# ALLFIT: Lineshape Fitting Program for Nuclear Reactions

James J. Kelly
Department of Physics
University of Maryland, College Park, MD 20742
e-mail: jjkelly@physics.umd.edu

*Last updated: December 3, 1997*

## Abstract

ALLFIT is a versatile lineshape fitting program that analyzes spectra for a large variety of nuclear reactions. Sophisticated multiparameter resolution functions provide sufficient flexibility to accurately describe the lineshapes encountered in modern spectrometer experiments, including tails which can make a significant contribution to the cross section and to the background underlying nearby peaks, but which are often neglected by simpler programs. Both theoretical and empirical representations of the radiative tail are available for electron scattering. Resolution functions can be convoluted with intrinsic lineshapes. A multicomponent background function is available to represent continuum contributions. The fitting procedure uses Poisson statistics and provides considerable flexibility and control over the search. In addition to detailed formatted output, output files suitable for plotting and tabulation programs is provided.

**Contents**

## I. INTRODUCTION

ALLFIT is a versatile lineshape fitting program that analyzes spectra for a large variety of nuclear reactions [1]. The program is routinely used to analyze data for electron scattering [2–5], proton scattering [6–8], $(p, n)$ reactions, and transfer reactions at many laboratories. The methods employed are of even greater generality and the program is designed to facilitate user-generated applications. Examples have included coincidence timing peaks and spectra from Møller polarimaters.

ALLFIT can be applied to simple spectra or to very complicated spectra containing many peaks (more than 100 peaks are sometimes used) and multicomponent backgrounds, possibly with complicated continuum shapes. Sophisticated multiparameter resolution functions provide sufficient flexibility to accurately describe the lineshapes encountered in modern spectrometer experiments, including tails which can make a significant contribution to the cross section and to the background underlying nearby peaks, but which are often neglected by simpler programs. These resolution functions can be convoluted with intrinsic lineshapes. For electron scattering, an empirical radiative response function can be fitted to the data or a theoretical response function can be convoluted with resolution and intrinsic lineshapes. Proper use of Poisson statistics avoids the underfitting of low-statistics spectra that plagues traditional $\chi^2$ codes based on Gaussian statistics.

ALLFIT provides for a great measure of both flexibility and control over the search procedure. Any meaningful subset of parameters may be free to vary, fixed to initial values, of locked together. Free parameters are prevented from becoming unphysical. Locking a set of parameters together means that corresponding parameters in a set of peaks are constrained to vary together. For peak heights, locking constrains the ratio between the locked heights. For peak positions or thresholds, locking constrains the separations. For shape parameters, locking maintains numerical equality between locked parameters. All peaks sharing a locked parameter contribute to the determination of that parameter. Provision has been made for a search procedure that evolves through the use of several sets of vary codes. After convergence has achieved with one set of vary codes, the next set may permit more freedom to the fit.

The flexibility in the search procedure enables one to optimize both speed and quality of the fit. More importantly, it allows reliable extraction of unresolved peaks. When the levels of interest have negligible intrinsic width, or when their intrinsic lineshapes are known, a cluster of unresolved peaks can be fitted by locking their peak shapes or resolution functions to some strong isolated peak and locking their separations to known kinematic values. Thus, only the heights and a single overall cluster position are allowed to vary. This procedure has proven to be quite reliable in a large variety of applications over many years.

Nevertheless, despite the complexity of the spectra and the numerous options available, the input required to control the fit has been streamlined considerably. Summary files are used to supply the results to sorting and tabulation programs that minimize manual bookkeeping.

This document describes the use of the standard version of ALLFIT. The basic model is described in Sec. II and the search procedure in Sec. III. The organization of the program is described in Sec. IV. The input for the standard version is described in Secs. V and VI. Some practical advice to new users is given in Sec.VII and several sample cases are presented in Sec. VIII. Finally, input to several facility-specific versions is described in Appendix A.

This manual, which is still under development, was only recently converted into LaTeX from prehistoric formats at about the same time the program was tranferred from `VMS` to `UNIX`. Suggestions and improvements are always welcome and can be forwarded by mail or to e-mail address `jjkelly@physics.umd.edu`. Please alert me to any errors or bugs that you may find. Also, if you make a custom version for a major facility, please send me the appropriate input and summary routines along with documentation and a test case. If approved, I will include custom versions in future releases. However, please be aware that several unofficial and unsupported versions have appeared at various laboratories diverging from the mainstream as early as 1983. I cannot vouch for the reliability of those versions and will provide support only for versions that I have approved and distributed.


## ACKNOWLEDGMENTS

The original version was developed at MIT in 1981, drawing heavily upon earlier programs developed by many other authors. In particular, the procedures for radiative corrections are based in part upon work by J. Bergstrom, C. Creswell, and M. Deady. Assistance in implementing these procedures was provided by C.E. Hyde-Wright and F.W. Hersman. The hypergaussian lineshape was developed by B.E. Norum, who wrote the `PPFIT` program I used as the starting point for `ALLFIT`. The Poisson fitting procedure was developed with M.V. Hynes and B. Peterson. The multicomponent background was developed to meet the needs of C.E. Hyde-Wright and M. Manley. More efficient versions of the code were developed at the University of Maryland in 1987 to meet the needs of H. Seifert and A.E. Feldman. Several suggestions by J. Templon for reducing the system dependence of the source code have been implemented.


## II. FITTING FUNCTIONS

### A. Model

The fitting function consists of a background function $B$ plus a sum over individual peaks $y_i$ that may be represented as

$$y(x) = \epsilon(x)\left[B(x) + \sum_{i=1}^{N} y_i(x)\right]$$

where $\epsilon$ is an efficiency function. Each peak may be described as the convolution of an intrinsic lineshape $I(x)$, a resolution function $R(x)$, and a radiative response function $T(x)$, and is written schematically as

$$y_i(x) = I_i \otimes R \otimes T.$$

When appropriate, any of these components can be omitted. For example, states whose intrinsic width is much smaller than the experimental resolution do not require an intrinsic lineshape, so that $y_i$ reduces simply to $R \otimes T$. Furthermore, radiative corrections are only important for electron scattering and are not applied to hadronic reactions, so that narrow lineshapes reduce to $y_i(x) = R(x)$.

Spectra for electron scattering can be analyzed using either a theoretical calculation of the radiative response function $T$ or using an empirical lineshape, fitted to data, to represent the convolution of $R$ and $T$. The first method, referred to as the *theoretical tail*, is preferred whenever the data does not adequately constrain the radiative tail. For example, if there is no strong isolated peak in the spectrum or if the statistics are poor it may be difficult to fit an accurate response function to the data. However, if there is a strong isolated peak, it then becomes more efficient to fit an *empirical tail* to the data and to bypass some of the convolutions. The empirical tail calculation is much faster than the theoretical tail and generally gives almost identical results when the method is applicable.

Convolutions are performed using a fast Fourier transform routine (FFT). To expedite theoretical tail calculations, the Fourier transforms of the response functions for each nucleus can be written into a file. Significant reductions in computing time can then be realized by reading that file if it becomes necessary to repeat the fit for a particular energy and target.

Cross section calculations are made using analytic formulae for the integral of the lineshape. Thus, accurate cross sections can be obtained even for broad peaks that are not entirely contained within the fitting region. However, data are binned into discrete histograms. If these bins are coarse, the sum over a histogram may not equal the integral of the lineshape. Furthermore, it is sometimes difficult to produce accurate fits or rapid convergence when the peaks are so narrow that the full width at half maximum (FWHM) is only a couple of channels. Therefore, it is possible to subdivide the experimental channels during intermediate stages of the calculation and thereby achieve accurate agreement between sums and integrals. The fine channels are resumed before convolution and before comparison with data.

### B. Resolution Functions

The resolution function is assumed to depend upon the properties of the detection equipment and hence to be a function of projectile kinematics in the lab frame. Hence, the value of the resolution function in channel i is properly expressed in the form

$$R(i) = R(x_i)dx_i$$

where $x_i$ is the appropriate kinematic variable and $dx_i$ is the width of bin i with respect to x. For spectrometer experiments, $x$ is usually laboratory momentum for the projectile. Alternatively, data are sometimes binned in terms of missing mass for a particular reference reaction. For time-of flight experiments, it is convenient to use inverse velocity $x = 1/v$ instead of time so that the path length is eliminated.

Two types of resolution function are presently available: a simple asymmetric Gaussian and a sophisticated multiparameter lineshape that may be described as an asymmetric "hypergaussian" plus tails. The hypergaussian, considered to be the "standard" lineshape, requires ten parameters and contains the asymmetric Gaussian as a special case. The standard lineshape is divided into three regions. The left or high-energy tail is parameterized by an exponential. The central region is the asymmetric hypergaussian itself and may be either more or less pointed than an ordinary Gaussian depending upon the choice of exponent parameter $\gamma$. The right or low-energy side of the peak is also described by a tail function. Usually, the right tail is exponential. However, an inverse polynomial is also available to describe the electron radiative tail.

