ORNL/TM-8720

ornl

OAK RIDGE NATIONAL LABORATORY

UNION CARBIDE

A User's Manual for the FERDO and FERD Unfolding Codes

Bert W. Rust
Daniel T. Ingersoll
Walter R. Burrus

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A06; Microfiche A01

ORNL/TM-8720

Engineering Physics Division

A USER'S MANUAL FOR THE FERDO AND FERD UNFOLDING CODES

Bert W. Rust*
Daniel T. Ingersoll
and
Walter R. Burrus†

Manuscript Completed - August 1983

Date Published - September 1983

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
U.S. DEPARTMENT OF ENERGY
Under Contract No. W7405-eng-26

*United States Department of Commerce, National Bureau of Standards, Washington, D.C.

†Science Applications, Inc., Oak Ridge, Tennessee.

CONTENTS

LIST OF FIGURES

ABSTRACT

FERDO and FERD are unfolding codes which can be used to correct observed pulse-height distributions for the nonideal response of a pulse-height spectrometer or to solve poorly conditioned linear equations. It is assumed that the response of the spectrometer is given by $\underline{\underline{A}}\underline{x} = \underline{b}$, where $\underline{\underline{A}}$ is the spectrometer response function matrix, $\underline{x}$ is the unknown spectrum, and $\underline{b}$ is the pulse-height distribution. FERDO does not solve directly for x, but instead solves for $\underline{p} = \underline{\underline{W}}\underline{x}$, where $\underline{\underline{W}}$ is a "window function matrix." Typically, $\underline{\underline{W}}$ is the resolution function of an ideal spectrometer which has a single Gaussian response. The effective resolution of the unfolding solution may be varied by choice of $\underline{\underline{W}}$. Confidence intervals ($p^{lo}$, p^{up}) are found for each element of the solution p. FERDO and FERD are written in IBM 360/370 Fortran (Level H). This manual describes and derives the mathematical procedures used by the codes, tells how to write input data for them, and gives solutions to some sample problems to illustrate the output.

1. INTRODUCTION

This user's manual for FERDO and FERD aims to: (1) describe the basic mathematical machinery needed to understand what the codes do (i.e. the problems that they solve), (2) document the mathematical procedures actually used by the two codes, (3) explain how to set up data to run the codes, and (4) explain how to interpret the various output quantities. Many users may not want to get into all of the gritty details of (2), but all users should be familiar with (1) before attempting to operate with (3) and (4). Chapters 2-8 cover (1), perhaps more thoroughly than is necessary for some users. The gritty details of (2) are contained in Chapters 8-11 which should be read in order. Chapters 12 and 13 cover (3) and (4), but when the user begins to become familiar with the output he will find Chapter 11 is also useful for interpreting some of the various output quantities. Users who are contemplating making changes in either of these codes will definitely want to read Chapter 11.

The methods described in this report were originally developed by Walter R. Burrus and his coworkers. Various versions of the codes have enjoyed great success throughout the world in spite of the lack of coherent documentation of either the methods or the codes. The guiding philosophy and underlying mathematics have been extensively discussed in Rust and Burrus (1972), and the mathematical theory of the FERDO and FERD methods has been recently discussed by Burrus et al (1980), but neither of these works gives an adequate description of the two codes actually being used to solve unfolding problems.

The earliest descriptions of these methods were given at the time they were being developed in various brief papers and presentations (Burrus, 1960, 1961a, 1961b, 1962). In 1961 Burrus wrote a well-organized overview of this work in the form of an unpublished Ph.D. thesis draft. He and his collaborators continued to develop the methods with striking success, cf. Bogert and Burrus (1962), Burrus (1963), and Burrus and Verbinski (1965). In 1965 he wrote a new Ph.D. thesis (Burrus, 1965) which summarized the method

and gave several applied examples. The main result of this work was the computer code FERDO and its many variants. These include FERDOR, SLOP, and COOLC which have been described in Burrus and Verbinski (1965, 1969), Kendrick and Sperling (1970), Rust and Burrus (1971), Peelle (1971), Burrus (1976), and Rust (1976). The version of FERDO described in this report has been updated in many ways, especially in the input/output sections, but except for a few minor points, the mathematical method is still the same one used in the earlier versions.

The FERD code has not been so widely used as FERDO, and almost no documentation for it has been published. The mathematical foundations and the philosophy of the method have been discussed in Rust and Burrus (1972) and Burrus (1976), but it was not explicitly described in those works. A very brief description of the code was given in Peelle (1971) and the mathematical methods was briefly described in Burrus et al. (1980). This report gives a more complete description, but the reader is advised to carefully read section 11.4 as well as Chapter 10 in order to see how the method actually works. The version of FERD given here also has been updated, but again the mathematics is essentially the same as in the older versions. The input is designed to be almost identical to that required by FERDO so that a user can easily run the same data through both codes.

2. THE MATHEMATICAL PROBLEM

The FERDO and FERD programs are designed to solve radiation spectrum unfolding problems in which the true neutron or gamma spectrum $x(E)$ is related to the measured pulse height spectrum $\bar{b}_i$ by an integral equation of the form

$$\int_{E_{lo}}^{E_{up}} A_i(E)x(E)dE = \bar{b}_i + \varepsilon_i \ , \quad i=1,2,\cdots,m, \tag{2-1}$$

where E represents energy, i represents pulse height bin number, and $A_i(E)$ is the response function of the instrument, i.e., the quantity $A_i(E)dE$ is the probability that a neutron or gamma ray with energy in the range $(E \pm \frac{1}{2}dE)$ will produce a count in bin number i.

