopped completely. Line 9 gives the initial and final level ID's, transition energy, and initial and final spins for the first transition, and specifies that both the shifted and unshifted data be included in the fit. Line 10 indicates that this transition is a doublet, and gives initial and final level ID's and spins for the second transition of the doublet. Lines 11-14 give shifted and unshifted intensities and errors for each of the four flight distances. Lines 15-24 give the same information for the second and third transitions.

Lines 25-31 give similar information for the second experiment. Line 27 indicates that the recoil velocity is 0.06 ± 0.006, the slowing foil material is ^{197}Au, the slowing foil has a thickness of 4.5

mg/cm^2, and the slowed velocity is 0.04 ± 0.004. The DETECTOR line is omitted here because the target-detector distance and detector dimensions in this example are assumed to be the same as in experiment 1.

If one wishes to use the shifted and unshifted centroid data to compute the recoil velocity with solid-angle corrections, a Velocity Data File (see Sec. II. C) may be constructed. For our example, it might look like this:

```
EXPERIMENT  1, 0, 0.5, 0
TRANSITION  (L3,L2)
CENTROID    816, 0.1, 800, 0.1
CENTROID    816, 0.1, 800, 0.1
CENTROID    816, 0.1, 800, 0.1
CENTROID    816, 0.1, 800, 0.1
TRANSITION  (L3,L1)
CENTROID    1428, 0.2, 1400, 0.2
CENTROID    1428, 0.2, 1400, 0.2
CENTROID    1428, 0.2, 1400, 0.2
CENTROID    1428, 0.2, 1400, 0.2

EXPERIMENT  2, 0, 0.5, 0
TRANSITION  (L3,L2)
CENTROID    848, 0.1, 800, 0.1, 832, 0.1
CENTROID    848, 0.1, 800, 0.1, 832, 0.1
CENTROID    848, 0.1, 800, 0.1, 832, 0.1

END
```

The final step is to run the program. Suppose the Parameter File is called CASE1.P01, the Data File CASE1.D01, and the Velocity Data File CASE1.V01. At the terminal, enter

```
LIFE CASE1,P01,D01,V01
```

and wait for the terminal to respond,

```
WHAT NEXT?
```

Then enter a series of commands, one at a time, each time WHAT NEXT? appears. A typical series of commands might be:

```
SEEK
CORRECT
SIMPLEX
VELCOMP
CORRECT
MIGRAD
CORRECT
SAVE
END
```

This series of commands tells the program to perform a SEEK (Monte Carlo) minimization, calculate the corrections to the data, perform a SIMPLEX minimization, calculate the corrected velocity, correct the data again, perform a MIGRAD minimization, correct the data again, save the final values of the parameters, and end with a full printout. (The command SAVE must be followed by answers to questions asked by the terminal after command is entered, specifically, the name of the file in which to save the parameters and whether a new title is desired.)

We have used the program LIFETIME to analyze several sets of recoil-distance data at ORNL (See Ref. 3), and suggest the following procedure. First, divide the levels into reasonable groups, e.g. each isolated band and the region where the bands come together. Use two-

step cascade feeding into each level for which the observed intensity in is less than the observed intensity out. Estimate initial populations from the intensity balance, dividing it between the two levels in the feeder cascade. Estimate transition rates from other nearby nuclei when available. Set lower limits on initial populations and transition rates to zero, and upper limits at a physically realistic point. Use ICOR = 1 and call CORRECT after each fit. Taking one group at a time, call SEEK and SIMPLEX with all parameters variable. Then, starting with the highest level in the group and working down, call SIMPLEX and MIGRAD first varying only the parameters associated with this level, then varying the parameters associated with this level and all above it. The last step calls SIMPLEX and MIGRAD with all parameters variable, and, without calling CORRECT, call MINOS to get uncertainties in the lifetimes which include correlations with the other parameters.

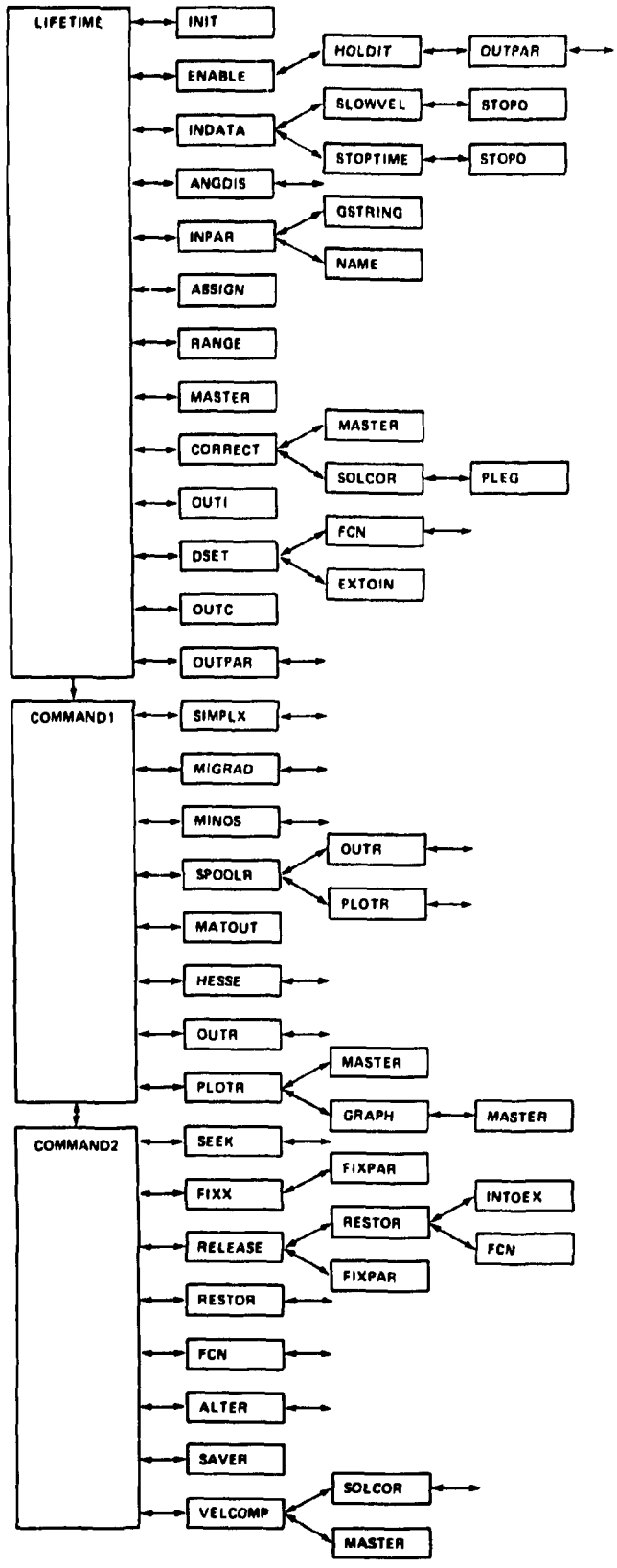
It is often the case that data for lower levels have smaller uncertainties than those for higher levels and, when all parameters are variable, the effect of the lower levels may shift the transition rates of the higher levels. This may be noticed during the fitting as one comes down the cascade in step two. A suggested procedure in this case is to start at the top and fit three levels at a time, adopting the lifetime of the middle level as the best value. (E.g., vary parameters associated with levels 10, 9, and 8 to get lifetime of level 9, vary parameters of levels 9, 8, and 7 with 10 fixed to get lifetime of level 8, etc.)

V. ORGANIZATION OF THE PROGRAM

The program is in two files, LIFETIME.FTN and MINUIT.FTN. LIFETIME.FTN contains the main program and the physics calculations. MINUIT.FTN contains the minimization and error-analysis subroutines.

The main program LIFETIME reads the data file and parameter file by calling INDATA and INPAR, performs necessary initializations, performs corrections on the data if ICOR = 1, prints out initial information, and transfers control to COMMAND1. COMMAND1, along with COMMAND2, decodes and executes each command entered from the terminal. The value of χ^2 is calculated by ASSIGN and FCN each time it is needed. Corrections to the data are performed by CORRECT every time FCN is called if ICOR = 2, or else initially and following the commands CORRECT and ALTER if ICOR = 1. Values of shifted and unshifted intensities are calculated by MASTER, in which the time-independent coefficients are calculated by COEFF, and the time-dependent intensities by DECAY. A flow chart is shown in Fig. 2.

ORNL-DWG 85-7225



ORNL-DWG 85-7226

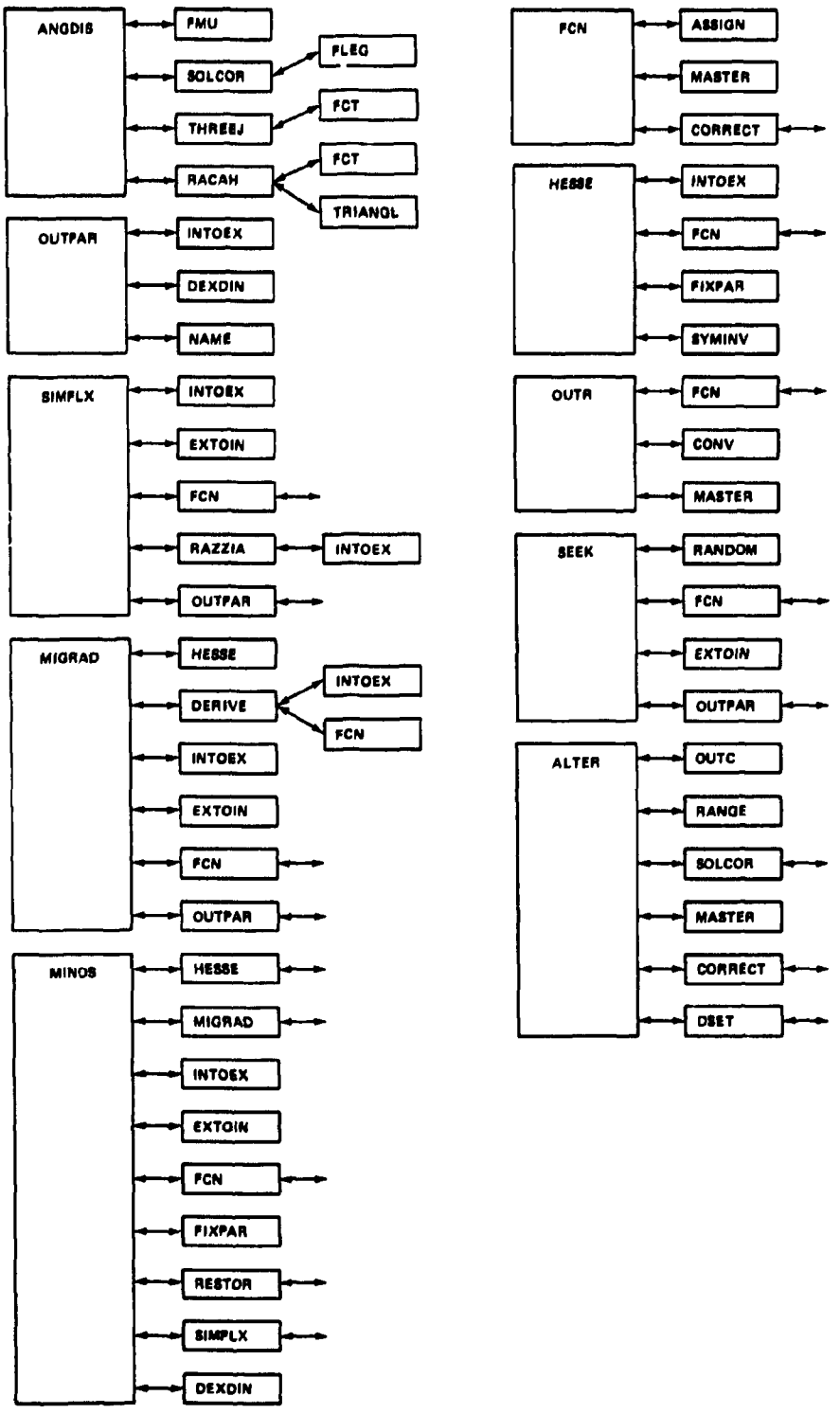


Fig. 2. Flow chart for the program LIFETIME.

Although MINUIT has been adapted to some extent to make it more interactive, it is basically as described in Ref. 2, to which the reader is referred for more detail. We include here some of the description of MINUIT from this reference.

MINUIT incorporates three minimization routines: (1) SEEK - a Monte Carlo searching subroutine; (2) SIMPLEX - a minimizing subroutine using the simplex method of Nelder and Mead,[4] which is "safe" and reasonably fast when far from minimum and which does not compute the covariance matrix, giving instead order-of-magnitude estimates of its diagonal elements (parameter errors); (3) MIGRAD - a minimizing subroutine based on a variable-metric method by Fletcher,[5] which is extremely fast near a minimum or in any "nearly quadratic" region but slower if the function is badly behaved.