The asymmetric Gaussian has the form

$$R(x) = He^{-(x-p)^2/\sigma_L^2} \qquad x < p$$

$$R(x) = He^{-(x-p)^2/\sigma_R^2} \qquad x > p$$

where

$$\sigma_L = (1-\alpha)\sigma \qquad \sigma_R = (1+\alpha)\sigma.$$

Thus, this shape is described by four parameters consisting of height $H$, position $p$, width $\sigma$, and asymmetry $\alpha$. For the convenience of the user, the position is entered as a $Q$-value and the width is entered as full width at half maximum (FWHM)

$$w = \sigma 2(\ell n 2)^{1/2}.$$

The standard lineshape is described by the segmented function:

$$R(x) = Hf_L e^{-\lambda_L(m_L+p-x)/w_L} \qquad x < p - m_L$$

$$R(x) = He^{-[(p-x)/\kappa_L w_L]^\gamma} \qquad p - m_L < x < p$$

$$R(x) = He^{-[(x-p)/\kappa_R w_R]^\gamma} \qquad p < x < p + m_R$$

$$R(x) = Hf_R e^{-[(x-p-m_R)/w_R]^\mu} \qquad p + m_R < x$$

$$R(x) = Hf_R w[t_1(x-p-m_R-s)^{-1} + t_2 s(x-p-m_R-s)^{-2}] \text{ radiative tail.}$$

The exponential form of the right tail is used except when an empirical radiative tail is desired for electron scattering. The independent parameters common to either choice are designated height $H$, position $p$, width $w$, asymmetry $\alpha$, left match fraction $f_L$, left slope $\lambda_L$, and right match fraction $f_R$. For the exponential right tail, the parameters $t_1$ and $t_2$ are interpreted as slope and tail exponent $\lambda_R$ and $\mu$, respectively.

In addition, the dependent parameters defined below are used to simplify some of the expressions.

$$w_L = (1-\alpha)w \qquad w_R = (1+\alpha)w$$

$$m_i = \kappa_\gamma w_i(-\ell n f_i)^{-1/\gamma} \qquad (i = L, R)$$

$$\kappa_\gamma = 0.5(\ln 2)^{-1/\gamma}$$

$$s = (t_2 - t_1)w/f_R$$

The left and right match points $m_L$ and $m_R$ correspond to the boundaries between the central region and the tails. The width parameters $w_L$ and $w_R$ are chosen as the widths at half maximum for the left and right halves of the asymmetric central shape. The parameter $\kappa_\gamma$ converts the $w$ parameters into Gaussian widths $\sigma$. The parameter $s$ is chosen to ensure smoothness at the match point between the central shape and the empirical radiative tail for electron scattering. The functional form is designed to make the shape and tail parameters approximately independent of width and bin size.

A further refinement of the standard lineshape is provided by a smoothing option in which small regions surrounding the match points are replaced by the function

$$f(x) = \exp(a + bx + cx^2 + dx^3)$$

where the four parameters $(a, b, c, d)$ are chosen to ensure continuity and smoothness at the two ends of the smoothed region. This feature is purely cosmetic.

We have found that the flexibility provided by the standard resolution is essential to the analysis of data collected on many spectrometers, such as the QDDM or the K600 at IUCF and the HRS at LAMPF. The exponential tails often contribute as much as 10% to the cross section and cannot be adequately described by simpler Gaussian shapes. If the tails are neglected, it is not uncommon to underfit peaks areas by 10% or more. This problem is exacerbated by background and level density. In these situations, the background is frequently and erroneously assigned counts that belong in the tails of peaks.

Similarly, the central shapes are almost always asymmetric and are often much more pointed the a Gaussian ($\gamma < 2$). The details of the instrumental resolution, on the other hand, are somewhat difficult to predict because they depend in a complicated manner upon the spectrometer aberrations, beam profile, detector response, and target thickness and uniformity. An analysis of these factors is beyond the scope of the report. The important point is that accurate analysis requires a lineshape that accommodates these aspects of the instrumental resolution.

### C. Intrinsic Lineshapes

Intrinsic lineshapes are assumed to depend upon the $Q$-value of the target rather than on the kinematics of the projectile. Hence, the value of an intrinsic lineshape in channel $i$ is given by

$$I_i = I(Q_i)dQ_i$$

where $Q_i$ is the $Q$-value and $dQ_i$ is the bin width for the appropriate isotope. Therefore, `ALLFIT` stores arrays of $Q$ and $DQ$ for each isotope described by the target file.

Three intrinsic lineshapes are presently available, Gaussian, Breit-Wigner, or Lorentzian. The functional forms listed below involve the parameters: height $H$, position $Q_0$, width $\sigma$ or $\Gamma$, and threshold $Q_t$.

Gaussian: $I(Q) = He^{-(Q-Q_0)^2/\sigma^2}$

Breit-Wigner: $I(Q) = \frac{H(\Gamma/2)^2}{(Q-Q_0)^2+(\Gamma/2)^2}$

Lorentzian: $I(Q) = \frac{H(Q-Q_t)^2\Gamma^2}{[(Q-Q_t)^2+(Q_0-Q_t)^2]^2+(Q-Q_t)^2\Gamma^2}$ $\quad Q < Q_t$

$I(Q) = 0.0$ $\quad Q > Q_t$

### D. Background

The background function $B$ can receive up to three contributions

$$B = B_0 + B_1 + B_{ext}$$

where $B_0$ represents the ambient background, $B_1$ represents the nuclear continuum, and $B_{ext}$ represents an external function. The ambient background is described by a cubic function

$$B_0(x_i) = b_0 + b_1 x_i + b_2 x_i^2 + b_3 x_i^3$$

where $x_i$ is the appropriate kinematic variable at channel $i$. The continuum is assumed to begin at some threshold $Q_t$ and is described by the polynomial function of

$$B_1(x_i) = b'_1(x_i - x_t) + b'_2(x_i - x_t)^2 + b'_3(x_i - x_t)^3$$

where $x_t$ is the kinematic variable associated with threshold $Q_t$. The threshold is associated with element `IBACK` drawn from the target description. The external background function is based upon a histogram $y_B(i)$ that is read from an external file and is then shift and scaled according to

$$B_{ext}(i) = S y_B(i + \Delta i)$$

where $\Delta i$ is a discrete shift.

The external background option is sometimes useful when analyzing data acquired with a mixed isotope target. For example, we have used this option to analyze the high excitation region of $^{16}$O spectra collected with BeO targets. The $^{9}$Be continuum represented a rather complicated background beneath the $^{16}$O peaks of interest. Similar data were also collected with a pure $^{9}$Be target. We analyzed the $^{9}$Be spectrum first and then subtracted from the data lineshapes fitted to the narrow peaks. The resulting continuum spectrum was then smoothed. The smoothed spectrum was supplied to the BeO analysis as one component of the background. The scale factor was used to correct for normalization differences and the shift was used to correct for differential energy loss in targets with different thickness. As can be seen from the figures in Ref. [3] , this technique was extremely helpful in analyzing these spectra.

### E. Radiative Corrections for Electron Scattering

**Missing section**

## III. SEARCH PROCEDURE

The search procedure uses the Marquardt gradient-expansion algorithm and is loosely based upon the `CURFIT` subroutine of Bevington [9]. A gradient search is suitable when the parameters are relatively far from the $\chi^2$ minimum but converges slowly near the minimum. Linearization of the fitting function is optimal near the minimum, but is unreliable beyond the parabolic region of the $\chi^2$ hypersurface. The Marquardt algorithm interpolates between these extremes and gives better performance.

**Incomplete section: discuss modifications for Poisson statistics**

### A. Constraints

The desirability of enforcing constraints among the fitting parameters stems from both practical and physical considerations. On the practical side, the most general lineshapes have many parameters. If all of the parameters of each peak were allowed to vary independently, the sheer number of free parameters would require a huge amount of storage, would introduce a great degree of correlation, and would bog down the computation such that it would collapse under its own weight. Furthermore, due to the proximity of peaks and the presence of backgrounds, it is generally impossible to determine all of the parameters. On the physical side, we generally expect that the experimental resolution varies slowly and smoothly across the spectrum. We also usually know the positions of most of the peaks and use that knowledge to advantage when the peaks are not clearly resolved.

Two types of constraints are available: fixing a parameter to its initial value of locking a set of parameters together. The operational effect of locking parameters depends upon the interpretation of the parameter. Background parameters cannot be locked to any other parameter. Locking a height parameter of one peak to another maintains a constant ratio between these heights. Locking position parameters together maintains constant separations between the peaks that are locked. However, the overall position of a cluster of locked peaks may be allowed to vary. Locking a threshold parameter to the corresponding position parameter for the same peak maintains a constant separation between peak and threshold. Similarly, locking different thresholds together maintains constant separation between them. Finally, locking a shape parameter in one peak to the corresponding parameter in another maintains numerical equality between these parameters.

The constraints are specified by one or more sets of vary codes (`VC`'s). The first set governs the initial search path. If desired, one can supply additional sets which are used after a specified number of iterations has been performed or after convergence has been achieved with the preceding set. This feature can be very useful. It is generally advisable to begin a fit with a restrictive set of vary codes that permits only the most essential parameters to vary. Then, after the fit becomes reasonably good, some of the constraints can be relaxed in subsequent vary codes as the fit is fine-tuned.

The vary code for each parameter is specified by an integer value and is given in formats described in Sec. V. A value of 0 is used to fix a parameter. A free parameter is indicated by a code `IFREE` selected in the include file. This parameter must be larger than the maximum number of peaks. For historical reasons, `IFREE` is usually assigned the value 90. An intermediate value specifies the number of the peak to which the parameter is locked. For a threshold parameter, a vary code equal to its own peak index locks the threshold to its own position. For all other cases, vary codes can refer only to preceding peaks. However, this limitation has no practical consequences because all peaks sharing a locked parameter contribute to the determination of that parameter. Therefore, there is no difference between locking a parameter in a large peak to a small peak or vice versa. It is also permissible to lock a to $b$ to $c$ in a "daisy chain" fashion.