The limits of integration are shown in this paper as E_{lo} and E_{up}. In radiation spectrometers, the physical limits are set by the energy range of the radiation and the range of the non-zero portion of the response function. For neutron and gamma radiation which is associated with spontaneous or reactor induced processes, the physical energy range of the radiation is typically 0 to around 10 MeV. In the case of cosmic radiation, the upper radiation limit may be very high. When formulating the integral equation for numerical solution, the lowest energy portion of the solution is frequently divergent or very uncertain. Thus, the lowest energy which contributes significantly to the viable solution depends upon the details of the spectrometer and ranges from a few tenths of an eV for low energy absorption filter type spectrometers to perhaps a few 10's of Kv for typical scintillation spectrometers. Usually there is a reasonable choice for E_{up} for which there is negligible contribution to the b_i components.

The quantity ε_i in the above relation is the stochastic counting error, and it is assumed that the counting errors are independent of one another and are normally distributed with zero means. Using the symbols E to denote the expectation operator, σ to denote standard deviation, Var to denote variance, and ρ_{ij} to denote the correlation between ε_i and ε_j, we can write

$$E(\varepsilon_i) = 0 \ , i = 1,2,\cdots,m, \tag{2-2a}$$

$$\mathrm{Var}(\varepsilon_i) = E(\varepsilon_i^2) = \sigma_i^2 \ , i = 1,2,\cdots,m, \tag{2-2b}$$

$$\rho_{ij} = E(\varepsilon_i \varepsilon_j) = 0, \ i \neq j, \ i,j=1,2,\cdots,m, \tag{2-2c}$$

and the probability distribution function for each ε_i is given by

$$f(\varepsilon_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left[- \frac{\varepsilon_i^2}{2\sigma_i^2} \right], \ i=1,2,\cdots,m. \tag{2-3}$$

When the error term ε_i is not present, integral equations of the form of (2-1) are called Fredholm integral equations of the first kind. It is well known that in general such equations do not determine a unique solution $x(E)$. When the error is included the situation is even worse, and there will always be many possible very different functions $x(E)$ which satisfy the equation within the limits set by the statistical error. Of course, most such solutions can be ruled out by physical considerations, but even so, the presence of small errors in the b_i allows wide variations in the set of physically plausible solutions. Part of the problem is that a given finite set of b_i cannot determine a unique function $x(E)$ at an infinite number of points, but even if one adopts a less ambitious course of action and seeks only to estimate the values $x_j = x(E_j)$, $j=1,2,\cdots,n$, at a finite number of energy points, one finds that very small variations in the b_i produce very large variations in the estimated x_j.

Another way to say all this is that if there are any errors in the b_i, then it is not possible to accurately determine the value $x_j=x(E_j)$ at any point E_j in the spectrum. Accordingly we seek instead to estimate quantities of the form

$$\phi_k = \int_{E_{lo}}^{E_{up}} w_k(E)x(E)dE, \quad k=1,2,\cdots,p \tag{2-4}$$

where the functions $w_k(E)$ are called window functions. One could think of the $w_k(E)$ as weighting functions and of the ϕ_k as the corresponding weighted averages of $x(E)$. A more intuitive approach is to think of the $w_k(E)$ as being the response functions of an idealized instrument which does not distort the true spectrum as much as does the real instrument. The values $\phi_1,\phi_2,\cdots,\phi_p$ would then be a discrete approximation to the pulse-height spectrum that would have been observed with that idealized instrument.

Figure 2.1 illustrates the response function for a widely used NE-213 spectrometer, showing the pulse height response for several discrete energy levels as a continuous function of the variable pulse height. The figure may be a little misleading because we have been writing the response function as $A_i(E)$, with a continuous variable E for energy and a discrete variable i for pulse height. One such response function, for a single, discrete pulse height value, is shown in the inset at the upper right of the figure.

Sometimes, the instrument channel variable (e.g. pulse height) is in theory really continuous (although we must measure a finite histogram approximation to it), but sometimes the measurement variable is really discrete (as when the channels refer to the pulse count or current output of a spectrometer when different filter thicknesses or different filter materials are used). The energy variable is in principle continuous, but in practice, some sort of finite representation must be used for the numerical process. We shall discuss this finite representation process in more detail in the next chapter. For the time being it will suffice to think of each of the shaded functions shown in the figure as being replaced by a histogram (in the case of a pulse height spectrum) or by a finite number of points in the case of an absorption type spectrometer (where each function represents the response of a different filter or filter thickness), but with the energy being continuous.

In order to obtain a finite computation, the computational process deals with the continuous energy variable by tabulating the m response functions at a selected set of points E_1, E_2, ... E_n. These can be thought of as representative of the points in a range dE, or as a set of "comparison points." The rule for choosing the location of the comparison points is to pick them sufficiently close that the response functions and the windows vary approximately linearly between the comparison points. (Discontinuities or spikes in the response function and/or the window functions can be represented by putting a comparison point at the location of the spike or by putting a pair of comparison points on both "sides" of the discontinuity.)