The value of any variable parameter can be forced to stay within specified limits. This is done by creating two parameter sets, called internal and external. These are related by

$$U = A + (\sin X + 1)(B-A)/2,$$

where X is an internal parameter, U is the corresponding external parameter, and A and B are the lower and upper bounds on U, respectively. It is seen that although X may vary from $-\infty$ to $+\infty$, U will remain bounded by A and B. The minimization routines see only the internal parameters, while the external values are used to compute FCN, which yields χ^2. If an external parameter is unbounded, then U = X. Since X is unbounded, it is possible that its value may exceed

$(16)^{14}$, for which sin X cannot be calculated and is set to zero. This has the effect of fixing U at (A+B)/2. If a message on the terminal indicates that this has occurred, the user should use SEND STOP to terminate the routine (SIMPLEX or MIGRAD). This automatically puts all internal parameters back between $-\pi/2$ and $\pi/2$. The user may then call the routine again.

An initially variable parameter may be fixed, in which case it is removed from the internal parameter list but remains in the external list and is included in the calculation of FCN.

Parameter errors are determined from the diagonal elements of the covariance matrix. This is the inverse of the matrix whose elements are second derivatives of FCN with respect to the parameters. This covariance matrix is updated by MIGRAD as a by-product of the minimization process. An exact confidence interval for any parameter may be calculated by MINOS. This is the interval over which the function FCN takes on values less than FMIN + UP, where FMIN is the minimum of FCN and UP is a positive constant. The default value of UP is 1.0, which, for a chi-square function, determines a confidence level of 68%.

The covariance matrix V(I,J) may be approximate with varying degrees of accuracy during the calculations. A covariance matrix code, LVCODE, is used to indicate the degree of accuracy as follows:

LVCODE = 0: The covariance matrix does not exist.

LVCODE = 1: Only the diagonal elements are estimated. This is done by SIMPLEX if a better matrix does not exist, and by HESSE when its full calculation fails.

LVCODE = 2: The calculation by HESSE (inverting the second-derivative matrix) was successful, or MIGRAD has completed at least one successful iteration.

LVCODE = 3: MIGRAD has converged.

One should use the parabolic errors on the parameters with caution unless (a) MIGRAD has converged after a number of successful iterations or (b) HESSE has successfully calculated the full covariance matrix.

VI. SUBROUTINES AND FUNCTIONS

A. LIFETIME.FTN

Program LIFETIME

This is the main program. In the Perkin-Elmer version, it sets up the interrupt routine to handle SEND STOP and SEND PROGRESS by calling INIT and ENABLE. It then reads the Data and Parameter Files with INDATA and INPAR. Necessary initializations are performed by these subroutines and by ANGDIS, ASSIGN, RANGE, DSET, COEFF, and CORRECT. Initial output is provided by OUTI, OUTC, and OUTPAR. Control is then transferred to COMMAND1 which, along with COMMAND2, implements each of the available commands.

Subroutine ALTER

This subroutine allows the user to make changes in (1) the correction options, (2) the data curves being used in the fit, (3) the branching ratios being used in the fit, and (4) the range of levels used in the calculations without terminating the program and making the changes in the input files.

For each of these, the current status is displayed on the terminal, and the user is instructed how to make the changes. Then, if the changes require it, the data curves are reset to uncorrected values, the reference solid angles are recalculated, the corrections are recalculated, and the value of minimum FCN is recalculated. This subroutine is called by COMMAND2 following the command ALTER. ALTER2 is the entry point for the command CORRECT.

Subroutine ANGDIS .

This subroutine calculates the angular distribution coefficients A_2 and A_4 for a nucleus assumed to be completely aligned in a state with $J_0 = 50$, and which then deexcites by stretched-E2 transitions to state J = AJI and makes an observed transition from J = AJI to J = AJF.

In the formalism of Yamazaki,[6] for a stretched-E2 transition,

$$A_k(J_i,J_f) = \rho_k(J_i)\ F_k(J_i,J_f,2,2)$$

where the statistical tensor ρ_k is obtained from that of the preceding state by

$$\rho_k(J_{n-2}) = \rho_k(J_n)(-1)^{2J_n}((2J_n+1)(2J_n-3))^{1/2}\ W(J_n\ J_n\ J_n-2\ J_n-2;\ k\ 2).$$

For complete alignment,

$$\rho_k(J_0) = \begin{cases} (2J_0+1)^{1/2}(-1)^{J_0}\ (J_0\ 0\ J_0\ 0 \mid k\ 0) & (J_0 \text{ integral}) \\ (2J_0+1)^{1/2}(-1)^{J_0-1/2}\ (J_0\ 1/2\ J_0\ -1/2 \mid k\ 0) & (J_0 \text{ half-integral}) \end{cases}$$

In the above,

$$F_k(J_i,J_f,L_1,L_2) = (-1)^{J_f-J_i-1}((2L_1+1)(2L_2+1)(2J_i+1))^{1/2}$$

$$X\ (L_1\ 1\ L_2\ -1 \mid K\ 0)\ W(J_i\ J_i\ L_1\ L_2;\ k\ J_f)$$

Here, (a b c d | e f) is a Clebsch-Gordan coefficient, and W is a Racah coefficient. For $J_0 = 50$, one obtains $A_2 = 0.368$ and $A_4 = -0.112$, independent of the spin of the final state. (In the limit as $J_0 \to \infty$, $A_2 = 0.357$ and $A_4 = -0.101$.) If the spin change, AJI - AJF, is less

than 2, ANGDIS assumes that the multipolarity is pure E1, multiplies A_2 by $F_2(AJI,AJF,1,1)/F_2(AJI,AJI-2,2,2)$, and sets A_4 equal to zero. ANGDIS also includes the solid-angle correction factors in A_2 and A_4, which are calculated by the function SOLCOR.

ANGDIS calculates reference values for the solid-angle corrections and the alignment-attenuation corrections. The reference solid angle is chosen to be that of a nucleus at rest at the position of the target. The reference angular distribution is chosen to be that of a stretched-E2 transition from a nucleus with $A_2 = 0.368$ and $A_4 = -0.112$:

$$W_0 = 1 + 0.368P_2(1) - 0.112P_4(1) = 1.256.$$

This subroutine is called once by LIFETIME.

Subroutine ASSIGN

This subroutine assigns to the physical parameters (POP, FAM, DZERO, ANORM, QUAD) the values of the external parameters (U) in MINUIT. This is simply a convenient piece of bookkeeping. The assignment is as follows:

MP1 > 0, MP2 = 0,	POP(MP1) = U
MP1 > MP2 > 0,	FAM(MP1,MP2) = U
MP1 = 0, MP2 > 0,	DZERO(MP2) = U
MP1 = -1, MP2 > 0,	ANORM(MP2) = U
MP1 > 0, MP2 = -1,	QUAD(MP1) = U.

In this last case, QUAD is used to evaluate the transition probabilities of the transitions in the rotational band, where, with E_γ in keV, Q_0 in e•b, and λ_{nm} in ps^{-1},

$$\lambda_{nm} = 1.2253\times10^{-15}\, E_\gamma^5\, Q_0^2\, (J_n\ 2\ 0\ 0 \mid J_m\ 0)^2$$

$$= 1.8379\times10^{-15}\, E_\gamma^5\, Q_0^2\, J_n(J_n - 1)/(2J_n + 1)(2J_n - 1).$$

This subroutine is called immediately after the parameters are read in by INPAR, and by FCN each time it is executed.

Subroutine CONV(E,L,Z)

This subroutine calculates internal conversion coefficients by interpolating or extrapolating from values given Roesel et al.[7]

The table of values includes E1, M1, E2, M2, E3, and E4 conversion coefficients for E > 2.2 keV and for Z = 58, 60, 62, 66, 70, 77, 79, and 90. Other values of Z may be added as desired. This subroutine is called by OUTR to calculate B(E2) values.

Subroutine CORRECT

This subroutine calculates corrections to the data for changes in solid angle due to the changing position of the recoil nucleus along the flight path, for the relativistic velocity transformation of the solid angle, for changes in the angular distribution due to attenuation of alignment in flight, and corrections for the gamma-ray intensity between the shifted and unshifted peaks emitted by recoils slowing down in the stopper. (See discussion in Refs. 8 and 9.)

The solid angle subtended by the detector when the recoil is at some point along the flight path is given by

$$\Delta\Omega = \int \sin\theta\, d\phi\, d\theta\, dr,$$

where the integration is performed over the volume of the detector.

If the recoil is moving with velocity v, then the angle θ must be transformed from a moving to a rest frame of reference by the transformation

$$\cos\theta = (\cos\theta' + \frac{v}{c})/(1 + \frac{v}{c}\cos\theta').$$

The velocity transformation and the integral over the detector volume are calculated by the function SOLCOR.

The solid angle for the shifted component must be averaged along the flight path,

$$\langle\Omega_S\rangle = \int_0^{t_1} \Omega(t)\lambda p(t)dt / \int_0^{t_1} \lambda p(t)dt,$$

where $t_1 = d/v$ is the flight time, λ is the decay rate and $p(t)$ is the population at time t calculated by subroutine MASTER.

The angular distribution of radiation from aligned nuclei can be written

$$W(t) = 1 + A_2(t)P_2(\cos\theta) + A_4(t)P_4(\cos\theta),$$

where A_2 and A_4 are the angular distribution coefficients and P_2 and P_4 are Legendre polynomials.

It is assumed that the attenuation can be described by an exponential model, following the frequently-used treatment by Abragam and Pound,[10]

$$A_2 = A_{20}\, e^{-t/\tau_2}$$

and

$$A_4 = A_{40}\, e^{-t/\tau_4},$$

where A_{20} and A_{40} are the coefficients assuming complete alignment at state J = 50 and no attenuation. These are calculated by subroutine ANGDIS.

The intensity for the shifted component must be averaged along the flight path,

$$\langle W_S \rangle = \int_0^{t_1} W(t)\lambda p(t)dt / \int_0^{t_1} \lambda p(t)dt.$$

These three integrals are performed by Simpson's Rule over the flight time using NPTS integration points. The default value of 11 usually gives sufficient accuracy, but this may be changed at the user's option. If a data curve is a multiplet, the corrections are averaged over the transitions in the multiplet.

For the unshifted component, the solid angle is simply calculated at the position of the stopper for a recoil at rest, and the alignment attenuation is assumed to stop when the recoil enters the lattice of the stopper, so that

$$\langle \Omega_U \rangle = \Omega(t_1)$$

and

$$\langle W_U \rangle = W(t_1).$$

Again, the corrections are averaged over the transitions in a multiplet.

If the experiment involves slowing the recoil rather than stopping it completely, then the solid angle and intensity of the less shifted component must be averaged along the flight path after passing through the slowing foil,

$$\langle \Omega_1 \rangle = \int_{t_1}^{t_2} \Omega(t)\lambda p(t)dt / \int_{t_1}^{t_2} \lambda p(t)dt$$

and

$$\langle W_1 \rangle = \int_{t_1}^{t_2} W(t)\lambda p(t)dt / \int_{t_1}^{t_2} \lambda p(t)dt,$$

where $t_1 = d/v$ and t_2 is a time sufficiently long that all activity will essentially have ceased, but not so long that every point in the numerical integration after the first would be zero. The value chosen for t_2 is seven times the largest of either the lifetime of the transition itself or the lifetime of any transition feeding it.

Finally, each of the data points is multiplied by the factor $\Omega_0 W_0 / \langle \Omega_k \rangle \langle W_k \rangle$, where k represents s for shifted data, u for unshifted data and 1 for less-shifted data, and where Ω_0 is the reference solid angle subtended by the detector for a nucleus at rest at the position of the target, and $W_0 = 1.256$ is the reference angular distribution. Both Ω_0 and W_0 are calculated by ANGDIS.