Some care must be exercised in using constrained peak heights. Often, we wish to fix a peak height because we already know the cross section for that peak, especially when the peak corresponds to an impurity or a small fraction of a mixed isotope target. Therefore, although initial values for free heights are supplied as counts, initial values for fixed heights are entered as cross sections in the center-of-mass. `ALLFIT` then determines the appropriate height from the area of the peak and the normalization data. Similarly, a height locked to another is entered as the ratio between the cross sections for the two peaks. For example, suppose that the height of peak $B$ is locked to that of peak A. If the height of peak A is free, it is entered in counts; if it is fixed, it is entered as a cross section. The height of peak $B$ is then entered in the input file as the ratio between the cross sections $B/A$. The proper ratio between the two height parameters is determined from the initial cross section ratio and is held constant during the iteration procedure. However, if the shapes these two peaks are varied independently, the cross section ratio will change and the spirit of locking will be defeated.

## IV. PROGRAM ORGANIZATION

### A. Language

`ALLFIT` is written in nearly-standard `FORTRAN-77` that employs only the most common extensions to the ANSI standard, such as generic functions and the `INCLUDE` statement. The basic array dimensions are declared as parameters, from which all related dimensions are computed. Hence, it is easy to adjust the sizes of data arrays and the maximum numbers of peaks and free parameters as needed. However, to ensure consistency between all of the subprograms, these parameters and virtually all the common blocks are collected in a file that is included in the appropriate modules using compiler directives. This directive takes the form

```
INCLUDE 'allfit.inc'
```

and it is the responsibility of the user to ensure that a link to the appropriate `allfit.inc` file is present. Note that the entire program must be recompiled if the include file is changed. In addition, some of the versions also use application-specific include files. Most arithmetic is performed in single precision, with the exception of the matrix inversion routine which accepts single-precision arguments but uses double-precision calculations.

The `ALLFIT` program consists of a titular main program and a library called `allfit_lib`. Most files in `allfit_lib` contain a single subprogram or a few closely related subprograms and should be independent of platform or operating system. The subroutines which read data files and produce output files intended for plotting and tabulation programs are collected in the file `allfit_iofiles.f` and may depend upon platform. Early versions run under the `VMS` operating system used unformatted data files for which i/o functions are platform dependent; specific information about those versions can be found in Appendix A. However, because the earlier motivation to conserve disk space is no longer as compelling and because both the `VMS` operating system and the experimental facilities for which those versions were written are rapidly becoming obsolete, the generic version that is presently distributed uses simple formatted data, plot, and summary files. The generic version should run under either `UNIX` or `VMS` with virtually no adjustments for platform. Nevertheless, in place of the generic version many users will find it useful to link to a custom version of `allfit_iofiles` that reads data files produced by a replay program and which produces output suited to their favorite display packages.

Custom versions may involve user-written application libraries and/or modications of `allfit_iofiles`. The minimum set of subroutines that must be supplied within either the application library or the custom `allfit_iofiles` consists of:

- `BANNER` : produces a banner upon the output file announcing the program version;
- `DREAD` : reads the experimental data and computes conversion factors relating peak sum to cross section;
- `READB` : reads external background file, if needed
- `PLOT` : produces a file containing the data, fit, background, and lineshapes in a form suitable for input to a plotting package;
- `SUMMARY` : produces a summary file used by subsidiary programs to tabulate results.

Any additional subprograms referenced by `DREAD` or `SUMMARY` should also be placed in the application library or in `allfit_iofiles`. Each unique version of `ALLFIT` is created by linking the main routine, user-written routines, and finally `allfit_lib`.

## B. I/O Files

The basic input is read from logical unit number 5 (`LUN5`) and the printed output is written to `LUN6`. Depending upon the application, several other files may be used for input or output purposes and are assigned logical unit numbers from the following table. In addition, custom versions may employ additional files also.

- `LUN5` : basic input is read from `stdin`
- `LUN6` : basic output is written to `stdout`
- `LUN3` : describes intrinsic properties for each final state (optional)
- `LUN4` : input file for external background (optional)
- `LUN7` : output file for plotting (optional)
- `LUN8` : input/output file for radiative tail (optional)
- `LUN9` : output file for fit summary (optional)

### C. Installation Procedure

ALLFIT is distributed as an archive file that is self-extracting under UNIX. The file allfit.uu is created using tar to archive, gzip to compress, and uuencode to encode and contains extraction instructions at the beginning of the file. Simply create a main directory for ALLFIT, move the allfit.uu to that location, and then under UNIX issue the commands

```
chmod 744 allfit.uu
allfit.uu
```

The top directory will then contain:

| | | |
|---|---|---|
| allfit.f | : | main program |
| allfit.inc | : | include file |
| Makefile | : | make file for compilation and linking |

plus subdirectories LIB containing the library source and make files and SAMPLES containing sample input and output files. To install the program, follow these steps.

- Check that the parameters in the include file are sufficient for your needs.
- Verify that LIB contains a link to the include file.
- Modify the Makefiles to account for system-dependent qualifiers.
- Execute make in the library subdirectory.
- Execute make in the main ALLFIT directory.
- Create an environment variable $allfit_dir pointing to the main directory.
- Save the sample output files for comparison.
- Run the sample cases using the script $allfit_dir/SAMPLES/fit and compare output, summary, and plot files with those provided. For example, fit pp_sample1 would run the first $(p, p')$ sample.
- If the PHYSICA plotting package is available, inspect sample results using the plotfit.pcm command macro.

After successful completion of the installation procedure, you may wish to create a customized version suitable for your data format or plotting packages.

## V. INPUT DESCRIPTION

The input read from LUN5 can be subdivided into six subsections:

| | |
|---|---|
| A: | initial input of file names and reaction; |
| B: | input from DREAD; |
| C: | kinematic calibration data; |
| D: | specification of fitting region; |
| E: | control integers; |
| F: | initial fitting parameters and vary codes; |
| G: | additional sets of vary codes, if needed. |

The subsections are described separately.

In general, each record of input will be listed in the form:

LINE A.1 (format): variable list.

The format description can either be a specific `FORTRAN` format descriptor code, such as `A80`, or an asterisk (*) signifying free format.

All variables in the list must be supplied, delimited by either spaces or a comma, when using free format. Character variables must be enclosed by single quotes. At least one character, possibly a blank space, must be enclosed. However, the spacing within the list is arbitrary, and may occupy several lines. This flexibility permits the user to separate sections of input by one or more lines for clarity. The spacing of each free format line can be chosen for visual appeal.

It is possible to terminate reading a free-format list of numerical values using a slash (/) at the end of the record; the remaining variables will be unaffected. `ALLFIT` resets uninitialized variables to appropriate default values. Thus, it is not necessary to supply redundant lineshape parameters or to input a string of commas in place of zeros. However, this terminator may not work for character variables.

### A. Initial Input

| | | |
|---|---|---|
| LINE A.1 | (A): | Title of fit used on output file. |
| LINE A.2 | (*): | Reaction label of the form $(a, b)$, where $a$ and $b$ are symbols representing the projectile and ejectile, respectively. |
| LINE A.3 | (A): | Name of data file, including device and directory. |
| LINE A.4 | (A): | Name of file containing external background arrays. |
| LINE A.5 | (A): | Name of plot file. |
| LINE A.6 | (A): | Name of file containing level-diagram information used to initialize peak parameters (see Sec. VI A). |
| LINE A.7 | (A): | Name of summary file. |

All lines must be present, but lines 1, 4, 5, or 6 may be left blank. No plot file is produced if line 5 is blank. The reactions presently recognized include `REACT` = `'(e,e)'`, `'(p,p)'`, or `'(p,n)'`. For other reactions, the following line must follow Line A.2.

| | | |
|---|---|---|
| LINE A.2.1 (*): | `LEPTON` | $= 0 \rightarrow$ not electron scattering |
| | | $= 1 \rightarrow 1$ electron scattering |
| | `PMASSI` | = projectile mass in pmu (938.256 MeV) |
| | `PMASSE` | = ejectile mass in pmu |
| | `PCHARG` | = projectile charge (e) |

## B. DREAD Input

The subroutine DREAD is called at this point to obtain the spectrum, the experimental parameters, and target information from LUN4. DREAD must also return an array of conversion factors { SIGFACT(J), J=1,NELS } for each element of the target which relates the differential cross section for a peak to its yield according to

$$\mathtt{SIGMA(I)} = \mathtt{SIGFACT(IDPK(I))} * \mathtt{PKAREA(I)}$$

where I is the peak index, IDPK(I) is the element identification for peak I, PKAREA(I) is the peak integral computed from its lineshape parameters, and SIGMA(I) is the differential cross section in the laboratory frame. This section describes the input structure for the standard version, while Appendix A describes the input for several specialized versions.