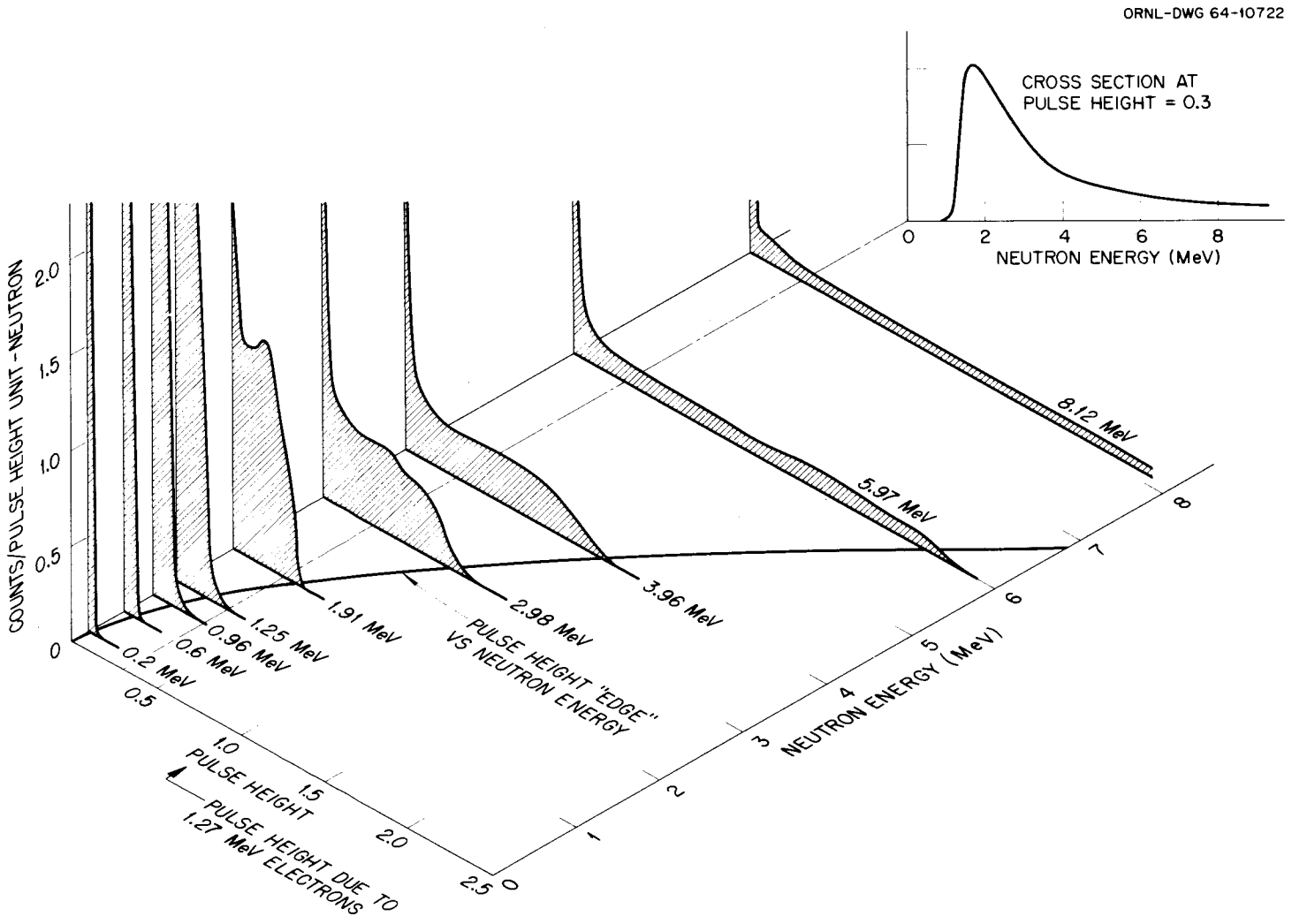


Fig. 2.1. The response surface for an NE-213 spectrometer.

The thing we want to stress here is the distortion in the observed spectrum caused by the irregular shapes of the response functions. Not only will there be a loss in energy resolution caused by the large widths of the response profiles, but also there will be a redistribution in energy caused by the odd shapes and asymmetries. Therefore, the window functions are chosen to have a more regular behavior. A typical choice is shown in Figure 2.2 where the axis labelled "RESPONSE" corresponds to idealized pulse height bins which also correspond to the real pulse height bins that would be chosen in discretizing the pulse height variable in Figure 2.1. Notice that the window function response profiles are, in this case, chosen to be Gaussians which lie along a straight line that determines a linear relationship between neutron energy and pulse-height bin number. It is clear from Figure 2.1 that the corresponding relationship for the real response functions is not linear. Since the window function response functions are symmetric and the relation between energy and pulse height is linear, the idealized spectrometer would not redistribute the spectrum in energy except for the smearing or loss of energy resolution caused by the finite widths of the Gaussian peaks. The best possible idealized instrument would have each of these Gaussians shrink to a delta-function so that $\phi_1,\phi_2,\cdots,\phi_p$ would be an exact point approximation to the true spectrum if the corresponding pulse height bin values were converted to energy units. This is not a good practical working choice of window functions, however, since it essentially reduces the problem back to that of solving the integral equation (2-1), thus defeating the whole purpose of choosing windows.

In choosing the widths of the Gaussian window functions there are two competing considerations that must be taken into account: the statistical uncertainty and the desired energy resolution for the estimated spectrum. Both the FERDO and FERD programs calculate lower and upper bounds for each of the desired window responses ϕ_k. Each such pair of bounds comprise a statistical confidence interval for the corresponding response. The usual procedure in using the codes is to input estimates of the standard deviations σ_i of the counting errors ε_i along with the counts $\overline{b}_i$. These statistical uncertainties are