A correction is made to include the gamma-ray intensity emitted while the recoils are slowing down in the stopper. This intensity would lie between the shifted and unshifted peaks, or between the shifted and less shifted peaks, as shown in Fig. 3. The peak-fitting program takes a background above the region I rather than the true background under I. Since the flight distance, and therefore the flight time t_1, is measured to the front surface of the stopper, counts in I must be added to U. This is done by calculating the stopping time t_s with the function STOPTIME, and then multiplying the

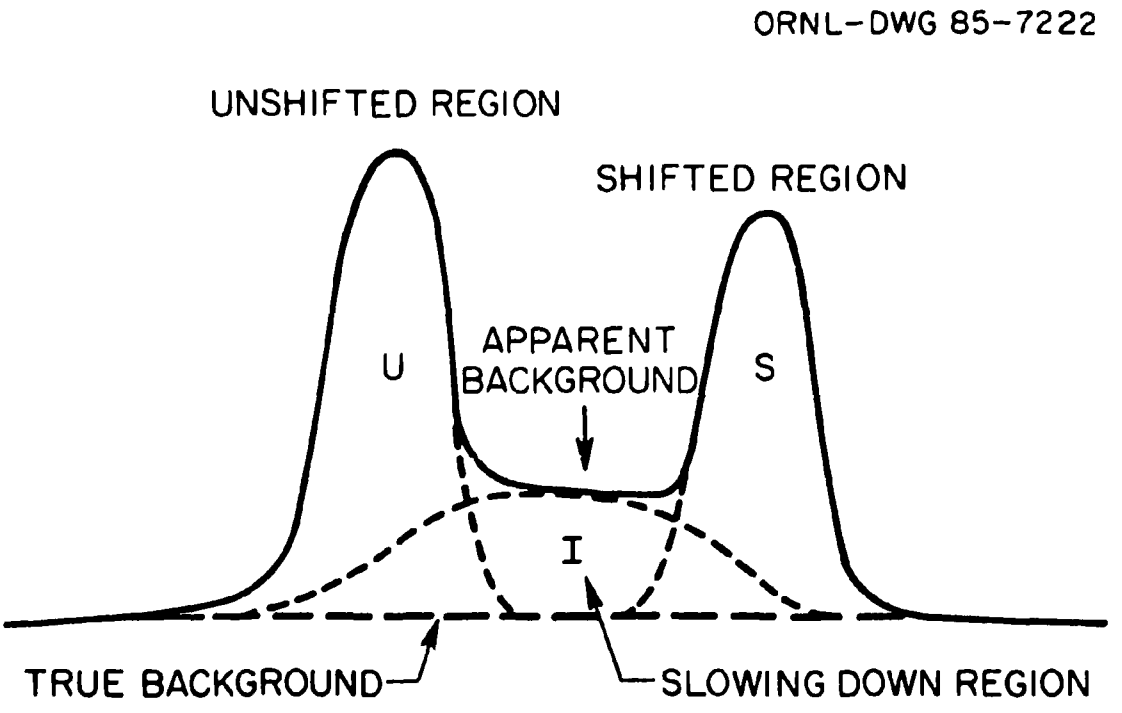


Fig. 3. Schematic spectrum showing typical shifted and unshifted photopeaks, and the region between them due to γ rays emitted while the recoil nucleus is slowing down. This intermediate intensity is included in the unshifted intensity.

unshifted intensity data by the ratio $U(t_1)/U(t_1 + t_s)$, where $U(t)$ is the calculated unshifted intensity for a flight time t.

Subroutine FCN(F)

This subroutine calculates the chi-square function:

$$\chi^2 = \sum_{ijk} (U_{ij}^{exp}(t_k) - U_{ij}^{cal}(t_k))^2/(\Delta U_{ij}(t_k))^2 + (S_{ij}^{exp}(t_k) - S_{ij}^{cal}(t_k))^2/(\Delta S_{ij}(t_k))^2,$$

where U_{ij} and S_{ij} are the unshifted and shifted intensities between level i and level j, ΔU_{ij} and ΔS_{ij} are the corresponding uncertainties, and $t_k = d_k/v$ is the flight time.

If a γ ray is known to be multiple, then the calculated intensities for all members of the multiplet are summed before computing χ^2.

If branching-ratio data are included, the following terms are added to χ^2:

$$\sum_{ij} (B_{ij}^{exp} - B_{ij}^{cal})^2/(\Delta B_{ij})^2,$$

where

$$B_{ij}^{cal} = \lambda_{ij}/(\sum_{j'} \lambda_{ij'})$$

and λ_{ij} is the fitted transition rate from level i to level j.

This subroutine calls CORRECT if ICOR = 2.

Function FCT(N)

This function calculates the value of

$$FCT = (N - 1)!/10^{N-1}.$$

It is called by THREEJ and RACAH.

Function FMU(E)

This function calculates the linear absorption coefficient in μm^{-1} of a γ ray of energy E (in keV) in germanium. It performs an interpolation between entries in a table using a cubic polynomial (Aitkin's method). These absorption coefficients are used by function SOLCOR.

Subroutine GRAPH(XHI,IP,NP,LSET,TIME,IPLT,IUNIT)

This subroutine produces a line printer plot showing shifted and unshifted intensity data and fitted decay curves. It is called by PLOT.

Subroutine HOLDIT(MESSAGE) (Perkin-Elmer version only.)

This is the subroutine to which control is transferred by ENABLE when the asynchronous trap command SEND "message" is executed. If "message" is STOP, the fitting procedure is terminated at the end of the current iteration. Otherwise, OUTPAR is called to display FCN, NFCN, and parameter values and changes on the terminal. Control is then transferred to the instruction following the one being executed when the interrupt occurred.

Subroutine INDATA

This subroutine reads the Data File. It initializes certain arrays and variables, checks that data arrays do not exceed dimensions, and converts all data units to μm for length, ps for time, and keV for energy. It calls the function SLOWVEL to calculate the slowed velocity from the target thickness if the experiment involves a slowing foil and this velocity is not supplied. It calls the function

STOPTIME to calculate the stopping or slowing-down time of recoils in the foil. This quantity is needed to make the lineshape correction. INDATA is called by LIFETIME.

Function KINDA(CHAR,KODE)

This function identifies a character type as a numeric, connector, delimiter, carriage return, or other. If KODE=1, a dash (-) is identified as a minus sign. If KODE=2, a dash is identified as a connector between a range of numbers (e.g., 1-10). The delimiters recognized are: blank, comma, semi-colon, (,), <, >, =. It is called by UNPACK.

Subroutine MASTER

This subroutine uses the Bateman equations to calculate the population of each level at a given time in terms of the initial populations of the levels and the transition rates. It also calculates the total number of decays before and after the given time. These are the shifted and unshifted intensities. The calculation proceeds as follows.

Assume a set of N levels with initial populations $P_i(0)$, and assume that transitions only occur from higher-index to lower-index levels, so that transition rates $\lambda_{ij} > 0$ if $i > j$ and $\lambda_{ij} = 0$ if $i < j$. Let Γ_i be the total transition rate out of level i, i.e.,

$$\Gamma_i = \sum_{j=1}^{i-1} \lambda_{ij}.$$

An additional assumption required to simplify the solution of the differential equations is that $\Gamma_i \neq \Gamma_j$ if $i \neq j$. The differential equations describing the decay process can be written:

$$\frac{d}{dt} P_i(t) = \sum_{j=i+1}^{N} P_j(t)\lambda_{ji} - P_i(t)\Gamma_i.$$

This system of master equations has the solution

$$P_i(t) = \sum_{j=i+1}^{N} a_{ij} P_j(t) + b_i \exp[-\Gamma_i t],$$

where, for $i < j$,

$$a_{ij} = \begin{cases} \lambda_{ji}/(\Gamma_i-\Gamma_j) & (j = i + 1) \\ (\lambda_{ji} - \sum_{k=i+1}^{j-1} \lambda_{jk} a_{ik})/(\Gamma_i-\Gamma_j) & (j > i + 1). \end{cases}$$

Also,

$$b_i = P_i(0) - \sum_{j=i+1}^{N} a_{ij} P_j(0).$$

If we define

$$w_i(t_1) = \int_0^{t_1} P_i(t)dt = \sum_{j=i+1}^{N} a_{ij} w_j(t_1) + \frac{b_i}{\Gamma_i} (1-\exp[-\Gamma_i t_1]),$$

then the total shifted intensity at $t = t_1$ for the transition $i \to j$ is given by

$$s_{ij}(t_1) = \lambda_{ij} w_i(t_1) \qquad (j < i).$$

Similarly, if

$$v_i(t_1) = \int_{t_1}^{\infty} P_i(t)dt = \sum_{j=i+1}^{N} a_{ij} v_j(t_1) + \frac{b_i}{\Gamma_i} \exp[-\Gamma_i t_1],$$

then the total unshifted intensity at $t = t_1$ is given by

$$u_{ij}(t_1) = \lambda_{ij}\, v_i(t_1) \qquad (j < i).$$

The total intensity I_i of all transitions from level i is given by

$$I_i = \begin{cases} P_N(0) & (i = N) \\ P_i(0) + \sum_{j=i+1}^{N} \lambda_{ji} I_j / \Gamma_j & (i < N), \end{cases}$$

or by

$$I_i = \Gamma_i w_i(\infty).$$

MASTER is written in two independent parts. Entry COEFF calculates the coefficients a_{ij} and b_i and the total intensity I_i. These are all independent of time. Entry DECAY calculates $P_i(t_1)$, $S_{ij}(t_1)$, and $U_{ij}(t_1)$. So COEFF is called once, then DECAY is called for each flight time in turn.

To save time, the calculations are made only for levels in the range LMIN - LMAX. Also, if the level population $P_i(t_1)$ is all that is needed, the calculation of $S_{ij}(t_1)$ and $U_{ij}(t_1)$ is omitted by setting the logical variable JPOP = .TRUE.; otherwise JPOP = .FALSE. causes the shifted and unshifted intensities to be calculated.

Function NAME(IPAR)

This function composes a character name for each parameter using the level identification names for the levels that the parameter represents.

Subroutine OUTC(IUNIT)

This subroutine produces on the terminal or line printer output which gives information on those fitting conditions which can be changed by ALTER during the run. These comprise (1) the corrections being applied, (2) the decay curves being fitted, (3) the branching ratios being fitted, and (4) the range of levels. It is called by LIFETIME to record initial conditions, and by the subroutine ALTER each time the command ALTER is executed.

Subroutine OUTI

This subroutine produces on the line printer output which gives information on those input data which remain unchanged as the program runs. These consist of the charge and mass of the recoil, stopper thickness and density, recoil velocity, target-detector distance, detector dimensions, and alignment relaxation times. It is called once by LIFETIME.

Subroutine OUTR(NFP,IUNIT)

This subroutine produces on the line printer output which gives the final results of the fit. These include the best chi-square, the lifetimes, transition rates, B(E2)'s, transition quadrupole moments, quadrupole moments of rotational bands, zero distances, level populations as a function of time, branching ratios, and original and corrected data curves. It is called by COMMAND1 following the PRINT or END commands.

The errors in the parameters are derived from the diagonal elements of the covariance matrix. The errors in lifetimes are derived from errors in the transition rates. In addition, the relative error in the recoil velocity is added in quadrature to the relative error in the transition rates and lifetimes since, to first order, an error in the velocity would produce an error in all of the flight times in a linear way. If there is more than one experiment, only the recoil velocity error which is smallest is included.

Function PLEG(U,K)

This function calculates the value of the Legendre Polynomial $P_k(u)$. It uses the recursion relation

$$(k+1)P_{k+1}(u) = (2k+1)uP_k(u) - kP_{k-1}(u).$$

It is called by SOLCOR.

Subroutine PLOT(IPLT,IUNIT)

This subroutine produces line printer plots of the shifted and unshifted intensity data and fitted curves for all data being included in the fit. It calls subroutine GRAPH to produce each plot. It is called by COMMAND1 following the PLOT or END commands.

Function RACAH(A,B,C,D,E,F)

This function calculates the value of the Racah coefficient W(ABCD;EF). It is called by ANGDIS.

Subroutine RANGE

This subroutine examines levels above the level with LID LMAX to see if any of them have significant transition rates into levels in the range LMIN to LMAX. If any are found, it raises LMAX to include

them. It repeats the procedure until no additional feeders are found. It thus establishes that the range LMIN to LMAX contains all significant feeders to the levels of interest. It is called initially by LIFETIME and by subroutine ALTER each time the command ALTER is executed.

Subroutine SAVER

This subroutine stores the current values of the parameters in a file specified by the user and in a format such that this file can later be read as an Input Parameter File. A new title may be written if desired.

Function SLOWVEL(L)

This function calculates the velocity v_1 of recoil nuclei having an initial velocity v_0 which pass through a foil of thickness X_f. It solves numerically, using Simpson's Rule, the integral equation

$$X_f = - \int_{E_0}^{E_1} \frac{dE}{S(E)}$$

for E_1 where $E_0 = mv_o^2/2$, m is the recoil mass, and S(E) is the stopping power of the recoil in the foil obtained from the function STOPO. Then

$$v_1 = (2E_1/m)^{1/2}.$$

This function is called by INDATA if v_1 is needed and not supplied as input.