The normalization parameters are read using the format below. Note that LINE B.3 is repeated for each element of the target and that only the relative abundances matter – it is not necessary for the fractional abundances to sum to unity. The target angle $\theta_t$ is defined as the angle between the normal and the beam, and is read as variable TANGLE in degrees. Also note that the acceptances in scattering angle, ACCH and ACCV, are not used by ALLFIT but are passed along to tabulation or other secondary programs through the SUMMARY file.

| | | | |
|---|---|---|---|
| LINE B.1 | (A): | Name of target | |
| LINE B.2 | (*): | NELS | = number of elements |
| | | THICK | = areal thickness in mg/cm$^2$ |
| | | DENS | = density in g/cm$^3$ |
| LINE B.3 | (*): | ISOLAB(I) | = isotope label |
| | | Z(I) | = atomic number |
| | | ATMSS(I) | = atomic weight |
| | | FRACT(I) | = fractional abundance |
| LINE B.4 | (*): | E0 | = beam energy in MeV |
| | | THETAL | = scattering angle in degrees |
| | | TANGLE | = target angle in degrees |
| | | ACCH | = horizontal acceptance in mr |
| | | ACCV | = vertical acceptance in mr |
| | | ACC | = solid angle in msr |
| | | CHARGE | = beam charge in Coulombs |
| | | DTC | = dead-time correction factor |

The standard version assumes binary reactions and prints cross sections in units of mb/sr. The cross section normalization factors are then

$$\mathtt{SIGFACT(I)} = 2.6598 \times 10^{-10} \frac{\mathtt{DTC} * \mathtt{SIGAM} * \cos\theta_t}{\mathtt{FRACT(I)} * \mathtt{THICK} * \mathtt{CHARGE} * \mathtt{ACC}}$$

where

$$\mathtt{SIGAM} = \sum_{\mathtt{I}=1}^{\mathtt{NELS}} \mathtt{ATMSS(I)} * \mathtt{FRACT(I)}$$

corresponds to the mean atomic weight for normalized abundances. Note that custom versions may find it more convenient to express the experimental parameters in other units or to obtain normalization information from the output files produced by replay programs, but must finally return normalization factors which give the laboratory cross sections are expressed in units of mb/sr. For more general reactions you will need to modify the conversion factors and should also adjust the labels on tabulated cross sections.

Finally, the experimental spectrum is read from the same file, `LUN4`, in free format

```
K, Y(K), K=1,MNPTS
```

for as many channels as are present or up to the dimensions of the data array.

### C. Kinematic Calibration

The input structure for this subsection depends upon the choice of kinematic variable `KINVAR` associated with the histogram channels. Four variables are available:

**(a)** `KINVAR` = ‘`CHANNEL`’: no calibration is performed for this choice because no kinematic variable has been chosen.

**(b)** `KINVAR` = ‘`EXCITATION`’: the excitation energy scale is based upon the reference reaction.

**(c)** `KINVAR` = ‘`MOMENTUM`’: laboratory momentum loss is calibrated to the positions of selected peaks.

**(d)** `KINVAR` = ‘`TIME`’: laboratory inverse-velocity is calibrated to the positions of selected peaks.

The `MOMENTUM` and `TIME` variables establish kinematic scales for which recoil corrections can be applied automatically, using the target mass and the $Q$-value for the peak. The `CHANNEL` and `EXCITATION` scales are not kinematic in nature; recoil corrections cannot be applied to these variables. Therefore, differential recoil corrections must be included in the specification of peak positions explicitly when using non-kinematic scales.

The kinematic calibration associates the appropriate variable $X$ with channel `I` using the polynomial

$$\mathtt{X(I)} = \sum_{\mathtt{K}=1}^{\mathtt{NED}} \mathtt{ECAL(K)} * \mathtt{I} * *(\mathtt{K} - 1).$$

At present `NED`$\leq$ 3. The calibration parameters `ECAL` can either be read directly, when `CALMODE=‘READ’`, or can be fitted to the positions of `NED` peaks, when `CALMODE=‘CALC’`. If fitting to the `EXCITATION` scale, `NED` pairs of (excitation energy, channel) are read at line C.4. If fitting to a kinematic scale, `NED` triplets of (atomic mass number, $Q$-value, channel) are read instead.

LINE C.1 (*): `KINVAR`
LINE C.2 (*): `AREF` = atomic mass number of reference target
`QREF` = $Q$-value for reference reaction

LINE C.3 (*): `CALMODE`
LINE C.4 (*): `NED`, calibration data

Note that lines are 2-4 are omitted when using `KINVAR='CHANNEL'`.

### D. Fitting Region

The fitting region can be selected using either channel input or energy input, except that when `KINVAR='CHANNEL'` only channel input is allowed. Excitation energy is used for the `EXCITATION` scale, whereas $Q$-values are used for kinematic scales.

| | | | |
|---|---|---|---|
| LINE D.1 (*): | `MODE` | = | 'CHANNEL' or 'ENERGY' |
| | `BEGIN` | = | beginning of fitting region |
| | `END` | = | end of fitting region |

Note that the variable `MODE` is omitted when `KINVAR='CHANNEL'`.

### E. Control Integers

Line E.2 is only read for electron scattering using the theoretical radiative response.

| | | | |
|---|---|---|---|
| LINE E.1 (*): | `NPKS` | = | number of peaks |
| | `NOVCS` | = | number of vary-code sets |
| | `MODPOS` | : | governs input of positions and widths |
| | | = | 0 → energy units |
| | | = | 1 → channel input |
| | `LIS1ST` | = | 0 → omit list of initial fit |
| | | = | 1 → list initial fit |
| | `LSTLST` | = | 0 → omit list of final fit |
| | | = | 1 → list final fit |
| | `KTAIL` | : | determines form of radiative tail; only meaningful for electron scattering. |
| | | = | 0 → empirical radiative tail |
| | | = | 1 → theoretical radiative response |
| | `ISMTH` | : | Measures size of region surrounding match points in which smoothing may be performed. |
| | | < | 0 → no smoothing |
| | | = | 0 → smoothing within 2 channels on either side of match points |
| | | > | 0 → use `ISMTH` channels on either side |
| | `ITWICE` | : | Controls array size for convolution. |
| | | = | 0 → convolution array uses `MNPTS` |
| | | = | 1 → convolution array used `MNPTS*2` |
| | `ICOND` | : | Governs inclusion of derivatives of resolution function convoluted against intrinsic lineshapes. |

| | | | |
|---|---|---|---|
| | | = | 0 → omit |
| | | = | 1 → include |
| | `IBIN` | : | Governs mesh size for lineshape computations. |
| | | = | 0 → internal decision |
| | | = | 1 → 1-to-1 binning |
| | | > | 1 → fine mesh with `IBIN`-to-1 binning |
| | `ISTAT` | : | specifies goodness-of-fit criterion |
| | | = | 0 → Poisson statistics |
| | | = | 1 → Gaussian statistics |
| LINE E.2 (A): | `RADFIL` | = | name of file containing Fourier transform of theoretical radiative response function. If the file exists, it will be read and the calculation will be skipped. If the name of a nonexistent file is supplied, it will be created and filled. If `RADFIL=NONE`, no file will be produced. Since the radiative response only depends on target, use of this file can save considerable computation time. |
| LINE E.3 (*): | `DLAM` | = | Lambda step size used in computation of radiative tail. This parameter is read but ignored if a valid `RADFIL` is present. |

### F. Initial Fit Parameters

The initial lineshape parameters can be entered in either energy or channel form, as selected by `MODPOS`. The parameters will then be transformed into appropriate internal form and the constraints enforced. These parameters are then transformed back into the original input form so that the listing can be conveniently used to verify the input. However, after having undergone several stages of manipulation, the listed parameters may no longer be exactly equal to their input values. This is not a mistake, but is an unavoidable consequence of providing so much flexibility for parameter input.

Resolution lineshapes are expressed as functions of the kinematic variable, such as momentum loss. However, most users find it preferable to express the width in energy so that the FWHM represents the energy resolution. Therefore, this width parameter is entered as energy (or channel) width and transformed to the internal kinematic variable. It is returned to the original form when listed on the output. Intrinsic lineshapes, such as Lorentzians, are naturally expressed as functions of $Q$-value. Therefore, the width and threshold parameters remain in energy form. However, the positions of all peaks are expressed internally in kinematic form and externally in energy form.

There are two schemes for supplying the parameters of the initial guess. First, if the file name `LEVELS` is blank or if the channel scale is employed, the input structure is very similar to that used by early versions of `ALLFIT`. This method requires the initial parameters for each peak to be supplied in one block, followed by a block a vary codes. Setting up these vary codes requires considerable care in enumerating the peaks and is prone to error when many more peaks are needed than will fit on an editor screen.

The second method uses the level-diagram information to condense the input and minimize redundancy. This method is much less cumbersome and is recommended as normal procedure. After defining the variables common to the two methods, separate subsections present each method.

Several methods can be used to specify resolution functions when convolution against intrinsic lineshapes is desired. The choice of method is made by specifying a resolution name `RESNAM(I)` corresponding to each peak name `PKNAME(I)`. If `RESNAM(I)` is blank, peak `I` is assumed to be narrow and is not convoluted with any resolution function. IF `RESNAM(I)=PKNAME(I)`, the width parameter for a simple Gaussian resolution function is appended to the lineshape parameters for peak `I`. This option is not available when a `LEVELS` file is employed. If `RESNAM(I)` is the same as some other `PKNAME(J)` and `J` $\neq$ `I`, we use the shape of peak `J` as the resolution function for peak `I`. Of course, peak `J` should be a narrow state of the same isotope containing the broad state `I`. Finally, if `RESNAM(I)` is not blank but is not equal to any of the peak labels, we assume that an external resolution function of `'STANDARD'` type will be described by its own lineshape parameters at a later point in the input stream.

The search with a particular set of constraints is terminated when any of the following three criteria are satisfied.

| | | |
|---|---|---|
| `ITVCS` | = | Maximum number of iterations, possibly zero. |
| `CONV` | : | Relative change in chi-square between iterations drops below `CONV`. |
| `CHIVCS` | = | Chi-square drops below `CHIVCS`. |

The convolution of derivatives of an intrinsic lineshape can consume much time profitlessly, particularly when the fit is far from convergence. These computations can be optionally disabled according to the criteria below.

| | | |
|---|---|---|
| `CHID` | = | Maximum chi-square for convolution of derivatives. |
| `DCHID` | = | Maximum relative change in chi-square for convolution of derivatives. |

However, the radiative response function for electron scattering is always important. Therefore, these options are ignored for electron scattering.