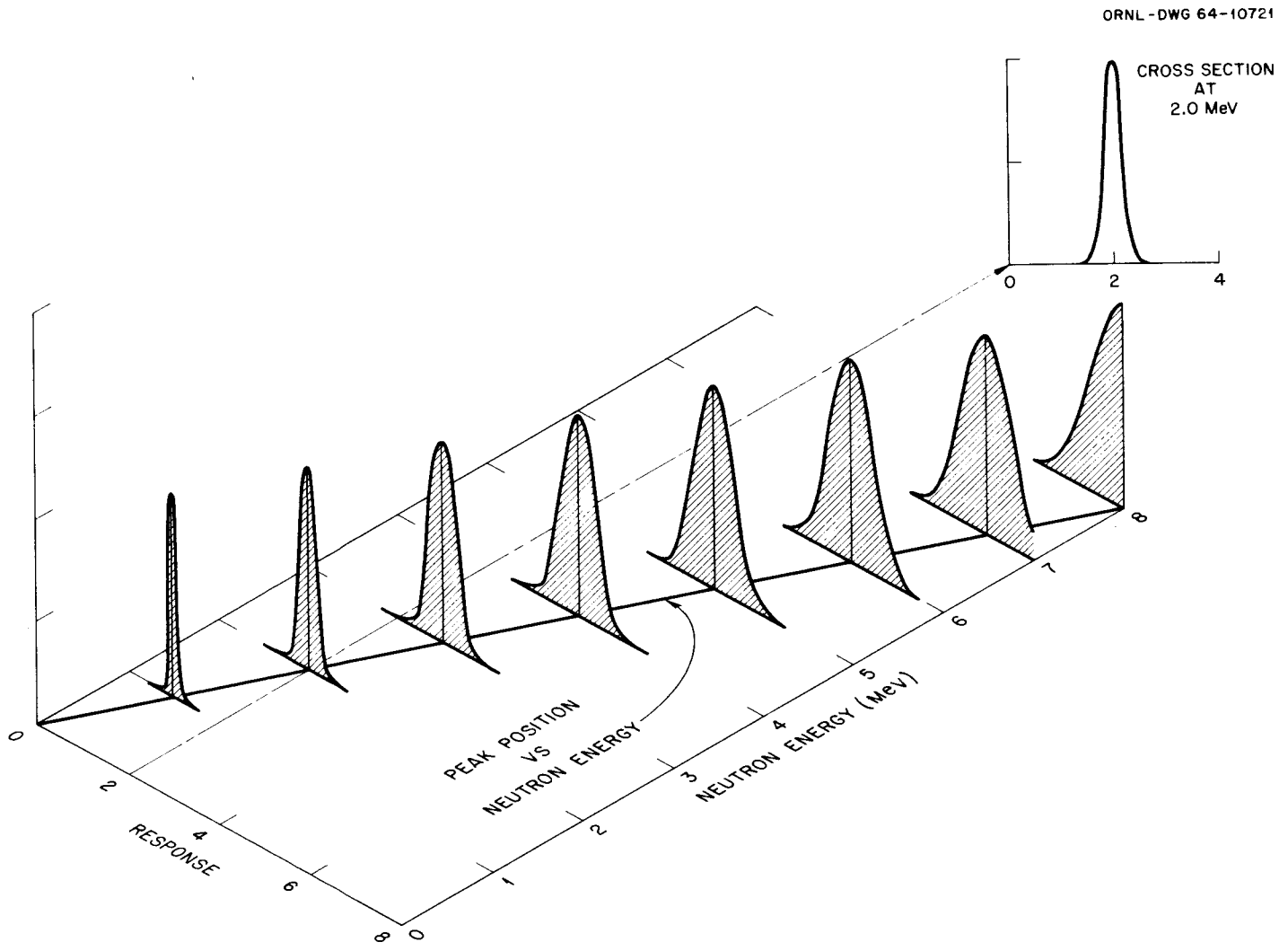


Fig. 2.2. The response surface for an idealized neutron spectrometer.

propagated through the calculations in such a way that the lower and upper bounds ϕ_k^{lo} and ϕ_k^{up} are equal to the desired ϕ_k minus and plus one standard deviation. Since we are assuming normally distributed errors, the resulting interval $[\phi_k^{lo},\phi_k^{up}]$ will be a 68.26% confidence interval for ϕ_k. The width of the interval depends on both the sizes of statistical errors in the counts and on the widths of the Guassian windows. In general, wider windows will produce narrower intervals but may loose energy resolution (i.e., may fail to resolve close pairs of peaks, etc.), while more narrow windows will give wider intervals because such windows are trying to reproduce more information about the structure of the true spectrum, and it is just not possible to obtain such information with the same degree of certainty. The problem in choosing window widths is to somehow balance these two effects in order to maximize the amount of information obtained.

Figure 2.3 shows an observed pulse-height distribution of neutrons from the $Li^6(p,n)Be^6$ reaction using 2.8 and 3.1 MeV protons, and Figure 2.4 shows the corresponding unfolded spectrum which apparently consists of two discrete peaks with no significant continuum component. Each of the dark bars represents a confidence interval estimate for a Gaussian window centered at that particular energy and the lightly shaded area was obtained by simple interpolation between these estimated window responses. One could, of course, obtain more certainty about the spectrum in these interpolated regions by choosing more windows centered in these regions, but even if this were done it is obvious from the apparent widths of the two partially resolved peaks that they would still not be completely resolved. One could attempt to resolve them by choosing the windows more narrow but such a procedure might not help matters because of the resulting growth of the interval widths.