Function SOLCOR(L,D,AMU,BETA,K,IVOL)

This function calculates the integral

$$J_k(E_\gamma) = \int P_k(\cos\theta)(1 - \exp[-\mu(E_\gamma)X(\theta)])\sin\theta d\theta$$

over the volume of a coaxial Ge(Li) detector using the method described by K.S. Krane.[11] In this expression, θ is the angle a γ ray makes with the detector axis, P_k is a Legendre polynomial, $\mu(E_\gamma)$ is the γ-ray absorption coefficient for γ rays of energy E_γ in Ge, and $X(\theta)$ is the path length through the detector at angle θ.

Since $(1 - \exp[-\mu(E_\gamma)X(\theta)])$ is proportional to the γ-ray intensity absorbed along a path of length $X(\theta)$, then the integral for $k = 0$ is proportional to the absolute detctor efficiency (or the effective solid angle for complete absorption). For $k = 1$, it gives the average value of $\cos\theta$, which is needed by VELCOMP to calculate the recoil velocity. $J_2(E_\gamma)/J_0(E_\gamma)$ and $J_4(E_\gamma)/J_0(E_\gamma)$ are the solid angle correction factors for the angular distribution coefficients A_2 and A_4.

For coaxial Ge(Li) detectors, the integral is broken into three parts, as shown in Fig. 4. The limits of the regions of integration are:

$$\theta_1 = \arctan(R_I/(D+L))$$

$$\theta_2 = \arctan(R_I/D)$$

$$\theta_3 = \arctan(R_O/(D+L))$$

$$\theta_4 = \arctan(R_O/D)$$

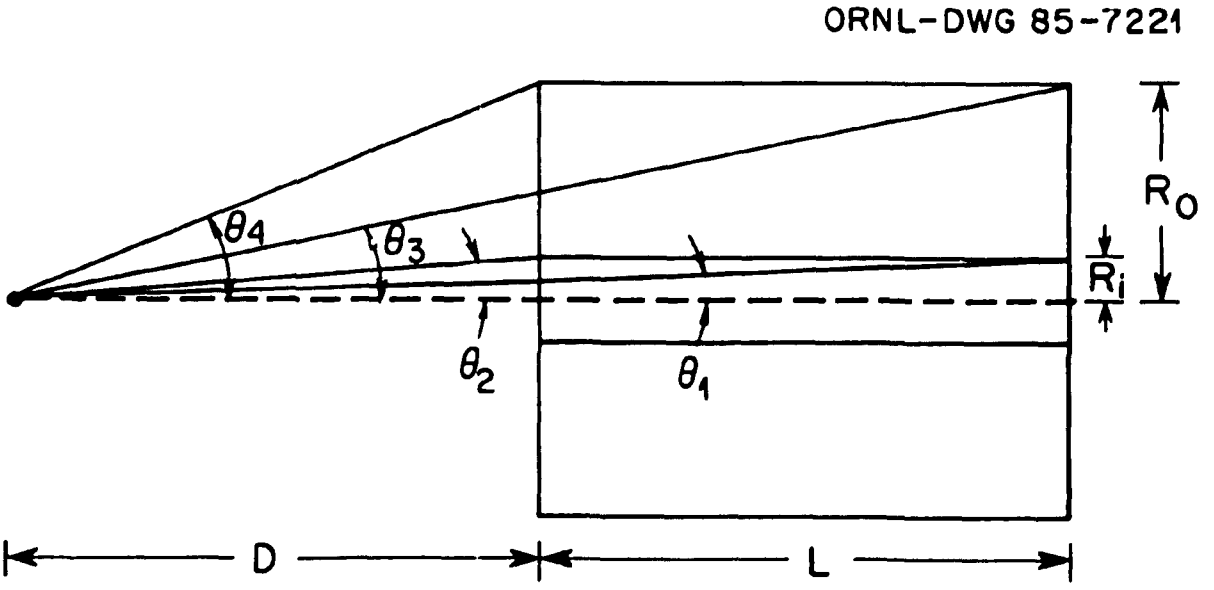


Fig. 4. Schematic cross section of a Ge(Li) detector showing the regions used for integrating over the detector volume. The active region is the annular cylinder between R_i and R_o.

If the dimensions are such that $\theta_3 > \theta_2$, then the path length is

$$X(\theta) = \begin{cases} (D+L)\sec\theta - R_1\csc\theta & (\theta_1 < \theta < \theta_2) \\ L\sec\theta & (\theta_2 < \theta < \theta_3) \\ R_0\csc\theta - D\sec\theta & (\theta_3 < \theta < \theta_4) \end{cases}$$

Otherwise, if $\theta_2 > \theta_3$, one has

$$X(\theta) = (R_0-R_1)\csc\theta \qquad (\theta_3 < \theta < \theta_2)$$

with the roles of θ_2 and θ_3 in the outer regions of integration being interchanged.

For $\theta_1 < \theta < \theta_2$, the integrand of $J_k(E_\gamma)$ is multiplied by the additional factor $\exp[-\mu(E_\gamma)X'(\theta)]$ to take into account the attenuation of the γ ray in the nonactive core of the detector. Here,

$$X'(\theta) = R_1\csc\theta - D\sec\theta.$$

If $\theta_3 < \theta_2$, this factor is also introduced for the region $\theta_3 < \theta < \theta_2$.

Since the recoil nucleus is moving with a relativistic velocity, the angles must be transformed to the moving frame of reference, so in this case the integral has the form

$$J_k(E_\gamma) = \sum_{n=1}^{3} \int_{\theta'(\theta_n)}^{\theta'(\theta_{n+1})} P_k(\cos\phi)(1 - \exp[-\mu(E_\gamma)X(\theta(\phi))])\sin\phi d\phi,$$

where the aberrant angle $\theta'(\theta)$ is given by

$$\cos\theta' = (\cos\theta - \beta)/(1 - \beta\cos\theta)$$

and conversely $\theta(\phi)$ by

$$\cos\theta = (\cos\phi + \beta)/(1 + \beta\cos\phi).$$

Thus, the limits of integration are transformed to the moving frame, in which the integration is performed, but the variable of integration

ϕ must be transformed into the lab frame to calculate the path length through the detector.

The integral is evaluated numerically in each region using Simpson's rule, in which even terms are doubled and the first and last terms are halved. To avoid unnecessary divisions, these are identified by the function BTEST(M,31) which returns .TRUE. if M is odd and .FALSE. if M is even. The number of terms in the integration is NPTG. The default value of 11 usually gives sufficient accuracy, but this may be changed at the user's option.

As a time-saving device, SOLCOR can be instructed to calculate the function J_k with the assumption that the detector is black; that is, that all γ rays are absorbed at the front surface. In this case, the integral can be evaluated analytically:

$$J_k = (P_{k-1}(\cos\theta'(\theta_4)) - P_{k+1}(\cos\theta'(\theta_4)) + P_{k+1}(\cos\theta'(\theta_2)) - P_{k-1}(\cos\theta'(\theta_2)))/(2k + 1).$$

This option is chosen if IVOL = 0, otherwise the full volume integral is computed.

The variables in this subroutine are

D = target-detector distance in μm

XL = detector length in μm

RI = radius of the inactive core in μm

RO = outer radius in μm

AMU = $\mu(E_\gamma)$ in μm^{-1}.

This subroutine is called by ANGDIS, CORRECT, and VELCOMP.

Subroutine SPOOLR(SUB,I)

This subroutine calls OUTR or PLOT and sends the results to the line printer immediately without the program terminating. It is called by subroutine COMMAND1 following execution of the PRINT or PLOT commands.

Function STOPO(ZP,AP,ZT,AT,RHO,EKEV)

This routine calculates the stopping powers of ions in all solid elements, using the parameters and formalism of J.F. Ziegler.[12,13]

The first step is to calculate the electronic stopping power $S(E_p)$ of protons[12] with the same velocity as that of the ion of interest in the stopping material. The quantity E_p is given by

$$E_p = m_p E_I/m_I,$$

where E_I is the ion energy, m_I and m_p are the masses of the ion and the proton, respectively, and E_p is the energy of the proton with the same velocity as the ion. Then, if $E_p < 10$ keV,

$$S(E_p) = A_1 E_p^{1/2};$$

if 10 keV $< E_p <$ 1000 keV,

$$S(E_p) = S_L S_H/(S_L + S_H),$$

where

$$S_L = A_2 E_p^{0.45}$$

$$S_H = (A_3/E_p) \ln[1 + (A_4/E_p) + A_5 E_p];$$

and if 1000 keV $< E_p$,

$$S(E_p) = (A_6/\beta^2)(\ln[A_7\beta^2/(1 - \beta^2)] - \beta^2 - \sum_{i=0}^{4} A_{i+8}(\ln E_p)^i).$$

The quantities A_1 to A_{12} are coefficients which depend on the specified element of the stopping material,[12] and

$$\beta^2 = v^2/c^2 = 1 - 1/(1 + E_p/938837)^2.$$

These equations give $S(E_p)$ in units of $eV/(10^{15}$ atoms/cm$^2)$.

The next step is to calculate the effective charge Z_{eff} of the projectile. For protons, $Z_{eff} = 1$. For He ions, $Z_{eff} = 2\gamma(1 - \exp[-F])$, where

$$F = 0.7446 + 0.1429(\ln E) + 0.01562(\ln E)^2 - 0.00267(\ln E)^3 + 1.325\times10^{-6}(\ln E)^8$$

and

$$\gamma = 1 + (0.007 + 0.00005\ Z_S)\exp[-(7.6 - \ln E)^2],$$

Z_S being the charge of the stopping material, and E the larger of E_p and 15 keV. For Li ions, $Z_{eff} = 3\gamma(1 - \exp[-G])$, where

$$G = 0.7138 + 0.002797\ E + 1.348\times10^{-6}\ E^2,$$

γ being the same as for He, and E the larger of E_p and 150 keV. For heavy ions,[13] $Z_{eff} = Z_I(1 - \exp[-A])(1.034 - 0.1777\ \exp[-0.08114Z_I])$, where

$$A = B + 0.0378\ \sin(\pi B/2)$$

$$B = 0.1772\ E^{1/2}Z_I^{-2/3}.$$

Here, E is the larger of E_p and the lesser of $9Z_I + 19$ and 200 keV, and Z_I is the ion charge.

The third step is to calculate the nuclear stopping power,[13]

$$S_n(E) = 4.231Z_IZ_Sm_I\ \ln(1 + \varepsilon)/(D(\varepsilon + 0.10718\varepsilon^{0.37544})),$$

where

$$\epsilon = 32.53\ m_S E_I/(Z_I Z_S D)$$
$$D = (m_I + m_S)(Z_I^{2/3} + Z_S^{2/3})^{1/2}.$$

Here, E_I is the ion energy in keV, Z_I and Z_S are the ion and stopper changes and m_I and m_S are the ion and stopper masses in u. The total stopping power is then

$$S_T(E) = Z_{eff}^2\ S(E_p) + S_n(E)$$

in units of eV/(10^{15} atoms/cm^2). This can be converted to keV/μm by multiplying by 60.22094 ρ/m_S, where ρ is the density of the stopper in g/cm^2 and m_S is the mass of the stopper in u.

Function STOPTIME(L)

This function calculates the stopping or slowing-down time t_s of the recoil nucleus in the foil by evaluating numerically, using Simpson's Rule, the integral

$$t_S = -\int_{v_0}^{v_1} \frac{m\,dv}{S(E(v))}\ ,$$

where m is the recoil mass, $E(v) = mv^2/2$, S(E) is the stopping power of the recoil in the foil obtained from the function STOPO, v_0 is the initial velocity of the recoil, and v_1 is the recoil's final velocity, or, if the recoil is completely stopped, v_1 is set equal to a small value determined by the width of the unshifted photopeak. The above integral is derived as follows:

$$S(E) = -\frac{dE}{dx}\ ,$$

and since $dx = vdt$ and $dE = d(mv^2/2) = mvdv$, one has

$$dt = \frac{mdv}{S(mv^2/2)} .$$

This function is called by INDATA if t_s = STIME is not supplied as input.

Function THREEJ(A1,B1,A2,B2,A3,B3)

This function calculates the value of the 3-j symbol

$$\begin{pmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{pmatrix} .$$

Clebsch-Gordon coefficients are related by the expression

$$(A_1\ B_1\ A_2\ B_2 \mid A_3\ -B_3) = (-1)^{(A_1-A_2-B_3)}(2A_3+1)^{1/2} \begin{pmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{pmatrix} .$$

This function is called by ANGDIS.

Function TRANGL(A,B,C)

This function returns zero if the arguments do not satisfy a triangle inequality, i.e., if any argument exceeds the sum of the other two, or if the sum of the arguments is not integral. Otherwise, it returns the value of

$$\frac{(A+B-C)!(B+C-A)!(C+A-B)!}{(A+B+C+1)!} .$$

It is called by RACAH.