### 1. Input without LEVELS

| | | |
|---|---|---|
| LINE F.1 (*) | : | `ITSVCS, CONV, CHIVCS, CHID, DCHID` |
| LINE F.2 (*) | : | Background parameters (requires 10). |
| LINE F.3 (*) | : | Background vary codes, followed by `IBACK`. |

The following pair of lines in repeated for each of the peaks.

| | | |
|---|---|---|
| LINE F.4 (*) | : | `PKNAME, ISOTOPE, QVAL, TYPE, RESNAM`<br>$$$ NOTE: QVAL is omitted for kinematic scales. |
| LINE F.5 (*) | : | Lineshape parameters. Requires the number parameters to the lineshape `TYPE`. |

The line below is only required if an external resolution function, whose `RESNAM` does not match any `PKNAME`, is requested.

LINE F.6 (*) : `RESNAM`, resolution parameters.

The line below is repeated for each peak.

LINE F.7 (*) : Vary codes.

**2. Input using `LEVELS`**

LINE F.1 (*) : `ITSVCS, CONV, CHIVCS, CHID, DCHID`
LINE F.2 (*) : Background parameters (requires 10).
LINE F.3 (*) : Background vary codes, followed by `IBACK`.

The next section of input describes a set of default lineshapes. The lineshapes are not actual peaks, and thus do not require height or position parameters. Each actual peak will be assigned a shape from this list and given height and position parameters. It is necessary to describe each independent shape. External resolution functions, which do not change shape during the fit, can also be supplied here. Extra shapes, not used by the fit, can also be included for editing convenience.

The set of default lineshapes is handled in much the same way that peaks were handled by the early version of `ALLFIT`. Initial parameters entered as zero default to the value of the corresponding parameter in the first shape of the same type. Constraints within the set of shapes are specified by vary codes. The shape index is used to lock shape parameters together. These vary codes will be re-interpreted after the peaks are read.

LINE F.4 (*) : Number of shapes. The set of lines 5-7 are repeated for each shape.
LINE F.5 (*) : Shape name, shape type.
LINE F.6 (*) : Shape parameters, not including height or position.
LINE F.7 (*) : Shape vary codes.

The next input section describes the peaks; one line per peak is required. Locked height or position is specified using a vary code equal to a preceding peak number. Adding a counter at the end of the line can help set up these constraints. Provided that all of the entries required for each peak are supplied, this counter is not read.

LINE F.8 (*) : `PKNAME`, `SHAPE`, `RESNAM`, height, height vary-code, position vary-code.

### G. Additional Vary Codes

If additional sets of vary codes are desired (`NOVCS`>1), the input described in this section is read; otherwise this section is skipped. It is only necessary to supply vary codes for those peaks whose codes are to change. Line G.4 is repeated for each change.

LINE G.1 (*) : `ITSVCS, CONV, CHIVCS, CHID, DCHID`
LINE G.2 (*) : Background vary codes.
LINE G.3 (*) : Number of peaks with new vary codes.
LINE G.4 (*) : PKNAME, vary codes. At present, ten vary codes are required for each peak, even if the lineshape type uses fewer parameters. The extra entries are ignored, or can be omitted.

## VI. AUXILIARY FILES

### A. Levels File

For most applications involving many peaks, it usually convenient to specify the characteristics of nuclear states using a levels file, which is assigned to `LUN3`. Each level is described by a single line of input. The file can contain an unlimited number of entries and is searched for entries matching the names in the primary input file. Thus, one can include more levels than are needed for a particular spectrum and can, if desired, include a couple of versions with slightly different labels.

(*) : `LEVEL` = name of level
(*) : `ISOTOPE` = name of isotope to which level belongs
(*) : `QVALUE` = reaction $Q$-value in MeV
(*) : `WIDTH` = intrinsic width $\Gamma$ in MeV
(*) : `THRESH` = threshold $Q_t$ in MeV

### B. Target File

Although the generic version obtains target information from the data file itself, many specialized versions obtain that information from a file of target descriptions instead. Therefore, although the generic version does not use it, the subroutine `allfit_rtarg` is supplied within the `ALLFIT` library. Custom versions should call this subroutine or an equivalent alternative from `DREAD`.

The target file contains the target thickness and density and describes the isotopic composition. The file may contain many entries, one following the other. Each target entry should have the format below. Line 3 is repeated for each element in each target.

LINE 1 (*) : Target name.
LINE 2 (*) : `NELS` = number of elements
`THICK` = areal thickness in mg/cm$^2$
`DENS` = density in g/cm$^3$
LINE 3 (*) : Isotope label, Z, A, fractional abundance.

### C. External Background File

The data, $y_B(i)$, for construction of the external background contribution, $B_{ext}$, described in Sec. II D is read from `LUN4` by subroutine `READB` found in file `allfit_iofiles`. The standard version of this file contains three columns containing

$$i \quad y_B(i) \quad \delta y_B(i)$$

where $1 \leq i \leq$ `MNPTS` is the channel number and $\delta y_B(i)$ is the uncertainty in $y_B(i)$. `READB` may be modified to accomodate different file formats.

### D. Plot File

Visual inspection usually provides the most important test of fit quality, but because plotting packages remain quite dependent upon both platform and user preferences there exists no standard format for output. Many output formats and plotting packages have been employed over the history of the program. For the standard distribution package I have chosen a very simple format with the following structure.

**(1)** the name of the data file

**(2)** the number of histgrams

**(3)** for each histogram:

- **(a)** a label
- **(b)** the number of points
- **(c)** 2- or 3-column listing of { `EXCO(I), Y(I), DY(I), I=1,NPTS` }

Here `EXCO(I)` is the excitation energy at channel `I` for the reference reaction, `Y` is the yield for either the complete spectrum or some component of the model, and `DY` is the uncertainty in that yield.

The histograms are listed in the following order.

**(1)** data

**(2)** total fit

**(3)** total background

**(4)** external background (if present)

**(5)** continuum background (if different from total)

**(6)** individual peaks atop background (`NPKS` entries)

The data histogram contains three columns while all remaining histograms contain only two.

The distribution package includes a command macro `plotfit.pcm` in the `SAMPLES` directory which can be used with the `PHYSICA` plotting package from TRIUMF to display fit results either on a graphics screen or as a postscript file. At the `PHYSICA` prompt, simply type `plotfit.pcm plotfile` where `plotfile` is the name of the plot file. The macro will display all components of the fit and uses autoscaling for painless viewing. The name of a postscript file is requested for hardcopy purposes and may be omitted. The default format produces a semi-log plot, but linear plots can be obtained simply by changing the value of `YLOG` from 10 to 1. Similarly, one may wish to change scales or add labels for publication quality output and may easily suppress various contributions, such as the individual lineshapes, if desired.

Of course, custom versions may employ different output formats and plotting software. Modify `PLOT` and `PLTOUT` in `allfit_iofiles` accordingly.

### E. Summary File

Because most experiments produce a large number of spectra, it is convenient to have an automated procedure for compiling and processing the results of many fits. Although one could use the main output file itself, that file is rather large and its variability in structure complicates processing by secondary programs. Therefore, the subroutine `SUMMARY` contained in `allfit_iofiles` is intended to write a shorter and easily readable summary of the results in a form that can be used for sorting programs. As with plot output, many formats have been employed over the history of the program, some formatted and others unformatted with varying levels of detail. The standard version provides a prototype which reproduces the most important parts of the main output file in a shorter summary file. This summary file contains the file names, the parameters required to normalize cross section, selected fit diagnostics, and the table of final results.

For example, for proton-nucleus scattering experiments performed at IUCF and LAMPF I have written sorting programs which process a large number of fit results to produce tables and plots of the angular distributions of cross section and analyzing power for every final state observed. Similarly, for electron-scattering experiments performed at MIT-Bates I wrote a sorting program which produced form factor tables and plots. For each of these applications a custom version of `SUMMARY` was used to supply data to a sorting program. The sorting programs can also apply corrections for changes in target or experimental parameters, normalization factors, etc., and identify fits with substandard diagnostics. Although some of these custom versions could still be provided upon request, many of the facilities for which they were written are no longer operational. Nevertheless, I anticipate that sorting programs will become available for the analysis of data at newer facilities.

## VII. PRACTICAL CONSIDERATIONS

In most experiments, a large number of spectra are collected with similar targets and kinematic conditions. The basic characteristics of these spectra then tend to be quite similar. Considerable care should be paid to setting up one or more prototype input files that can be used as templates for systematic fitting of related spectra. These templates can be copied into other input files and the necessary changes are then made by editing. If the levels option is selected, which is recommended for almost all applications, the editing usually consists only of changes to file names, initial heights, and possibly kinematic calibrations. To minimize the number of changes, it is sometimes desirable to reserve a spot for small peaks even when they seem to be absent from a particular spectrum. If such peaks are too small to fit, their heights can simply be fixed to negligible cross sections. This may cost a little computation time, but it is still faster than renumbering peaks. Alternatively, peaks which are expected to be of marginal significance can be collected together at the end of the list of peaks. Also, I prefer to group peaks together by isotope rather than in order of missing mass. Note that the order that peaks are entered in the input file is immaterial and can be chosen for convenience.

Before beginning a major analysis project, some thought should be given to establishing a consistent naming convention for the peak labels used in the `ALLFIT` input file and the `LEVELS` file. After a number of summary files have been compiled, a sorting and tabulation program is often used to compile the results into convenient tabular form and to produce files suitable for plotting programs. Sorting and tabulation is usually based upon the peak labels.