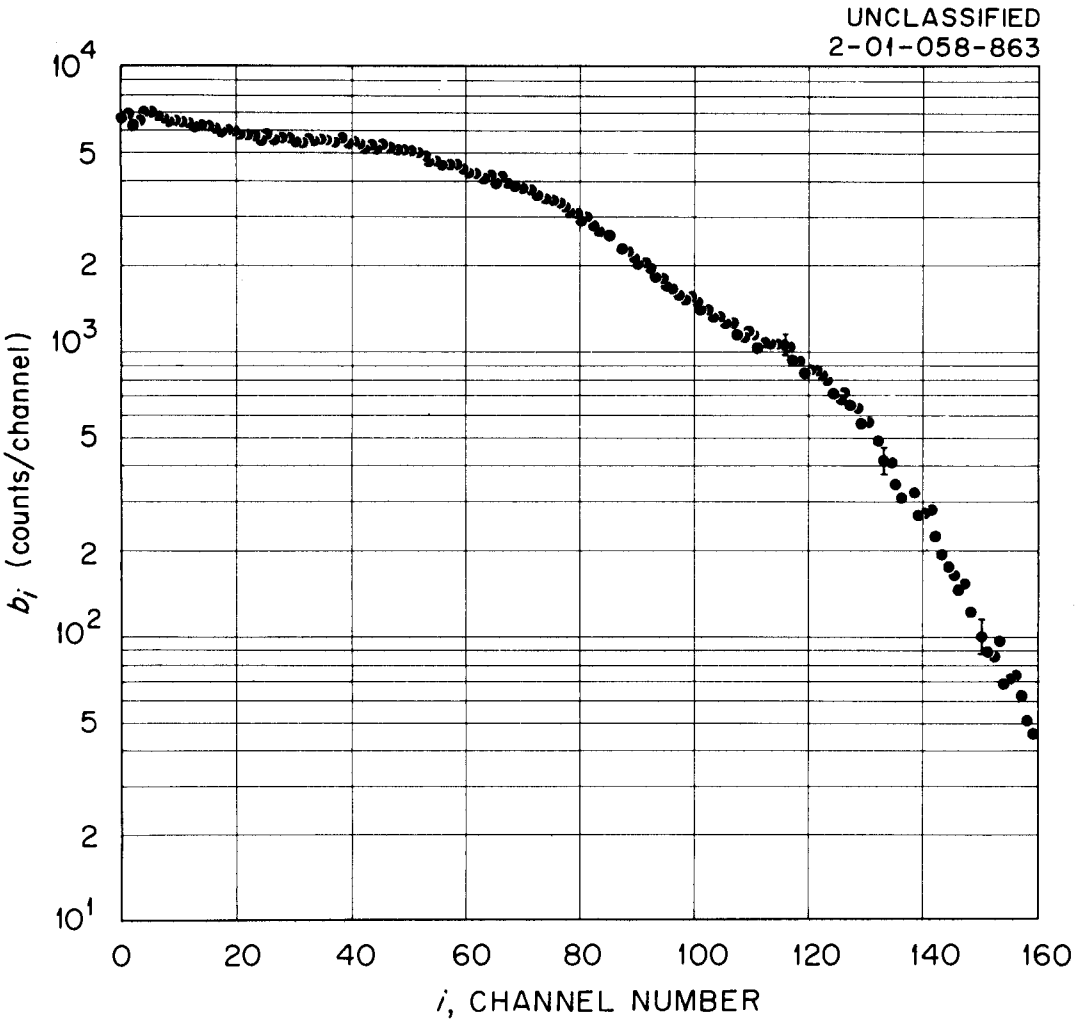


Fig. 2.3. Pulse-height distribution of neutrons from the $Li^6(p,n)Be^6$ reaction using 2.8 and 3.1 MeV protons.

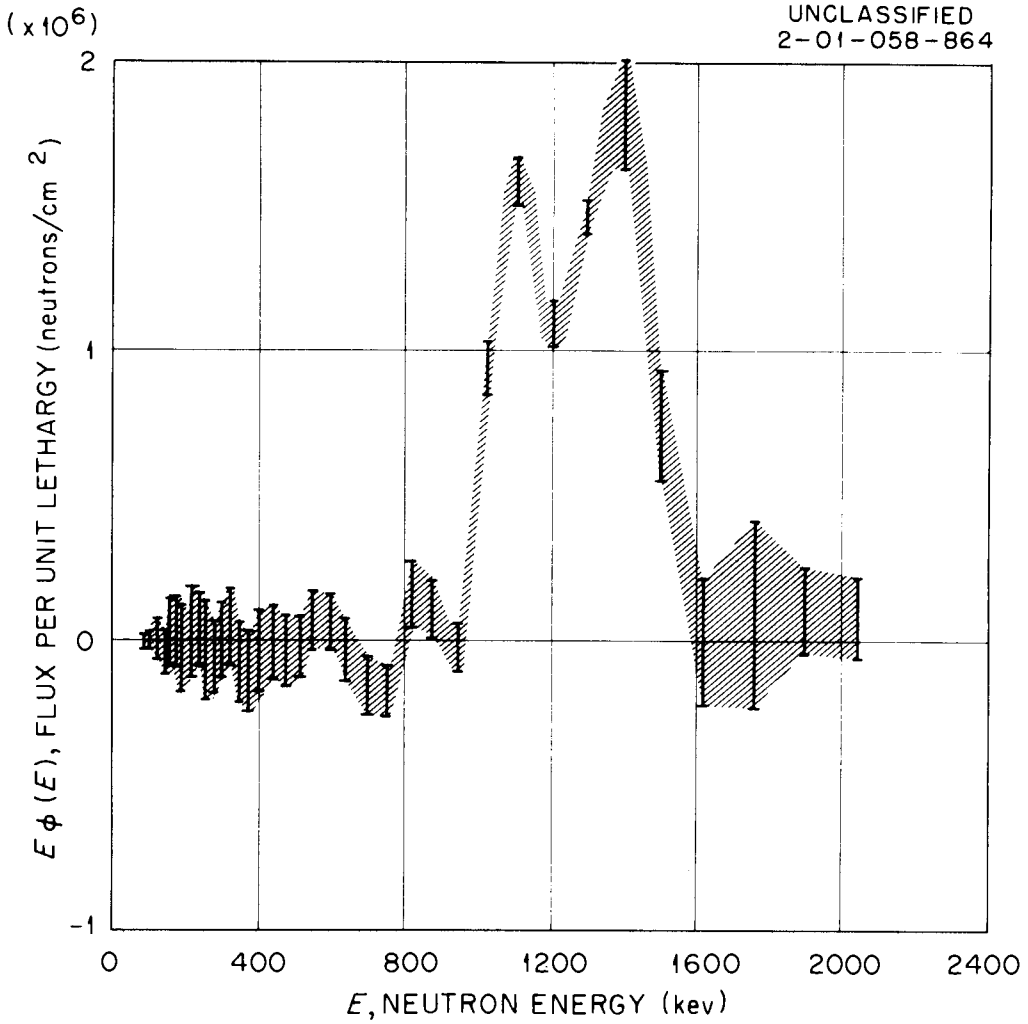


Fig. 2.4. Unfolded spectrum of neutrons resulting from the $Li^6(p,n)Be^6$ reactions, using 2.8 and 3.1 MeV protons.