Subroutine UNPACK(LINE,ARGS,IARGS,NARGS,KODE)

This subroutine reads a character string and decodes it into a series of signed floating point numbers if KODE=1, or positive integers if KODE=2, disregarding any nonnumeric characters at the beginning. It calls function KINDA to identify the type of each

character. If KODE=1, it decodes a dash (-) as a minus sign. If KODE=2, it decodes a dash as a connector between a range of numbers. For example, 1-5 would be decoded 1,2,3,4,5. It returns the values of these arguments and the number of arguments. Characters enclosed in parentheses are decoded as level ID's, and an internally-generated level number is returned for each of these. UNPACK is called by INDATA, INPAR, COMMAND2, FIXX, and RELEASE.

Subroutine VELCOMP

This subroutine calculates the recoil velocity from the shifted and unshifted photopeak centroids. It makes a correction for the fact that $\Delta E/E$ really represents the velocity averaged over the time-dependent solid angle subtended by the detector.

If one makes the assumption of cylindrical symmetry about the beam axis, then the average separation ΔE between the shifted and unshifted (or less shifted) peaks can be written in terms of the recoil velocity v as

$$\frac{\Delta E}{E} = \frac{v}{c} \frac{\int_0^{t_1} \int_0^{\theta'(t)} \cos\theta \, P_e(\theta,t) P_d(\theta,t) \sin\theta \, d\theta \, dt}{\int_0^{t_1} \int_0^{\theta'(t)} P_e(\theta,t) P_d(\theta,t) \sin\theta \, d\theta \, dt},$$

where E is the transition energy and c is the velocity of light. The quantity $P_e(\theta,t)$ is the probability of emission of a γ ray at angle θ at time t. It is assumed that the range of θ is small enough that the angular distribution can be disregarded, and so $P_e(\theta,t) = \lambda p(t)$, where λ is the transition rate and p(t) is the population at time t.

The quantity $P_d(\theta,t)$ is the probability that a γ ray emitted at angle θ at time t will be absorbed by the detector. The integration limit $\theta'(t)$ is the limit in the moving frame, obtained by the transformation

$$\cos\theta' = (\cos\theta - \frac{v}{c})/(1 - \frac{v}{c}\cos\theta).$$

Then,

$$\frac{v}{c} = \frac{\Delta E}{E} \frac{\int_0^{t_1} \lambda p(t)\left(\int_0^{\theta'(t)} P_d(\theta,t)\sin\theta\, d\theta\right)dt}{\int_0^{t_1} \lambda p(t)\left(\int_0^{\theta'(t)} \cos\theta\, P_d(\theta,t)\sin\theta\, d\theta\right)dt} = f\frac{\Delta E}{E}.$$

The integrals over θ are calculated by function SOLCOR, and those over t are calculated by VELCOMP using Simpson's Rule integration with NPTS integration points.

First, subroutine VELCOMP reads in the peak centroids and calculates energies using a quadratic calibration curve.

Next, the unshifted energies u_k for different flight times for a given transition are averaged using a weighted average based on the errors Δu_k in the energies:

$$\langle u\rangle = \sum_k (u_k/(\Delta u_k)^2)/\sum_k (1/(\Delta u_k)^2).$$

The error in this average is taken to be the larger of (1) the error computed from the individual errors and (2) the standard deviation of the unshifted energies themselves. The standard deviation is given by

$$\sigma_u^2 = (\langle u^2\rangle - \langle u\rangle^2)/(n_u - 1),$$

where

$$\langle u^2 \rangle = \sum_k (u_k^2/(\Delta u_k)^2)/\sum_k (1/(\Delta u_k)^2)$$

and n_u is the number of terms in the sum. Thus,

$$\Delta u = \mathrm{Max}[\sigma_u, 1/\sum_k (1/(\Delta u_k)^2)].$$

Then the corrected velocity v_k is calculated for each shifted energy s_k for a given transition,

$$v_k/c = f_k(s_k - \langle u \rangle)/\langle u \rangle = f_k s_k/\langle u \rangle - f_k,$$

where f_k is the ratio of integrals over θ and t appropriate to transition k, as discussed previously. These velocities are then averaged using a weighted average based on the errors Δs_k in s_k,

$$\langle v/c \rangle = \sum_k ((v_k/c)/(\Delta s_k)^2)/\sum_k (1/(\Delta s_k)^2).$$

The error is again taken to be the larger of (1) the error computed from the individual errors and (2) the standard deviation of the individual velocities themselves:

$$\Delta v/c = \mathrm{Max}[\sigma_v, D_v],$$

where

$$\sigma_v^2 = (\langle (v/c)^2 \rangle - \langle v/c \rangle^2)/(n_s - 1)$$

with

$$\langle (v/c)^2 \rangle = \sum_k ((v_k/c)^2/(\Delta s_k)^2)/\sum_k (1/(\Delta s_k)^2),$$

and

$$D_v^2 = ((1/\langle u \rangle)^2 \sum_k (f_k \Delta s_k/(\Delta s_k)^2)^2 + (\Delta u/\langle u \rangle^2)^2 (\sum_k f_k s_k/(\Delta s_k)^2)^2)/D_s^2$$

$$= ((\sum_k f_k s_k/(\Delta s_k)^2)^2 (\Delta u)^2 + \sum_k (f_k/\Delta s_k)^2 \langle u \rangle^2)/(\langle u \rangle^2 D_s)^2,$$

with

$$D_S = \sum (1/(\Delta s_k)^2).$$

Finally, a weighted average of v/c for all the transitions in the data set is calculated using the same procedure that was used for averaging the unshifted energies. Flight times are recalculated from flight distances using the new "best value" of v/c.

If the experiment involves a "slowed" velocity, it is calculated in exactly the same way, except that the limits of the integration over t are t_1 to t_2 instead of 0 to t_1, where t_1 is the flight time to the slowing foil and t_2 is seven times the longest lifetime of either the transition itself or any feeder to the transition. (See discussion on page 45.)

Subroutine VELCOMP is called by COMMAND1 following the VELCOMP command.

B. MINUIT.FTN

Subroutines COMMAND1 and COMMAND2(ISUB)

These two subroutines function together and are written separately only for the purpose of placing the longer COMMAND2 in an overlay, while leaving COMMAND1 in the root node.

COMMAND2 reads and decodes each command and either takes appropriate action directly or transfers control to COMMAND1 to call a subroutine in a different overlay.

Subroutine DERIVE(X,D,G)

This subroutine calculates the first derivatives of FCN with respect to each parameter, using the symmetric form

$$\partial F/\partial X = (F(X+D)-F(X-D))/2D.$$

The error is, to first order, independent of the step size D, while the error in an asymmetric form is proportional to D. The symmetric form is insensitive to a less than optimum choice of D and behaves well near the minimum where the first derivatives approach zero. The array G contains the first derivatives. DERIVE is called by MIGRAD.

Function DEXDIN(X,INT)

This function calculates the derivative of the external parameter U with respect to the internal parameter X,

$$dU/dX = \begin{cases} 1 & \text{(U unbounded)} \\ 0.5(B-A)\cos X & \text{(A < U < B).} \end{cases}$$

It is called by MINOS, OUTPAR, and OUTR when calculating parameter errors.

Subroutine DSET

This subroutine calculates the initial values of AMIN and of DIRIN(I). AMIN is the best (lowest) value of FCN yet obtained during the minimization and DIRIN(I) are the step sizes for the internal parameters corresponding to the given step sizes for the external parameters. DSET is called initially by LIFETIME and by ALTER after any changes are made that would affect FCN.

Subroutine EXTOIN(U,X)

This subroutine calculates the values of the internal parameters X in terms of the external parameters U. The relationship is

$$X = \begin{cases} U & \text{(U unbounded)} \\ \arcsin(2(U-A)/(B-A) - 1) & \text{(A < U < B).} \end{cases}$$

Subroutine FIXPAR(IX,IFIX)

This subroutine removes parameter numbered IX from the internal (variable) parameter list and renumbers the rest of the list to fill in the hole. IFIX is returned as the external parameter number if the parameter is variable, as 0 if it is not variable, and as -1 if NF parameters are already fixed. FIXPAR is called by FIXX following the command FIX and also by HESSE and MINOS.

Subroutine FIXX

This subroutine displays the currently variable parameters on the terminal and allows the user to enter the numbers of those he wishes to fix, either individually (e.g., 1, 2, 3) or as a range (e.g., 1-3). It then calls FIXPAR to fix each parameter designated. It is called by COMMAND2 following the FIX command.

Subroutine HESSE

This subroutine calculates the full second-derivative matrix of FCN by taking finite differences. It includes some safeguards against non-positive-definite matrices, and may set off-diagonal elements to zero in an attempt to force positivity. It fixes any parameter X for which both $\partial(FCN)/\partial X = 0$ and $\partial^2(FCN)/\partial X^2 = 0$. It inverts the symmetric second-derivative matrix to produce the covariance matrix

V(I,J) and calculates the estimated distance to the minimum value of FCN. If this estimated distance is negative or if V(I,J) is not positive definite, it then produces an estimate of the covariance matrix whose diagonal elements are reciprocals of the second derivatives and whose off-diagonal elements are zero. It is called by MIGRAD, by MINOS, and by COMMAND1 following the HESSE and MATOUT commands.

Subroutine INPAR

This subroutine reads the parameter file. It checks that the total number of parameters and number of variable parameters do not exceed dimensions, that transition rates occur only from higher to lower levels, that the initial values of parameters lie within limits, and that upper limits exceed lower limits. It also reads the title, decides whether the job is batch or interactive and discovers the names of the parameter and data files. If two transition rates are equal, it increases one of them by a factor of $(1 + 10^{-7})$ to avoid problems in solving the Bateman equations. It is called once by LIFETIME.

Subroutine INTOEX(X,U)

This subroutine calculates the values of the external parameters U in terms of the internal parameters X. The relationship is:

$$U = \begin{cases} X & \text{(U unbounded)} \\ A + (\sin X + 1)(B-A)/2 & (A < U < B). \end{cases}$$

It is seen that X can vary from $-\infty$ to $+\infty$ and U will remain bounded by A and B.

Subroutine MATOUT

This subroutine prints the covariance matrix V, and calculates and prints the individual correlation coefficients $V_{ij}/(V_{ii}\ V_{jj})^{1/2}$ which are the normalized off-diagonal elements of the covariance matrix. It is called by COMMAND1 following the MATOUT command.

Subroutine MIGRAD

This subroutine performs a function minimization using a gradient calculated by the subroutine DERIVE. The algorithm used is Fletcher's "switching" variation[5] of the original Davidon-Fletcher-Powell algorithm.[14] Starting values are given for the parameters X(I), the first derivatives G(I), and the covariance matrix V(I,J). Although G(I) must be a very good approximation to the gradient at X(I), the starting matrix V(I,J) may be poor, only a diagonal matrix, or even the unit matrix.

The minimization procedure involves taking a "Newton's step" to X'(J), where

$$X'(J) = X(J) - \sum_{I} V(I,J)\ G(I).$$

This would be the minimum if FCN were quadratic and if V(I,J) were the true covariance matrix. A new gradient G'(I) is calculated at X'(I). The matrix V(I,J) is corrected using an updating formula of the general form

$$V' = V + f(V,X,X',G,G').$$

The updating formula f(V,X,X',G,G') is either the original Davidon formula[14] or Fletcher's dual formula,[5] depending on Fletcher's "switching" criterion. Then G is replaced by G', X by X', and V by

V', and the above steps are repeated until some convergence criteria are satisfied.

The convergence criteria are based on the "estimated distance to minimum" SIGMA which is given by

$$\text{SIGMA} = \sum_{I,J} G(I)\, V(I,J)\, G(J).$$

This would be exactly twice the vertical distance FCN(X)-FMIN if the function FCN were exactly quadratic with covariance matrix V. Since this estimate is only as good as the estimate of V, an additional criterion is applied, namely that the successive estimates of V be not very different. The measure of difference used is the average fractional change, TRACE, in the diagonal elements where

$$\text{TRACE} = 2\Big(\sum_{I} \big|\, V'(I,I) - V(I,I) \,\big| \Big/ \sum_{I} \big(V'(I,I) + V(I,I)\big)\Big)/\text{NPAR},$$

and V' is the previous estimate of V. Convergence is achieved when both SIGMA < EPSI and TRACE < VTEST for two successive iterations or when SIGMA < 10^{-5} EPSI. EPSI and VTEST may be set by the user when the command MIGRAD is issued and have default values of EPSI = 0.1 UP and VTEST = 0.04. UP is set by the command ERROR and has a default value of 1.0.

Subroutine MINOS

This subroutine finds the true errors (confidence intervals) by examining the behavior of the chi-square function over the interval in question. (The theory is given in Ref. 15, pp. 203-205; see also Ref. 16.)