I recommend the following conventions. If the multipolarity of the peak is known, convenient names take forms similar to `34S.2P1` for the first $2^+$ state of $^{34}$S or `18O.1M4` for the $1^-_4$ (fourth $1^-$) state of $^{18}$O. Similarly, if the spin is half-integral the names can use twice the spin, *e.g.* `9BE.5M1` for the first $\frac{5}{2}^-$ state of $^9$Be. Alternatively, energy levels can be used for unknown multipolarities, such as `18O9.36` for a state in $^{18}$O near 9.36 MeV. The tabulation programs I use produce plot files with `VMS` names constructed by appending a prefix to the state label. For example, a cross section file might have the name `SIG18O.1M4` or `SIG18O9.36` and an analyzing power file might have the name `AY9BE.5M1`.

It is generally advantageous to fit as large a portion of the spectrum as possible with a single run of `ALLFIT`. The background is then strongly constrained and many peaks are available to determine the common lineshape parameters. Initially, the fit should be given only as much freedom as is absolutely necessary in the first set of vary codes. As the fitting function improves, some of the constraints can be relaxed in subsequent sets of vary codes.

For complicated spectra, it is often helpful to examine the starting point of the fit by performing an initial fit with 0 iterations. Obvious errors should then become apparent before much time is wasted. Similarly, fits with 0 iterations can reveal a problem if an unexpected crash is encountered.

The fit code will use any freedom it is given even if the data are not sufficient to determine all of the fitting parameters. Therefore, it can be dangerous to give the fit too much freedom. The parameters which most commonly cause trouble are the tail parameters and the asymmetry. Undetermined parameters result in unreasonably large uncertainties in the lineshape parameters and can interfere with error analysis. The most common symptom of this problem is the tendency of cross section uncertainties to become much larger than statistical uncertainties even for isolated peaks.

The left tail in many experiments, such as $(e, e')$ at MIT-BATES or NIKHEF and $(p, p')$ at HRS, is often almost absent. The slope parameter $\lambda_L$ is then apt to become unreasonably small and may cause the program to crash. Alternatively, the calculated peak area may become artificially inflated, even if $f_L$ is small, if $\lambda_L$ becomes so small that the left tail becomes a pedestal. Common symptoms of this problem are large area/sum ratios or large relative/statistical error ratios. The solution is to fix $f_L$ to a small value, such as $10^{-5}$, and $\lambda_L$ to a large value, such as 10. The left tail will then be suppressed quite effectively.

Similarly, it is usually not possible to determine the right tail when the theoretical radiative tail is employed. The radiative tail tends to overwhelm the right tail of the resolution function. The solution is to suppress the tail as described above.

## VIII. SAMPLE CASES

The distribution package provides 3 sample cases. Each case is represented by a set of files:

| | |
|---|---|
| case.inp | main input file |
| case.dat | data file |
| case.out | main output file |
| case.plt | plot file |
| case.sum | summary file |

where the available cases are `pp_sample1`, `pp_sample2`, and `ee_sample1`. In addition, all samples read their state descriptions from the same `levels` file. Finally, `ee_sample1.bkgd` is an external background file. The figures in this section were made with the `PHYSICA` command macro `plotfit.pcm`.

### A. `pp_sample1`

Fig. 1 shows a fitted spectrum for proton scattering from a beryllium oxide (BeO) target acquired at IUCF using the QDDM spectrometer [6]. The beam energy is 135 MeV and the scattering angle is 30°. The lowest strong peak corresponds to elastic scattering from $^{16}$O and the next to $^{9}$Be. These elastic peaks clearly illustrate the standard lineshape with its asymmetic hypergaussian central regions and its exponential tails. The abrupt increase in the background at a threshold approximately 1.665 MeV above the ground state of 9Be represents the break-up continuum and illustrates the use of the piecewise continuous background. A couple of broad states in 9Be are shown also, with the lowest of these being rather asymmetric due to its proximity to the continuum threshold. Further details about the analysis of spectra of this type can be obtained from Refs. [6–8].

### B. `pp_sample2`

Fig. 2 shows a fitted spectrum for BeO$(p, p')$ that covers a higher excitation energy region than `pp_sample1` but which was acquired under otherwise identical conditions. In this case a strong broad state in 9Be is represented by a Lorentzian lineshape convoluted with a resolution function based upon the narrow peaks. Further details about the analysis of spectra of this type can be obtained from Refs. [6–8].

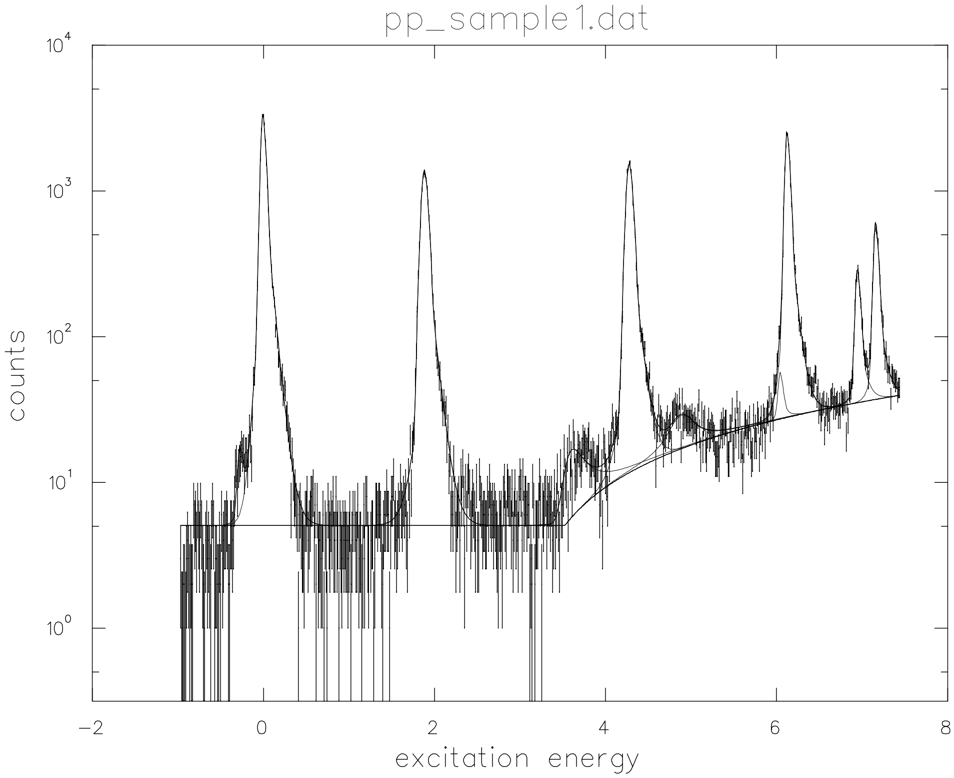

FIG. 1. Sample fit to a spectrum of protons scattered from a BeO target acquired with the QDDM spectrometer at IUCF using a beam energy of 135 MeV and scattering angle of 30°

### C. ee_sample1

Fig. 3 shows a fitted spectrum for electron scattering from a BeO target that was acquired at MIT-Bates using the ELSSY spectrometer [3]. The beam energy was 230 MeV and the scattering angle 160°. This spectrum covers excitation energies between approximately 10 and 24 MeV relative to $^{16}$O. In this region the discrete excitations are found astride a complicated background consisting of continua for both 9Be and $^{16}$O. The 9Be continuum was analyzed by using a spectrum acquired under the same conditions for a pure beryllium target. A fit of the beryllium spectrum was performed and the narrow peaks were subtracted from the data. The resulting continuum spectrum was then smoothed and written into the external background file. The external background was supplied as one component of the fit to the BeO spectrum, with a scale factor to adjust for normalization and a shift to account for differences in energy loss across targets of different thickness. Although not done for this example, the uncertainty in the continuum can be propagated into the uncertainty in the cross sections for fitted peaks. In addition, the smooth part of the $^{16}$O continuum was represented by a polynomial beginning at the neutron emission threhold. These two curves are clearly visible in Fig. 3. A large number of peaks, some narrow and some broad, were convoluted with the theoretical radiative tail and fit to the data using whatever constraints were available. Thus, this example illustrates a rather complex analysis. Further details may be found in Refs. [10,3,2,4].

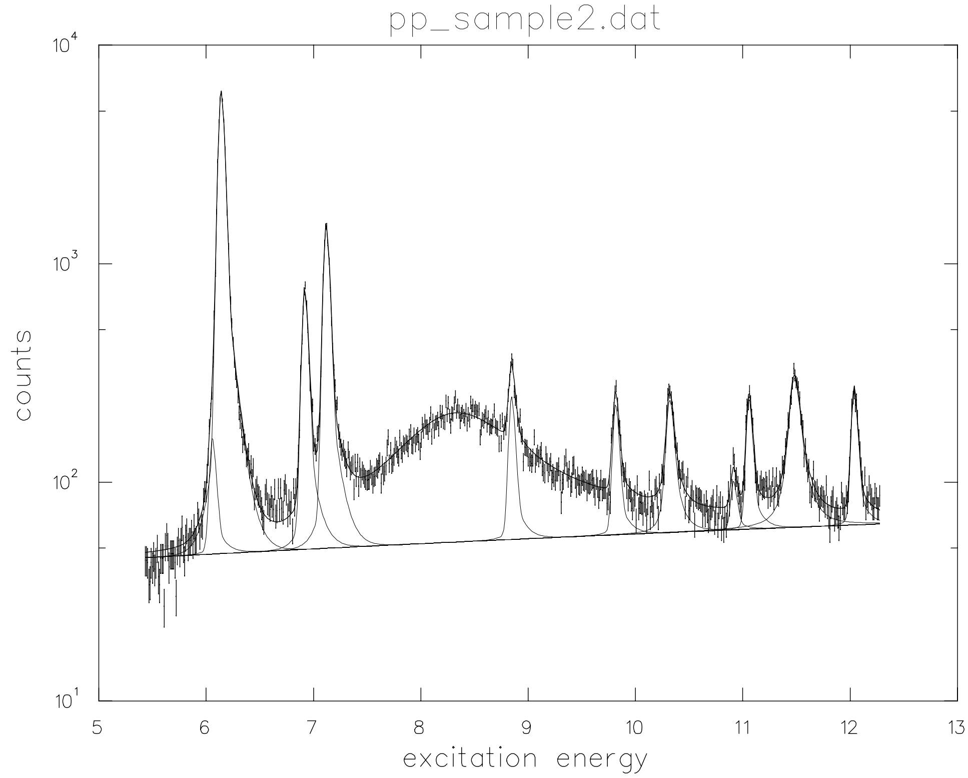


FIG. 2. Sample fit to a spectrum of protons scattered from a BeO target acquired with the QDDM spectrometer at IUCF using a beam energy of 135 MeV and scattering angle of 30°