3. THE DISCRETIZATION OF THE PROBLEM

In the preceding chapter we saw that the mathematical problem which the FERDO and FERD programs attempt to solve can be written as follows: given window functions $w_1(E)$, $w_2(E)$, ⋯, $w_p(E)$, to calculate for each such $w_k(E)$ the lower and upper bounds

$$\phi_k^{lo} = \min_{x(E)} \left\{ \int_{E_{lo}}^{E_{up}} w_k(E)x(E)dE \;\middle|\; \int_{E_{lo}}^{E_{up}} A_i(E)x(E)dE = \bar{b}_i + \varepsilon_i,\ \varepsilon_i \sim N(0,\sigma_i^2),\ i=1,2,\cdots,m \right\}, \tag{3-1}$$

$$\phi_k^{up} = \max_{x(E)} \left\{ \int_{E_{lo}}^{E_{up}} w_k(E)x(E)dE \;\middle|\; \int_{E_{lo}}^{E_{up}} A_i(E)x(E)dE = \bar{b}_i + \varepsilon_i,\ \varepsilon_i \sim N(0,\sigma_i^2),\ i=1,2,\cdots,m \right\}.$$

That is, ϕ_k^{lo} and ϕ_k^{up} are the smallest and largest values that $\int_{E_{lo}}^{E_{up}} w_k(E)\,x(E)dE$ can assume when the unknown function $x(E)$ is constrained to satisfy the integral equations

$$\int_{E_{lo}}^{E_{up}} A_i(E)x(E)dE = \bar{b}_i + \varepsilon_i,\ i=1,2,\cdots,m,$$

with the ε_i being independent samples from the normal distributions with means 0 and standard deviations σ_i.

The programs do not attempt to solve the exact problems stated above, but rather finite, discrete approximations to them. The first step in the discretization is to replace the continuous variable E with a mesh E_1, E_2, ⋯, E_n spanning the interval $[E_{lo},E_{up}]$, i.e., $E_{lo} \leq E_1 < E_2 < \cdots < E_n \leq E_{up}$. There are two ways to think about this step. We consider first the philosophy adopted by most users even though it is not the most suggestive or fruitful approach to the problem. Let the interval width associated with each mesh point E_i be denoted by ΔE_i, and for each window function $w_k(E)$ let

$$w_{kj} \equiv w_k(E_j),\ j=1,2,\cdots,n. \tag{3-2}$$

It follows then that

$$\int_{E_{lo}}^{E_{up}} w_k(E)x(E)dE \cong \sum_{j=1}^{n} w_{kj}\, x_j, \tag{3-3}$$

where

$$x_j \equiv x(E_j)\,\Delta E_j,\ j=1,2,\cdots,n \tag{3-4}$$

This definition of the quantity x_j has in the past been a source of confusion to some users of the FERDOR codes. In FERDO and FERD, each of the x_j represents the total number of neutrons (or gamma photons) in the corresponding energy interval ΔE_j. The constraint integral equations are replaced by a set of linear algebraic equations by using the quadrature approximations

$$\int_{E_{lo}}^{E_{up}} A_i(E)x(E)dE \cong \sum_{j=1}^{n} A_{ij}\, x_j,\ i=1,2,\cdots,m \tag{3-5}$$

where

$$A_{ij} = A_i(E_j),\ j=1,2,\cdots,n. \tag{3-6}$$

The approximate constraint equations are then

$$\sum_{j=1}^{n} A_{ij}\, x_j = \bar{b}_i + \varepsilon_i,\ i=1,2,\cdots,m, \tag{3-7}$$

and the problems (3-1) can be rewritten

$$\phi_k^{lo} = \min_{(x_1,x_2,\cdots,x_n)} \left\{ \sum_{j=1}^{n} w_{kj}\, x_j \;\middle|\; \sum_{j=1}^{n} A_{ij} x_j = \bar{b}_i + \varepsilon_i,\ \varepsilon_i \sim N(0,\sigma_i^2),\ i=1,2,\cdots,n \right\}, \tag{3-8}$$

$$\phi_k^{up} = \max_{(x_1,x_2,\cdots,x_n)} \left\{ \sum_{j=1}^{n} w_{kj}\, x_j \;\middle|\; \sum_{j=1}^{n} A_{ij} x_j = \bar{b}_i + \varepsilon_i,\ \varepsilon_i \sim N(0,\sigma_i^2),\ i=1,2,\cdots,n \right\}.$$

The problems (3-8) can be written more compactly by using vector-matrix notation. Let the n-dimension column vector $\underline{x}$ be defined by

$$\underline{x} = (x_1, x_2, \cdots, x_n)^T, \tag{3-9}$$

where the superscript T means transpose. For each window function $w_k(E)$ let the corresponding discrete representation be written as an n-dimensional window vector,

$$\underline{w}_k = (w_{k1}, w_{k2}, \cdots, w_{kn})^T \tag{3-10}$$

The quadrature approximation (3-3) can then be written:

$$\sum_{j=1}^{n} w_{kj} x_j = (w_{k1}, w_{k2}, \cdots, w_{kn}) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \underline{w}_k^T \underline{x} \ . \tag{3-11}$$