The MINOS errors are defined in terms of the size of the confidence interval specified by the value of UP. (UP may be set by the ERROR command. It has the default value UP = 1.0.) MINOS then determines the values of X_i for which the function $FM(X_i)$ attains the value FMIN + UP, where $FM(X_i)$ is the minimum of FCN with respect to all parameters except the parameter X_i, and FMIN is the overall minimum value of FCN. MINOS errors are then ERP_i and ERN_i, which are, respectively, the positive and negative increments which, when added to the value of X_i at FCN = FMIN, give a value to $FM(X_i)$ of FMIN + UP.

The details of the procedure are as follows: First, from the known covariance matrix of the function, MINOS determines the value of X_i and of the other parameters which would give $FM(X_i)$ = FMIN + UP if FCN were parabolic, that

$$X_i = X_{oi} + (V_{ii}UP)^{1/2}$$

$$X_j = X_{oj} + (V_{ii}UP)^{1/2} (V_{ij}/V_{ii}).$$

It then finds $FM(X_i)$ for this value of X_i by fixing X_i with FIXPAR and using MIGRAD to minimize FCN with respect to the remaining variable parameters. If $FM(X_i)$ is within EPSI of FMIN + UP, analysis stops. Otherwise, a new point X_i is chosen and $FM(X_i)$ is again calculated. The tolerance EPSI can be specified by the user when the command MINOS is executed and has the default value of 0.1 UP. This procedure is illustrated in Fig. 5.

X_i is varied in both the positive and negative directions from X_{oi}. The positive error ERP_i is $X_i - X_{oi}$ when $X_i > X_{oi}$, and the negative error ERN_i is $X_i - X_{oi}$ when $X_i < X_{oi}$.

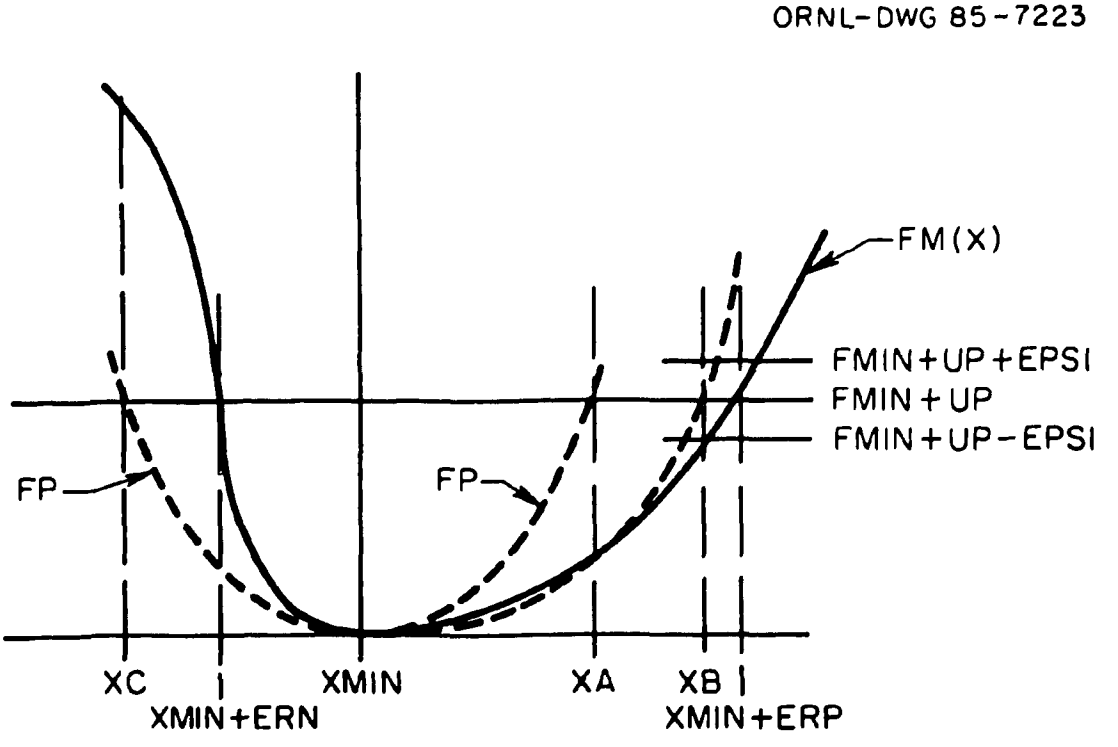


Fig. 5. Calculation of MINOS errors of parameter X. The symmetric parabola FP is predicted from the covariance matrix. The curve FM shows the true dependence of FCN on X. Since FM(XA) is not within EPSI of FMIN + UP, a new point XB is found by parabolic extrapolation, and the process is continued until FM(X) is within EPSI of FMIN + UP. A similar calculation is carried out on the negative side of XMIN.

When the program has difficulty converging, it plots $FM(X_i)$ as a function of X_i so the user can get a feeling for the nonlinearity of the problem.

If a new global minimum is found during a MINOS error analysis, a new SIMPLEX and MIGRAD minimization in the full variable space is attempted. After this step, MINOS error analysis recommences.

Subroutine OUTPAR(IKODE,IUNIT)

This subroutine generates output on the terminal or line printer, depending on the value of IUNIT. This includes the minimum FCN, number of calls to FCN, time, estimated distance tc minimum, parameter values, parameter errors, and changes in parameters since the last call to OUTPAR. The argument IKODE = 0, 1, 2 determines how much of this is printed.

Subroutine RANDOM

This is a shift-register random number generator which provides the random numbers needed by subroutine SEEK. It uses the methods of Kirkpatrick and Stoll[17] and James[18].

Subroutine RAZZIA(YNEW,PNEW)

This subroutine is called by SIMPLEX to replace a point in the current simplex with a new one. It also estimates the distance to the minimum and updates the step size and the estimate of the diagonal elements of the covariance matrix.

Subroutine RELEASE

This subroutine displays the parameters which are currently fixed, and allows the user to enter the numbers of those he wishes to

release, either individually (e.g., 1,2,3) or as a range (e.g., 1-3). It then calls RESTOR to release all of the parameters, and calls FIXPAR to fix those not designated. It is called by COMMAND2 following the RELEASE command.

Subroutine RESTOR(K)

This subroutine restores a fixed parameter to variable status by inserting it into the internal parameter list at the appropriate place. It estimates the diagonal element in the covariance matrix for this parameter as twice the reciprocal of the second derivative of FCN with respect to this parameter. If K = 0, all fixed parameters are restored. If K = 1, the last parameter fixed is restored. RESTOR is called by RELEASE, MINOS, and COMMAND2 following the RESTORE command.

Subroutine SEEK(NFP)

This subroutine performs a minimization of FCN using a Monte Carlo technique.[18,19] The function FCN is called NFP times. For each call, all variable parameters are chosen randomly, according to uniform distributions centered at the best previous parameter values, and with widths equal to the original step sizes or the parabolic errors from HESSE or MIGRAD. SEEK is called by COMMAND2 following the SEEK command.

Subroutine SIMPLEX

This subroutine uses the method of Nelder and Mead[4] for the minimization of a function of N variables. This method depends on the comparison of the function values at the (N + 1) vertices of a general simplex followed by the replacement of the vertex with the highest

value by another point. A simplex is the smallest N-dimensional geometrical figure with (N + 1) vertices: a triangle for N = 2, a tetrahedron for N = 3, etc. New simplices are formed by reflecting one point in the hyperplane of the other points. The simplex adapts itself to the local landscape of the function, elongating down inclined planes, changing direction upon encountering a valley at an angle, and contracting in the neighborhood of a minimum.

The initial simplex is obtained by coordinate variation. From the initial point (parameter set), the subroutine proceeds to find a local minimum along each coordinate (parameter) axis. These NPAR local minimi and the initial point define the initial simplex. This insures that the simplex has a reasonable size in each direction. If no minimum is found, the point used will be at least 0.064 DIRIN(I) and not more than 729 DIRIN(I) from the initial point.

The minimization of a function F proceeds as follows. Of the vertices, let P_L be the one for which F is smallest, and P_H be the one for which F is largest. Let $\bar{P}$ be the centroid of all vertices except P_H,

$$\bar{P} = \left(\sum_{i=1}^{N+1} P_i - P_H\right)/N.$$

A point P^* is produced by the reflection of P_H in the plane of the other points:

$$P^* = 2\bar{P} - P_H.$$

The action taken next depends on the value of $F(P^*)$:

(1) If $F(P^*) < F(P_L)$, the reflection has produced a new minimum, and

this direction is explored by computing

$$P^{**} = 2P^* - \overline{P}.$$

A parabola is fitted through $F(P_H)$, $F(P^*)$, and $F(P^{**})$. If it has a minimum, then the corresponding point is used to replace P_H. Otherwise, if $F(P^{**}) < F(P_L)$, P_H is replaced by P^{**}, and if $F(P^{**}) > F(P_L)$, P_H is replaced by P^*. The process is then iterated.

(2) Let P_h be the vertex with the second largest value of F. If $F(P_L) < F(P^*) < F(P_h)$, P_H is replaced by P^* and the process iterated.

(3) If $F(P^*) > F(P_h)$, where P_h is as in (2), then P_H is redefined as either P_H or P^*, whichever has the lower function value, and a point P^{**} is calculated by

$$P^{**} = 0.5(P_H + \overline{P}).$$

If $F(P^{**}) < F(P_H)$ P_H is replaced by P^{**} and the minimization repeated. Otherwise, only P_L is kept and a new starting simplex is evaluated by finding local minima.

The vertical distance to the minimum is of the order of $SIGMA = F(P_H) - F(P_L)$. Convergence is attained when SIGMA < EPSI for two successive iterations unless, on the last iteration, $F(P_H) - F(P) > 2$ EPSI.

MIGRAD has the advantage of producing a full covariance matrix, whereas SIMPLEX only gives estimates, which may be poor, of the diagonal elements. Thus, the choice between the two depends on the information required. However, a good estimate of the covariance matrix can usually be obtained after SIMPLEX by the command HESSE.

Subroutine SYMINV(A,N)

This subroutine inverts the symmetric N by N matrix A. It is called by HESSE.

Subroutine UCOPY(A,B,N)

This subroutine copies N words from array A into array B. It is called by MIGRAD and MINOS.

C. OTHER FUNCTIONS AND SUBROUTINES

These are specific to the version for the Perkin-Elmer computer.

Subroutine INIT

This subroutine initializes the program's real-time data structures prior to enabling task traps. See FORTRAN VII User Guide p. 6-26.

Subroutine ENABLE(3,HOLDIT)

This subroutine establishes the necessary data structures to allow the trap-handling subroutine HOLDIT to be entered and to have a message passed to it. See FORTRAN VII User Guide p. 6-27 and p. 6-30.

Subroutine GSTRING(STRING)

This subroutine returns the string of characters following the first comma of the START command in the CSS file. This string contains either B (for batch mode) or D (for interactive mode) and the names of the parameter and data files.

Function BTEST(N,31)

This logical function returns .TRUE. if bit 31 of integer N is a 1, which means N is odd, and returns .FALSE. if bit 31 is a 0, which means N is even.

VII. COMMON VARIABLES AND DIMENSIONS OF ARRAYS

Variables appearing in COMMON blocks which are not defined elsewhere in this report are described here.

A. COMMON VARIABLES IN LIFETIME

POP(L):	Initial population of level L.
POPINT(L):	Total decay intensity from level L.
FAM(L,M), M < L:	Transition rate from level L to level M.
FAM(L,L)	Negative of total transition rate out of level L.
FAM(L,M), L < M:	Coefficient a_{lm} of the Bateman equations.
DIAG(L):	Coefficient b_l of the Bateman equations.
RESULT(L,M), M < L:	Unshifted intensity of transition L → M.
RESULT(L,L)	Population of level L.
RESULT(M,L), M < L:	Shifted intensity of transition L → M.
TIM(I):	Flight time for data curve I.
ANORM(L):	Intensity normalization for experiment L.
RE(I,N,K):	Corrected intensity for shifted (K = 1) or unshifted (K = 2) data for flight distance I of data curve N.
FE(I,N,K):	Error in intensity RE(I,N,K).
NBRAT:	Number of branching-ratio data points.
BRANCH(N):	Value of branching-ratio data point N.
BRERR(N):	Error in branching-ratio data point N.
IBR(N,1):	Initial level for branching-ratio data point N.
IBR(N,2):	Final level for branching-ratio data point N.

IBR(N,3):	(= 1, include) (= 0 do not include) branching-ratio data point N in fit.
NMUL:	Total number of transitions, counting each member of multiplets separately, for which data curves are included.
AMU(N) :	γ-ray absorption coefficient in Ge for the γ-ray energy of data curve N.
OMEGAO(N):	Reference solid angle for the γ ray of data curve N.
WO:	Reference angular distrubution.
A2(N), A4(N):	Angular distribution coefficients for the γ ray of data curve N.