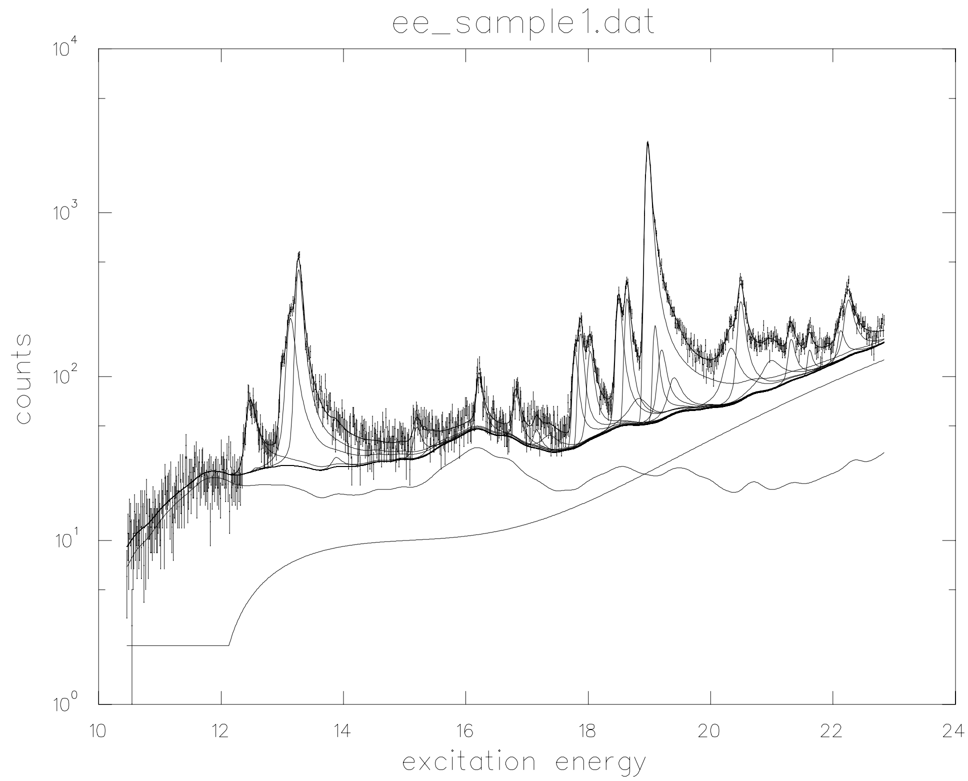


FIG. 3. Sample fit to a spectrum of electrons scattered from a BeO target acquired with the ELSSY spectrometer at MIT-Bates using a beam energy of 230 MeV and scattering angle of 160°

# REFERENCES


[1] J. J. Kelly, Ph.D. thesis, MIT, 1981.
[2] T. N. Buti *et al.*, Phys. Rev. **C 33**, 755 (1986).
[3] C. E. Hyde-Wright *et al.*, Phys. Rev. **C 35**, 880 (1987).
[4] D. M. Manley *et al.*, Phys. Rev. **C 35**, 880 (1986).
[5] J. P. Glickman *et al.*, Phys. Rev. **C 43**, 1740 (1991).
[6] J. J. Kelly *et al.*, Phys. Rev. **C 39**, 1222 (1989).
[7] S. Dixit *et al.*, Phys. Rev. **C 43**, 1758 (1991).
[8] H. Seifert *et al.*, Phys. Rev. **C 47**, 1615 (1993).
[9] P. R. Bevington, *Data Reduction and Error Analysis for the Physical Sciences* (McGraw-Hill, N.Y., 1969).
[10] C. E. Hyde-Wright, Ph.D. thesis, MIT, 1984.

## APPENDIX A: SPECIALIZED VMS VERSIONS

Several versions of ALLFIT have been implemented at the University of Maryland. A version applicable to data collected on the old QDDM spectrometer of IUCF is called qddm_fit and uses the library qddm_lib. A version called k600_fit handles newer data collected on the K600 spectrometer of IUCF and employs the application library k600_lib. Similarly, hrs_fit uses hrs_lib to analyze data collected on the HRS spectrometer at LAMPF. Prototype versions for the ELSSY and MEPS spectrometers at BATES are also available.

The K600 and HRS data are collected with the LAMPF acquisition system called Q. Their data files are written according to the HSV protocol. ALLFIT reads these files using subprograms found in the library hsv_lib. Because the Q system is generally used on VMS machines, the VAX extensions to FORTRAN are used liberally throughout the HSV library.

### 1. K600

The K600 version obtains the histogram and experimental parameters from an HSV file. In addition, a data base can be used in place of the PRM array to supply some of the experimental parameters. This option is useful whenever HSV files are made before all of the calibrations have been completed. It also provides a convenient means of altering PRM values without repeating the replay. If a file name is supplied for the data base, it will be used instead of the PRM array. However, the run number must still be supplied by the PRM array.

The HSV histograms are frequently too long for convenient processing. Therefore, we can omit the first ICHAN0 channels and rebin the remainder NBIN:1. Subsequent references to channels employ the new, shifted and rebinned, channels beginning at 1.

| | | |
|---|---|---|
| LINE 1 (A) | : | Histogram name. |
| LINE 2 (*) | : | ICHAN0, NBIN |
| LINE 3 (A) | : | File name for data base. |
| LINE 4 (A) | : | LOGBOOK name. |

Additional information describing the histograms and normalizations is obtained from a LOGBOOK file. This file is designed to provide maximum flexibility in the organization of scalers and tests. The first line gives a spectrometer offset to be added to the spectrometer angle read from either the PRM array or from the data base. The next line specifies the number of tests used to define spectra. For each of these tests, we read the test number, the number for the gate that determines angular acceptance, and the polarization state. It is not necessary to specify histogram names in the logbook.

| | | | |
|---|---|---|---|
| LINE 1 (*) : | spectrometer offset | | |
| LINE 2 (*) : | number of histogram tests. Line 3 is repeated for each histogram. | | |
| LINE 3 (*) : | test number | | |
| | gate number for scattering angle bin | | |
| | polarization | = 'U' $\rightarrow$ unpolarized | |
| | | = 'N' $\rightarrow$ normal | |
| | | = 'R' $\rightarrow$ reverse | |

The next group of entries specifies locations at which experimental parameters reside in the `PRM` array and in the data base. Each entry requires its own line containing a `PRM` location and a data base location.

LINE 4.1 (*) : run number
LINE 4.2 (*) : experiment number
LINE 4.3 (*) : aperture number
LINE 4.4 (*) : target number
LINE 4.5 (*) : integrator scale
LINE 4.6 (*) : spectrometer angle
LINE 4.7 (*) : beam energy
LINE 4.8 (*) : target angle
LINE 4.9 (*) : spin-up polarization
LINE 4.10 (*) : uncertainty in spin-up polarization
LINE 4.11 (*) : spin-dn polarization
LINE 4.12 (*) : uncertainty in spin-dn polarization
LINE 4.13 (*) : offset for title

The next group of entries specifies addresses at which scalar data may be found. Each entry requires a separate line containing addresses for unpolarized, normal, and reverse scalers.

LINE 5.1 (*) : clock scalar (T, N, R)
LINE 5.2 (*) : triggers
LINE 5.3 (*) : events
LINE 5.4 (*) : left integrator
LINE 5.5 (*) : left integrator not busy
LINE 5.6 (*) : right integrator
LINE 5.7 (*) : right integrator not busy

The next group of entries specifies addresses at which test results may be found. Each entry requires a separate line containing test numbers for total, normal, and reverse cycles.

LINE 6.1 (*) : particle identification
LINE 6.2 (*) : test counter
LINE 6.3 (*) : good particles
LINE 6.4 (*) : good patterns
LINE 6.5 (*) : multiple hits

The solid angle is determined by both the physical aperture and by a gate placed upon the scattering angle spectrum. The aperture must be chosen from the list within the logbook file. For each aperture, we read an identification number, a character description, the horizontal acceptance, the vertical acceptance, and the nominal solid angle in msr. If the vertical acceptance is zero, the aperture is assumed to be circular and its diameter to be specified by the horizontal acceptance. The acceptances and the distance between target and aperture must be supplied in the same units. If the gate is wider than the aperture, the nominal acceptance is used. Otherwise, the acceptance is computed from the geometrical parameters.

LINE 7 (*) : number of apertures, distance to aperture. Line 8 is repeated for each aperture.
LINE 8 (*) : aperture number
aperture name
horizontal acceptance
vertical acceptance
solid angle (msr)

The target number is also drawn from a list within the logbook file. This list associates the identification number with the character name used in the target file. The name of the target file is also read from the logbook file. This file, in standard `ALLFIT` format, then completes the description of the target.