To write the constraint equations more compactly, let the m-vectors $\underline{\bar{b}}$ and $\underline{\varepsilon}$ be defined by

$$\begin{aligned} \underline{\bar{b}} &= (\bar{b}_1, \bar{b}_2, \cdots, \bar{b}_m)^T \\ \underline{\varepsilon} &= (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m)^T \end{aligned} \tag{3-12}$$

and the m x n response matrix $\underline{\underline{A}}$ be

$$\underline{\underline{A}} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} . \tag{3-13}$$

(Note that the rows of the response matrix correspond to pulse height bins and the columns correspond to the energy mesh points E_j.) The quadrature approximations (3-5) can then all be written

$$\sum_{j=1}^{n} A_{ij} x_j = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \underline{\underline{A}} \, \underline{x}, \tag{3-14}$$

and the constraint equations become

$$\underline{\underline{A}}\ \underline{x} = \underline{b} + \underline{\varepsilon}\ , \tag{3-15}$$

with

$$\underline{\varepsilon} \sim N(\underline{0}, \underline{\underline{\Sigma}}^2)\ , \tag{3-16}$$

where $\underline{\underline{\Sigma}}^2$ is the diagonal variance-covariance matrix

$$\underline{\underline{\Sigma}}^2 = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \sigma_1^2 & \\ & & \ddots \\ 0 & & \sigma_n^2 \end{bmatrix} . \tag{3-17}$$

The expression (3-16) means that the statistical error vector $\underline{\varepsilon}$ has a multivariate normal distribution with the n-dimensional zero-vector as its mean and with a variance-covariance matrix $\underline{\underline{\Sigma}}^2$ whose off-diagonal elements are all zero because the individual elements ε_i are not correlated with one another. The diagonal elements of $\underline{\underline{\Sigma}}^2$ are just the variances of the corresponding individual ε_i. The expression (3-16) can be written more precisely by specifying the probability density function for the vector $\underline{\varepsilon}$, i.e.,

$$f(\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n) = \frac{1}{\sqrt{(2\pi)^m}(\sigma_1 \sigma_2 \cdots \sigma_m)} \exp\left[-\frac{1}{2}\sum_{i=1}^{m} \frac{\varepsilon_i^2}{\sigma_i^2}\right] , \tag{3-18a}$$

or, in vector-matrix notation,

$$f(\underline{\varepsilon}) = \frac{1}{\sqrt{(2\pi)^m \det(\underline{\underline{\Sigma}}^2)}} \exp\left[-\frac{1}{2}\underline{\varepsilon}^T \underline{\underline{\Sigma}}^{-2}\underline{\varepsilon}\right] , \tag{3-18b}$$

where det $(\underline{\underline{\Sigma}}^2)$ is the determinant of the matrix $\underline{\underline{\Sigma}}^2$.

Substituting equations (3-11, 15, 16) allows us to write the problems (3-8) in the form

$$\phi_k^{lo} = \min_{\underline{x}} \left\{ \underline{w}_k^T \underline{x} \;\middle|\; \underline{\underline{A}}\,\underline{x} = \underline{\bar{b}} + \underline{\varepsilon},\ \underline{\varepsilon} \sim N(\underline{0}, \underline{\underline{\Sigma}}^2) \right\},$$

$$\phi_k^{up} = \max_{\underline{x}} \left\{ \underline{w}_k^T \underline{x} \;\middle|\; \underline{\underline{A}}\,\underline{x} = \underline{\bar{b}} + \underline{\varepsilon},\ \underline{\varepsilon} \sim N(\underline{0}, \underline{\underline{\Sigma}}^2) \right\}. \qquad (3\text{-}19)$$

These expressions state the problems in the same vector-matrix language used by the programs. The following short table summarizes some of the terminology used extensively in the programs and in the remainder of this report.

Name	Mathematical Symbol	FORTRAN Variable
Response Matrix	$\underline{\underline{A}}$	A(I,J),I=1,NR,* J=1,NC.
No. Rows in $\underline{\underline{A}}$	m	NR
No. Columns in $\underline{\underline{A}}$	n	NC
Count Vector	$\underline{\bar{b}}$	B(I),I=1,NR
Window Vector No. k	$\underline{w}_k$	W(J),J=1,NC**
No. of Window Vectors	p	NW
Lower Bound for Window k	ϕ_k^{lo}	PLØ(K),K=1,NW
Upper Bound for Window k	ϕ_k^{up}	PUP(K),K=1,NW
Solution Vector	x	X(J),J=1,NC

*FERD actually uses lower and upper bounds for

$\underline{\underline{A}}$, i.e. ALØ(I,J) and AUP(I,J).

**Each window vector is either computed when it is needed or is copied into W(J) from a storage array called

WINDØS(K,J),K=1,NW,J=1,NC.

Although the expressions (3-19) state the problems very compactly, they are still not reduced to a form suitable for calculations. The reason is that the vector $\underline{\varepsilon}$ is not a known quantity. In general the experimenter will obtain a count vector $\underline{\bar{b}}$ which contains some error and even though it might be known that the error was drawn from a completely specified multivariate normal distribution, there is no way to determine what the actual error values were. To get around this difficulty it is necessary to apply some statistical analysis. This will be done in the next chapter.