B. COMMON VARIABLES IN MINUIT.

MPAR:	Total number of parameters, variable and fixed.
NPAR:	Number of currently variable (internal) parameters.
U(J):	External parameter set.
ALIM(J):	Lower limit on U(J).
BLIM(J):	Upper limit on U(J).
WERR(J):	Initial step size for U(J). After minimization, it contains the parabolic error in U(J).
LCODE(J):	= 0 if U(J) is fixed, = 1 if U(J) is unbounded, = 4 if U(J) has upper and lower limits.

X(I):	Internal parameter set.
DIRIN(I):	Step size for X(I). It is derived from WERR(J) initially by DSET and adjusted during minimization.
LCORSP(J):	Index which associates X(I) with U(J): I=LCORSP(J).
V(I,J):	Covariance matrix.
AMIN:	Best (smallest) value of FCN (χ^2).
NFCN:	Number of calls to FCN during current command.
MXFCN:	= .FALSE. initially, = .TRUE. when NFCN exceeds NFCNMX.
NPFIX:	Number of parameters fixed by FIXPAR.
IPFIX(J):	External parameter number of the Jth parameter fixed by FIXPAR.
MINOFF:	= .FALSE. when MINOS is executing, = .TRUE. otherwise.
LMI(K):	Set of (external) parameter numbers for which a MINOS analysis will be performed.
NEWMIN:	= .FALSE. initially, = .TRUE. when MINOS finds a new minimum.
MINPLT:	= 0 for command MINOS, = 1 for command MINPLOT.
RPLT:	Range of each side of minimum for which MINPLOT is made.
IOUT:	Determines amount of printout provided.

ENF: = .TRUE. when SEEK, SIMPLEX, MIGRAD, or MINOS is executing,
= .FALSE. otherwise.

MYKILL: = .TRUE. after SEND STOP is executed,
= .FALSE. otherwise.

BATCH: = .TRUE. when in batch mode,
= .FALSE. when in interactive mode.

C. DIMENSIONS OF ARRAYS.

The dimensions of arrays are specified in PARAMETER statements so that they can be changed easily. The present values are:

NA = 50: no. of arguments that can be decoded by UNPACK.

NB = 5: no. of rotational bands.

NC = 50: no. of data curves.

ND = 20: no. of flight distances.

NE = 2: no. of data sets (experiments).

NF = 50: no. of parameters which can be fixed by FIXPAR.

NL = 80: no. of levels.

NP =200: no. of external parameters.

NR = 20: no. of branching ratios.

NS = 5: no. of energy levels in a rotational band.

NT = 60: no. of transitions (data curves plus multiplicities).

NV = 50: no. of variable parameters.

VIII. SAMPLE OUTPUT

The output on the following pages was obtained by running the program LIFETIME using the Parameter File on p. 27, the Data File on p. 28, the Velocity Data File on p. 30, and the sequence of commands listed on p. 31.

```
                    ********************
                    *  RECOIL-DISTANCE   *
                    *    CRNL *** 1982   *
********************************************************************************
  CASE1: INITIAL ESTIMATES                                  01 OCT 1985    15:54:22
********************************************************************************
                  INITIAL VALUES OF THE PARAMETERS READ FROM FILE     CASE1.P01
                  EXPERIMENTAL DATA READ FROM FILE                    CASE1.D01
********************************************************************************

     PARAMETER                 INITIAL VALUE      STEP SIZE          LOWER LIMIT      UPPER LIMIT

   1     POP(F2  )              5.00000           1.00000            0.000            100.00
   2     POP(F2' )              5.00000           1.00000            0.000            100.00
   3     POP(F3  )              5.00000           1.00000            0.000            100.00
   4     POP(F3' )              5.00000           1.00000            0.000            100.00
   5     POP(B4  )              20.0000           1.00000            0.000            100.00
   6     TR(L2  =>L1  )      1.000000E-02      4.999999E-03          0.000            500.00
   7     TR(L3  =>L2  )      0.100000          5.00000E-02           0.000            500.00
   8     TR(L3  =>L1  )      0.150000          5.00000E-02           0.000            500.00
   9     TR(L4  =>L3  )      0.200000          5.00000E-02           0.000            500.00
  10     TR(F2  =>L2  )      0.250000          5.00000E-02           0.000            500.00
  11     TR(F2' =>L2  )      0.350000          5.00000E-02           0.000            500.00
  12     TR(F3  =>L3  )      0.500000          0.100000              0.000            500.00
  13     TR(F3' =>F3  )      1.00000           0.200000              0.000            500.00
  14     QUAD(L4  )          2.00000           0.20000
  15     DZERO(1)            0.000000          0.100000             -1.00             1.0000
  16     NORM(2)             1.00000           0.100000
********************************************************************************
```

```
**********************************************************************************************
  CASE1: INITIAL ESTIMATES                                                  01 OCT 1985    15:54:48
**********************************************************************************************

CHARGE AND MASS OF RECOILING NUCLEUS: 70, 168.0

DATA SET # 1

CHARGE AND MASS OF STOPPER NUCLEUS: 82, 208.0

RECOIL VELOCITY:                5.996 +- 0.600 UM/PS      V/C: 0.020000 +- 0.002000

DENSITY OF STOPPER OR FOIL:     11.35 G/CM**3

AVERAGE WIDTH OF UNSHIFTED PEAK: 1.5 KEV

TIME WHILE SLOWING IN STOPPER OR FOIL:  0.917 PS

TARGET-DETECTOR DISTANCE:   5.000 CM
DETECTOR LENGTH:            2.000 CM
DETECTOR OUTER RADIUS:      1.200 CM
DETECTOR INNER RADIUS:      0.200 CM

DATA SET # 2

CHARGE AND MASS OF STOPPER NUCLEUS: 79, 197.0

RECOIL VELOCITY:               17.988 +- 1.799 UM/PS      V/C: 0.060000 +- 0.006000

SLOWED VELOCITY (EXPERIMENTAL): 11.992 +- 1.199 UM/PS      V/C: 0.040000 +- 0.006000

FOIL THICKNESS:                 4.59 MG/CM**2

DENSITY OF STOPPER OR FOIL:     19.32 G/CM**3

AVERAGE WIDTH OF UNSHIFTED PEAK: 1.5 KEV

TIME WHILE SLOWING IN STOPPER OR FOIL:  0.205 PS

TARGET-DETECTOR DISTANCE:   5.000 CM
DETECTOR LENGTH:            2.000 CM
DETECTOR OUTER RADIUS:      1.200 CM
DETECTOR INNER RADIUS:      0.200 CM

TAU(2): 30.0 PS
TAU(4): 10.0 PS

CURRENT FITTING CONDITIONS -

     THE FOLLOWING CORRECTIONS ARE BEING APPLIED INITIALLY ONLY
         POSITIONAL SOLID-ANGLE
         VELOCITY-DEPENDENT SOLID-ANGLE
         ALIGNMENT ATTENUATION
         LINESHAPE

         # INTEGRATION POINTS ALONG FLIGHT PATH :   11

         THE SOLID ANGLE WAS CALCULATED BY INTEGRATING OVER THE VOLUME OF THE DETECTOR,
         USING THE ABSORPTION COEFFICIENT FOR EACH GAMMA-RAY ENERGY.

         # INTEGRATION POINTS OVER DETECTOR :   11
```

```
          DECAY CURVES BEING FITTED -

              DATA SET # 1

              CURVE  1  L2     ===> L1    (FITTED)
              CURVE  2  L?     ===> L2    (FITTED)
              CURVE  ?  L3     ===> L1    (FITTED)

              DATA SET # 2

              CURVE  4  L?     ===> L2    (FITTED)

          BRANCHING RATIOS BEING FITTED -
              BRANCHING RATIO  1  L?     ===> L1    (FITTED)
              BRANCHING RATIO  2  L3     ===> L2    (FITTED)

     RANGE OF LEVELS USED IN CALCULATIONS: (L1   , B4   )

     FCN VALUE        CALLS     TIME        EDM
     3.3717E+02           1   15:54:55   0.00E+00

*********SEEK

     THE BEST VALUE FOUND BY SEEK IS

     FCN VALUE        CALLS     TIME        EDM
     1.1237E+01         160   15:56:03   0.00E+00

*********CORRECT

     CORRECTIONS RECALCULATED

*********SIMPLEX

     SIMPLEX MINIMIZATION HAS CONVERGED

     FCN VALUE        CALLS     TIME        EDM
     3.7534E-01         411   15:59:23   8.75E-02

*********VELCOMP

     VELOCITY CALCULATION RESULTS

     DATA SET # 1: E = (   0.00000)*C**2 + (   0.50000)*C + (   0.00000)

     TRANSITION: L3   ===> L2

     DISTANCE   SHIFTED CENTROID      SHIFTED ENERGY     UNSHIFTED CENTROID    UNSHIFTED ENERGY

       10.0      816.00 +- 0.10       408.00 +- 0.05       800.00 +- 0.10       400.00 +- 0.05
       20.0      816.00 +- 0.10       408.00 +- 0.05       800.00 +- 0.10       400.00 +- 0.05
       50.0      816.00 +- 0.10       408.00 +- 0.05       800.00 +- 0.10       400.00 +- 0.05
      100.0      816.00 +- 0.10       408.00 +- 0.05       800.00 +- 0.10       400.00 +- 0.05

     AVERAGE UNSHIFTED ENERGY:  400.00 +- 0.03 KEV

     AVERAGE RECOIL VELOCITY: 0.020229 +- 0.000090
```

```
         TRANSITION: L3   ***> L1

         DISTANCE   SHIFTED CENTROID      SHIFTED ENERGY     UNSHIFTED CENTROID    UNSHIFTED ENERGY

           10.0     1428.00 +- 0.20       714.00 +- 0.10      1400.00 +- 0.20       700.00 +- 0.10
           20.0     1428.00 +- 0.20       714.00 +- 0.10      1400.00 +- 0.20       700.00 +- 0.10
           50.0     1428.00 +- 0.20       714.00 +- 0.10      1400.00 +- 0.20       700.00 +- 0.10
          100.0     1428.00 +- 0.20       714.00 +- 0.10      1400.00 +- 0.20       700.00 +- 0.10

         AVERAGE UNSHIFTED ENERGY:  700.00 +- 0.05 KEV

         AVERAGE RECOIL VELOCITY: 0.020727 +- 0.000103

         DATA SET # 2: E = (   0.00000)*C**2 + (   0.50000)*C + (   0.00000)

         TRANSITION: L3   ***> L2

DISTANCE   SHIFTED CENTROID      SHIFTED ENERGY   LESS SHIFTED CENTROID  LESS SHIFTED ENERGY  UNSHIFTED CENTROID   UNSHIFTED ENERGY

  15.0      848.00 +- 0.10       424.00 +- 0.05       832.00 +- 0.10       416.00 +- 0.05       800.00 +- 0.10       400.00 +- 0.05
  30.0      848.00 +- 0.10       424.00 +- 0.05       832.00 +- 0.10       416.00 +- 0.05       800.00 +- 0.10       400.00 +- 0.05
  60.0      848.00 +- 0.10       424.00 +- 0.05       832.00 +- 0.10       416.00 +- 0.05       800.00 +- 0.10       400.00 +- 0.05

         AVERAGE UNSHIFTED ENERGY:  400.00 +- 0.03 KEV

         AVERAGE RECOIL VELOCITY: 0.060744 +- 0.000106

         AVERAGE SLOWED VELOCITY: 0.040481 +- 0.000105

         FINAL RESULTS

         DATA SET # 1  RECOIL VELOCITY:      6.064 +- 0.020 UM/PS      V/C: 0.020228 +- 0.000068

         DATA SET # 2  RECOIL VELOCITY:     18.211 +- 0.032 UM/PS      V/C: 0.060744 +- 0.000106

                       SLOWED VELOCITY:     12.136 +- 0.032 UM/PS      V/C: 0.040481 +- 0.000105

**********CORRECT

         CORRECTIONS RECALCULATED

**********MIGRAD

         MIGRAD MINIMIZATION HAS CONVERGED
         MIGRAD FINDS IMPROVEMENT
         NUMBER OF SUCCESSFUL STEPS:    5,    COVARIANCE MATRIX CODE: 3

         FCN VALUE       CALLS    TIME       EDM
         1.7870E-01       330   16:02:38   1.1EE-03

**********CORRECT

         CORRECTIONS RECALCULATED

**********SAVE

         PARAMETERS SAVED IN FILE CASE1.002

**********END
```

```
**********************************************************************************************
  CASE1: INITIAL ESTIMATES                                                  01 OCT 1985    16:03:22
**********************************************************************************************

MINIMUM CHI SQUARE:                0.177
NUMBER OF DATA POINTS:              32
NUMBER OF PARAMETERS BEING VARIED: 16
NORMALIZED CHI SQUARE:             0.010

  LEVEL          TOTAL TR. RATE                LIFETIME              POP-ZERO       TOTAL DECAYS

12  B4        0.7858 +-   0.0948      1.2726 +-   0.1536     15.008 +-  0.247      15.008
11  B3        0.3960 +-   0.0478      2.5252 +-   0.3047      0.000 +-  0.000      15.008
10  B2        0.1784 +-   0.0215      5.6059 +-   0.6764      0.000 +-  0.000      15.008
 9  B1        0.0685 +-   0.0083     14.6022 +-   1.7619      0.000 +-  0.000      15.008
 8  F3'       1.6607 +-   3.1095      0.6022 +-   1.1276      8.358 +-  0.715       8.358
 7  F3        0.1577 +-   0.0403      6.3426 +-   1.6206      4.673 +-  0.769      13.031
 6  F2'     107.3942 +- 185.6427      0.0093 +-   0.0161      4.317 +-  0.370       4.317
 5  F2      102.4842 +- 154.0590      0.0098 +-   0.0147      2.710 +-  0.741       2.710
 4  L4        0.4386 +-   0.3192      2.2798 +-   1.6589      0.000 +-  0.000      15.008
 3  L3        0.3083 +-   0.0106      3.2431 +-   0.1112      0.000 +-  0.000      28.039
 2  L2        0.0158 +-   0.0124     63.2081 +-  49.6558      0.000 +-  0.000      18.085
 1  L1        0.0000 +-   0.0000  ********** +-**********    0.000 +-  0.000      35.066

NON-ZERO TRANSITION PROBABILITIES

                  JI   JF  E(GAMMA) ALPHA     TRANSITION RATE          B(E2)              Q(TRANS)

L4   ===> L3     4.0  2.0   300.0  0.075   0.4386 +-  0.3192   13.6405 +-  9.9257   21.9079 +-  7.9708
L3   ===> L2     2.0  1.0   400.0  0.010   0.1216 +-  0.0007    0.0000 +-  0.0000    0.0000 +-  0.0000
L3   ===> L1     2.0  0.0   700.0  0.008   0.1867 +-  0.0105    0.0895 +-  0.0051    2.1210 +-  0.0599
L2   ===> L1     1.0  0.0   300.0  0.020   0.0158 +-  0.0124    0.0000 +-  0.0000    0.0000 +-  0.0000

BAND NO.  1, QUAD(L4  ) =   2.3840 +-  0.143E E+8

                  JI   JF  E(GAMMA)    TRANSITION RATE          LIFETIME

B1   ===> L4     6.0  4.0   500.0   0.0685 +-  0.0083   14.6022 +-  1.7617
B2   ===> B1     8.0  6.0   600.0   0.1784 +-  0.0215    5.6059 +-  0.6763
B3   ===> B2    10.0  8.0   700.0   0.3960 +-  0.0478    2.5252 +-  0.3047
B4   ===> B3    12.0 10.0   800.0   0.7858 +-  0.0948    1.2726 +-  0.1535

AUXILLIARY VARIABLES

DATA SET #              1                    2
DZERO           0.405 +-  6.9378    0.000 +-  0.0000
NORMALIZATION   1.000 +-  0.0000    0.993 +-  0.0842

BRANCHING RATIOS

                     EXPERIMENTAL VALUE    FITTED VALUE     DIFF/ERR

L2   ===> L1          0.600 +- 0.100          0.606         -5.613E-02
L3   ===> L2          0.400 +- 0.100          0.394          5.613E-02

MEAN CHI SQUARED:                                          3.151E-03
```