LINE 9 (*) : number of targets. Line 10 is repeated for each target.
LINE 10 (*) : target number
target name
LINE 11 (*) : target file name

**2. HRS**

The HRS version obtains the histogram and experimental parameters from an `HSV` file. The `HSV` histograms are frequently too long for convenient processing. Therefore, we can omit the first `ICHANO` channels and rebin the remainder `NBIN`:1. Subsequent references to channels employ the new, shifted and rebinned, channels beginning at 1. Additional information describing the histograms and normalizations is obtained from a `LOGBOOK` file. The target description is obtained from a target file in standard `ALLFIT` format.

LINE 1 (A) : `EXPT` = experiment label, such as `'HRS'`
`BRIEF` = `.TRUE.` → short list of experimental parameters
= `.FALSE.` → detailed list of experimental parameters
LINE 2 (A) : Histogram name.
LINE 3 (*) : `ICHANO, NBIN`
LINE 4 (A) : `LOGBOOK` name.
LINE 5 (*) : Target label, target file name.

The logbook file provides efficiency and normalization information and directs interpretation of the `HSV` file. The first line gives the analyzing powers and uncertainties for the polarimeters in lines B and C. Next, we read parameters that describe the position dependence of the detector efficiency such that

$$eff(x) = \mathtt{AEFF} + \mathtt{BEFF} \times x.$$

If `AEFF`=0., we assume that the efficiency is unity everywhere. The beam charge is calibrated using an ionization chamber. The parameter `FNORM`, supplied for each histogram, normalizes the product of beam current and solid angle.

| | | |
|---|---|---|
| LINE 1 (*) | : | BLCAY, DBLCAY, BLBAY, DBLBAY |
| | | BL×AY = analyzing power of polarimeter in beam-line * |
| | | DBL×AY = uncertainty in BL×AY |
| LINE 2 (*) | : | AEFF, BEFF |
| LINE 3 (*) | : | number of histograms listed. Lines 4-5 are repeated for each histogram. |
| LINE 4 (A6) | : | Histogram name. |
| LINE 5 (*) | : | FNORM : normalization factor; |
| | | OFFSET : spectrometer angle offset in degrees, such that true THETA = set THETA + OFFSET; |
| | | POL = 'NORMAL' or 'REVERSE' beam orientation. |

The next group of entries specifies addresses in the PRM array at which the experimental parameters are to be found. Each parameter requires its own line.

| | | |
|---|---|---|
| LINE 6.1 (*) | : | run number |
| LINE 6.2 (*) | : | experiment number |
| LINE 6.3 (*) | : | spectrometer angle |
| LINE 6.4 (*) | : | incident energy |
| LINE 6.5 (*) | : | target angle |
| LINE 6.6 (*) | : | pointer to title (precedes start) |

The next group of entries specifies addresses for the scalers. Each scalar type requires its own line. For the first three entries, three addresses are required for normal (N), reverse (R), and quench (Q) cycles. The remainder do not need quench.

| | | | |
|---|---|---|---|
| LINE 7.1 (*) | : | beam monitor | (N, R, Q) |
| LINE 7.2 (*) | : | beam gates | (N, R, Q) |
| LINE 7.3 (*) | : | events | (N, R, Q) |
| LINE 7.4 (*) | : | triggers | (N, R) |
| LINE 7.5 (*) | : | taped triggers | (N, R) |
| LINE 7.6 (*) | : | LCPO left | (N, R) |
| LINE 7.7 (*) | : | LCPO left accidental | (N, R) |
| LINE 7.8 (*) | : | LCPO right | (N, R) |
| LINE 7.9 (*) | : | LCPO right accidental | (N, R) |
| LINE 7.10 (*) | : | LCPO up | (N, R) |
| LINE 7.11 (*) | : | LCPO up accidental | (N, R) |
| LINE 7.12 (*) | : | LCPO down | (N, R) |
| LINE 7.13 (*) | : | LCPO down accidental | (N, R) |
| LINE 7.14 (*) | : | LBPO left | (N, R) |
| LINE 7.15 (*) | : | LBPO left accidental | (N, R) |
| LINE 7.16 (*) | : | LBPO right | (N, R) |
| LINE 7.17 (*) | : | LBPO right accidental | (N, R) |
| LINE 7.18 (*) | : | LBPO up | (N, R) |
| LINE 7.19 (*) | : | LBPO up accidental | (N, R) |
| LINE 7.20 (*) | : | LBPO down | (N, R) |
| LINE 7.21 (*) | : | LBPO down accidental | (N, R) |

The next group of entries specifies the addresses at which the necessary test results may be found. Each entry requires a separate line with values for normal and reverse cycles.

| | | | | |
|---|---|---|---|---|
| LINE 8.1 | (*) | : | test counter | (N, R) |
| LINE 8.2 | (*) | : | PID | (N, R) |
| LINE 8.3 | (*) | : | good particles | (N, R) |

Finally, it is possible to replace the LCP0 beam polarization with a data base entry by supplying optional Lines 9-10.

| | | | |
|---|---|---|---|
| LINE 9 | (A) | : | file name for data base |
| LINE 10 | (*) | : | entry number for beam polarization |
| | | : | entry number for its uncertainty. |

This option is useful for experiments requiring nonstandard polarization calculations.

## 3. QDDM

The QDDM version handles data taken on the old QDDM spectrometer at IUCF. The spectrum is assumed to occupy 1024 channels. The spectra are read from unformatted sequential files with fixed length records of 512 bytes. Two data formats are available:

`FORM` = ‘`DATA2`’ → 256 `I2` integers per record;
`FORM` = ‘`DATA4`’ → 128 `I4` integers per record.

The experimental parameters are read from a logbook file. The only line read by `DREAD` from `LUN5` bears the name of the logbook file:

LINE 1 (A) : `LOGBOOK` name.

The logbook file has the format shown below. Lines 3-5 are repeated for each data file.

| | | | |
|---|---|---|---|
| LINE 1 (A) : | Target file name. | | |
| LINE 2 (A) : | `FORM` | | |
| LINE 3 (A) : | Name of data file, including device and directory. | | |
| LINE 4 (A) : | Target label. | | |
| LINE 5 (A) : | `E0` | = incident energy | (MeV) |
| | `THETAL` | = lab scattering angle | (deg) |
| | `TANGLE` | = target angle | (deg) |
| | `SPECTH` | = horizontal acceptance | (half-angle in mr) |
| | `SPECTV` | = vertical acceptance | (half-angle in mr) |
| | `CLICKS` | = integrator clicks | |
| | `FS` | = integrator full-scale | (nA) |
| | `DTC` | = dead-time correction. | |

## 4. ELSSY

The ELSSY version handles data taken on the ELSSY spectrometer at MIT-BATES. This version originated at the same time the author was performing QDDM experiments and is rather similar in style to the `QDDM` version described above.

The spectrum is assumed to occupy 1024 channels and is read from an unformatted sequential file with fixed length records of 512 bytes. Two data formats are available:

`FORM` = ‘`DATA2`’ $\rightarrow$ 256 `I2` integers per record;
`FORM` = ‘`DATA4`’ $\rightarrow$ 128 `I4` integers per record.

The experimental parameters are read from a logbook file. The only line read by `DREAD` from `LUN5` bears the name of the logbook file:

LINE 1 (A) : `LOGBOOK` name.

The logbook file has the format shown below. Lines 3-5 are repeated for each data file.

| | | | |
|---|---|---|---|
| LINE 1 (A): | Target file name. | | |
| LINE 2 (A): | `FORM` | | |
| LINE 3 (A): | Name of data file, including device and directory. | | |
| LINE 4 (A): | Target label. | | |
| LINE 5 (A): | `E0` | = incident energy | (MeV) |
| | `THETAL` | = lab scattering angle | (deg) |
| | `TANGLE` | = target angle | (deg) |
| | `SPECTH` | = horizontal acceptance | (inches) |
| | `SPECTV` | = vertical acceptance | ( ” ) |
| | `CLICKS` | = integrator clicks | |
| | `FS` | = integrator full-scale | ($\mu A$) |
| | `DTC` | = dead-time correction. | |

## 5. MEPS

The MEPS version handles data taken on the MEPS spectrometer at MIT-BATES. This is a simple prototype version that is not intended to be final. The spectrum is assumed to occupy 1024 channels and is read from an unformatted sequential file with fixed length records of 512 bytes. Two data formats are available:

`FORM` = ‘`DATA2`’ $\rightarrow$ 256 `I2` integers per record;
`FORM` = ‘`DATA4`’ $\rightarrow$ 128 `I4` integers per record.

The experimental parameters are read from a logbook file. The only line read by `DREAD` from `LUN5` bears the name of the logbook file:

LINE 1 (A) : `LOGBOOK` name.

The logbook file has the format shown below. Lines 3-5 are repeated for each data file.

| | | | | |
|---|---|---|---|---|
| LINE 1 (A) | : | Target file name. | | |
| LINE 2 (A) | : | `FORM` | | |
| LINE 3 (A) | : | Name of data file, including device and directory. | | |
| LINE 4 (A) | : | Target label. | | |
| LINE 5 (A) | : | `E0` | = incident energy | (MeV) |
| | | `THETAL` | = lab scattering angle | (deg) |
| | | `TANGLE` | = target angle | (deg) |
| | | `ACC` | = acceptance | (msr) |
| | | `CLICKS` | = integrator clicks | |
| | | `FS` | = integrator full-scale | ($\mu A$) |
| | | `DTC` | = dead-time correction. | |