TRANSITION: L2 ===> L1 PLUS L4 ===> L3

DISTANCE	SHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	UNSHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	SUM CORRECTED	ERROR	DF/ER
10.0	0.09	0.08	0.90	0.18	-0.11	33.68	32.97	[illegible]	2.92	0.05	33.05	1.32	0.03
20.0	0.27	0.24	0.90	0.39	-0.16	33.44	32.79	0.98	2.70	0.09	33.03	1.33	0.02
50.0	1.81	1.67	0.92	1.80	-0.14	30.81	31.31	1.02	31.29	0.02	32.96	1.37	-0.02
100.0	7.11	6.79	0.95	6.76	0.02	24.46	26.18	1.07	26.33	-0.14	32.96	1.43	-0.03
				MEAN SQ						MEAN SQ	MEAN	ST DEV	MEAN SQ
				0.01423						0.00787	33.01	0.03	0.00

TRANSITION: L3 ===> L2

DISTANCE	SHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	UNSHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	SUM CORRECTED	ERROR	DF/ER
10.0	0.09	0.14	1.59	0.17	-0.02	6.48	10.90	1.68	10.89	0.01	11.05	2.31	0.00
20.0	0.40	0.63	1.57	0.67	-0.03	6.21	10.39	1.67	10.39	0.00	11.02	2.30	-0.01
50.0	1.80	2.76	1.53	2.67	0.06	5.12	8.27	1.62	8.39	-0.07	11.03	2.23	0.00
100.0	3.69	5.48	1.49	5.53	-0.03	3.63	5.58	1.54	5.53	0.04	11.07	2.14	0.01
				MEAN SQ						MEAN SQ	MEAN	ST DEV	MEAN SQ
				0.00138						0.00165	11.04	0.02	0.00

TRANSITION: L3 ===> L1

DISTANCE	SHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	UNSHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	SUM CORRECTED	ERROR	DF/ER
10.0	0.23	0.22	0.96	0.26	-0.04	16.34	16.77	1.03	16.72	0.05	16.99	1.41	0.02
20.0	1.09	0.96	0.96	1.03	-0.07	15.34	15.98	1.04	15.95	0.03	16.95	1.42	-0.02
50.0	4.35	4.23	0.97	4.10	0.13	11.91	12.71	1.07	12.88	-0.16	16.94	1.44	-0.02
100.0	8.55	8.41	0.95	8.49	-0.08	7.73	8.58	1.11	8.49	0.08	16.99	1.48	0.02
				MEAN SQ						MEAN SQ	MEAN	ST DEV	MEAN SQ
				0.00773						0.00886	16.97	0.03	0.00

TRANSITION: L3 ===> L2

DISTANCE	SHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	UNSHIFTED PEAK EXP. VALUE ORIGINAL	CORRECTED	ERROR	RESULT	DIFF/ERR	SUM CORRECTED	ERROR	DF/ER
15.0	0.02	0.03	1.47	0.04	-0.01	8.03	10.92	1.36	10.94	-0.02	10.94	2.00	0.00
30.0	0.10	0.15	1.46	0.19	-0.03	7.93	10.80	1.36	10.80	0.00	10.95	2.00	0.00
60.0	0.43	0.62	1.45	0.69	-0.05	7.55	10.31	1.37	10.29	0.01	10.93	1.99	-0.01
				MEAN SQ						MEAN SQ	MEAN	ST DEV	MEAN SQ
				0.00103						0.00017	10.94	0.01	0.00

ACKNOWLEDGMENTS

The authors would like to express appreciation to Hans Emling of G.S.I., Darmstadt, for providing a copy of his program MASTER, which was the starting point for the development of LIFETIME. We also acknowledge many helpful discussions with colleagues at ORNL, especially with Cyrus Baktash, M. W. Guidry, I-Yang Lee, Francis McGowan, Masumi Oshima, and Yves Schutz. Finally, we wish to give Francis McGowan special thanks for a careful reading of the manuscript.

REFERENCES

1. H. Emling, private communication

2. F. James and M. Roos, Comput. Phys. Commun. 10, 343 (1975).

3. J. C. Wells, N. R. Johnson, J. Hattula, M. P. Fewell, D. R. Haenni, I. Y. Lee, F. K. McGowan, J. W. Johnson, and L. L. Riedinger, Phys. Rev. C 30, 1532 (1984).

4. J. A. Nelder and R. Mead, Comput. J. 7, 308 (1965).

5. R. Fletcher, Comput. J. 13, 317 (1970).

6. T. Yamazaki, Nucl. Data A3, 1 (1967).

7. F. Rösel, H. M. Fries, K. Alder, and H. C. Pauli, Atomic Data and Nuclear Data Tables 21, 91 (1978).

8. N. R. Johnson, R. J. Sturm, E. Eichler, M. W. Guidry, G. D. O'Kelley, R. O. Sayer, D. C. Hensley, N. C. Singhal, and J. H. Hamilton, Phys. Rev. C 12, 1927 (1975).

9. R. J. Sturm and M. W. Guidry, Nucl. Instrum. Meth. 138, 345 (1976).

10. A. Abragam and R.V. Pound, Phys. Rev. 92, 943 (1953).

11. K. S. Krane, Nucl. Instrum. Meth. 98, 205 (1972).

12. H. H. Andersen and J. F. Ziegler, Hydrogen Stopping Powers and Ranges in All Elements, Vol. 3 of The Stopping and Ranges of Ions in Matter (Pergammon Press, New York, 1980).

13. J. F. Ziegler, Handbook of Stopping Cross-Sections for Energetic Ions in All Elements, Vol. 5 of The Stopping and Ranges of Ions in Matter (Pergammon Press, New York, 1980).

14. R. Fletcher and M. J. D. Powell, Comput. J. 6, 163 (1963).

15. W. T. Eadie, D. Drijard, F. E. James, M. Roos, and B. Sadoulet, Statistical Methods in Experimental Physics (North-Holland, Armsterdam, 1971).

16. F. James, Comput. Phys. Commun. 20, 29 (1980).

17. S. Kirkpatrick and E.P. Stoll, J. Comput. Phys. 40, 517 (1981).

18. F. James, Rep. Prog. Phys. 43, 1145 (1980).

19. F. James, Monte Carlo for Particle Physicists (Section 6.1) in: Methods in Subnuclear Physics, Ed. M. Nikolic, (Gordon and Breach) Vol. IV, Part 3, p. 117.

ORNL/TM-9105

INTERNAL DISTRIBUTION

1	C. Baktash	42	J. W. McConnell
2	J. B. Ball	43	F. K. McGowan
3	J. R. Beene	44	W. T. Milner
4	C. E. Bemis	45	F. Plasil
5	F. E. Bertrand	46	M. N. Rao
6	J. A. Biggerstaff	47	L. L. Riedinger
7	C. R. Bingham	48	R. L. Robinson
8	K. A. Erb	49	R. O. Sayer
9	R. L. Ferguson	50	P. H. Stelson
10	J. Gomez del Campo	51-70	J. C. Wells
11	E. E. Gross	71	A. Zucker
12	M. W. Guidry	72-121	Physics Division Library
13	M. L. Halbert	122-123	Central Research Library
14	D. C. Hensley	124	Document Reference Section
15-39	N. R. Johnson	124-127	Laboratory Records Dept.
40	A. J. Larabee	128	Laboratory Records-RC
41	I. Y. Lee		

EXTERNAL DISTRIBUTION

129 F. L. Culp, Physics Dept., Tenn. Tech. Univ., Cookeville, TN 38505

130 H. Emling, G.S.I., Postfach 110541, 6100 Darmstadt 11, WEST GERMANY

131-135 M. P. Fewell, Dept. of Nuclear Physics, The Australian National Univ., Canberra ACT 2600, AUSTRALIA

136 D. R. Haenni, Cyclotron Institute, Texas A & M Univ., College Station, TX 77843

137 J. S. Hattula, Dept. of Physics, Univ. of Jyväskylä, Nisulankatu 78, SF-40 720 Jyväskylä 72, FINLAND

138 R. L. Kozub, Physics Dept., Tenn. Tech. Univ., Cookeville, TN 38505

139 J. Lerner, College of Arts & Sciences, Tenn. Tech. Univ., Cookeville, TN 38505

140 J. C. Lisle, Dept. of Physics, Schuster Laboratory, University of Manchester, Manchester M13 9PL, ENGLAND

141 J. F. Mateja, Physics Dept., Tenn. Tech. Univ., Cookeville, TN 38505

142 M. Oshima, Dept. of Physics, Japan Atomic Energy Research Institute, Tokai, Ibaraki 319-11, JAPAN
143 S. C. Pancholi, Dept. of Physics, University of Delhi, Delhi 110007, INDIA
144 R. V. Ribas, Universidade De Sao Paulo, Institute de Fisica, Cidade Universitaria, C. P. 20516-Sp., BRAZIL
145 Y. Schutz, GANIL, B.P. 5027, 14021 Caen Cedex, FRANCE
146 Assistant Manager, Energy Research and Development, DOE-ORO
147-173 Technical Information Center, Oak Ridge, TN 37830