Before addressing these statistical matters we briefly consider the second approach to discretizing the continuous integral equations. As the first step in the discretization, one may think of replacing the continuous response functions and window functions (perhaps with spikes and discontinuities) by a set of linearized functions which are linear between a set of comparison points E_1, E_2, ... E_n. The integral equations then become:

$$\int_{E_{lo}}^{E_{up}} A_i^{lin}(E)\, x\,(E)dE = \bar{b}_i + \varepsilon_i \qquad i = 1,2,\cdots,m \tag{3-20}$$

where $A_i^{lin}(E)$ is linear between the comparison points E_1, E_2, ..., E_n.

$$\phi_k = \int_{E_{lo}}^{E_{up}} w_k^{lin}(E)\, x\,(E)dE \qquad k = 1,2,\cdots,p \tag{3-21}$$

where $w_k^{lin}(E)$ is linear between the comparison points E_1, E_2, ..., E_n. Note that x(E) is still an unrestricted function of the continuous variable E. Also note that the comparison points for $A_i(E)$ are the same set as used for $w_k(E)$. Therefore, if either the $A_i(E)$ functions or the $w_k(E)$ have fine detail in a particular energy range, then a common set of comparison points should be chosen such that the behavior of both the $A_i(E)$ and the $w_k(E)$ functions are approximately linear between comparison points.

The next step in the discretization process is to note that one of the tenets of the FERD and FERDO methods is that one never attempts to solve for x(E) directly, but only for a set of p window functions (where p may be very large so that window functions corresponding to adjacent k's are very slightly different). In the limit as p gets very large, one can think of transforming the underlying x(E) variable to a $\phi(k)$ variable.

The piecewise linear functions $A_i^{lin}(E)$, $w_k^{lin}(E)$ can be written in terms of a set of piecewise linear spline basis functions $L_j(E)$ which satisfy

$$L_1(E) = \begin{cases} 1, & E = E_1, \\ 0, & E \geqslant E_2, \end{cases}$$

$$L_j(E) = \begin{cases} 0, & E \leqslant E_{j-1}, \quad j=2,3,\cdots, n-1, \\ 1, & E = E_j, \quad j=2,3,\cdots, n-1, \\ 0, & E \geqslant E_{j+1}, \quad j=2,3,\cdots, n-1, \end{cases}$$

$$L_n(E) = \begin{cases} 0, & E \leqslant E_{n-1}, \\ 1, & E = E_n . \end{cases}$$

The expansions are

$$A_i^{lin}(E) = \sum_{j=1}^{n} A_{ij} L_j(E), \quad i=1,2,\cdots, m, \tag{3-22}$$

and

$$w_k^{lin}(E) = \sum_{j=1}^{n} w_{kj} L_j(E), \quad i=1,2,\cdots, p, \tag{3-23}$$

where the A_{ij} and w_{kj} are appropriately chosen coefficients. Substituting these expansions into the integral equations and exchanging the order of the integral and summation operators gives

$$\sum_{j=1}^{n} A_{ij}\left[\int_{E_{lo}}^{E_{up}} L_j(E)x(E)dE\right] = \bar{b}_i + \varepsilon_i, \quad i=1,2,\cdots, m, \tag{3-24}$$

$$\phi_k = \sum_{j=1}^{n} w_{kj}\left[\int_{E_{lo}}^{E_{up}} L_j(E)x(E)dE\right], \quad k=1,2,\cdots, p. \tag{3-25}$$

If we now define the quantities x_j by

$$x_j = \int_{E_{lo}}^{E_{up}} L_j(E)x(E)dE, \quad j=1,2,\cdots, n, \tag{3-26}$$

then the integral equations become vector-matrix equations

$$\sum_{j=1}^{n} A_{ij}\, x_j = \bar{b}_i + \varepsilon_i, \quad i=1,2,\cdots, m,$$

$$\phi_k = \sum_{j=1}^{n} w_{kj}\, x_j, \quad k=1,2,\cdots, p,$$

identical in form to those obtained in the first method of discretization. The only difference is in the interpretation of the A_{ij}, w_{kj} and the x_j.

The piecewise linearity assumption is not crucial in the above described discretization. The $L_j(E)$ could be replaced by any set of spline basis functions to give the same vector-matrix equations although the values of A_{ij}, w_{kj} and x_j would differ. Equation (3-26) would still define the x_j which could be interpreted as a weighted average of the unknown function $x(E)$ with $L_j(E)$ being the weighting function. Note that the first method of discretization can be expressed in this form also if the $L_j(E)$ are taken to be "boxcar windows" with unit height and width ΔE_j

Some advice to those concerned with the codes. First, do not worry about what x_j means. x_j should never assume an important role and should

never enter into any computation for the solution (except perhaps as a numerical artifice). Second, when approximating the functions $K_j(E)$ and $w_k(E)$, do not worry about intervals dE. Instead, focus on the actual value of the functions $K_j(E)$ and $w_k(E)$ at the comparison points. The exact spacing of the comparison points is not important (so long as there is a sufficient number that the functions are approximately linear in between). And finally, do not worry at all about dE since it never occurs in any equations (whether as a computational artifice or otherwise). Unfortunately, some practioners of the FERD and FERDO codes have been confused on this point and have let some of this confusion creep into their codes. Usually, it does not hurt too much - the main test being to see if it can be eliminated without changing the values of the estimates for ϕ (which do mean something physical).

4. STATISTICAL CONSIDERATIONS

Since the error vector $\underline{\varepsilon}$, when considered as a vector of random variables, has the zero-vector as its mean, one can think of the count vector $\underline{b}$ as a sample drawn from a multivariate normal distribution whose mean vector $\underline{b}_0$ corresponds to a hypothetical measurement with no measuring error. Furthermore, the variance-covariance matrix for this $\underline{b}$-distribution will be the same as that for the $\underline{\varepsilon}$-distribution, namely $\underline{\underline{\Sigma}}^2$. Thus, we can write

$$\underline{b} \sim N(\underline{b}_0, \underline{\underline{\Sigma}}^2),$$

and the probability distribution function for $\underline{b}$ is

$$f(\underline{b}) = \frac{1}{\sqrt{(2\pi)^m \det(\underline{\underline{\Sigma}}^2)}} \exp\left[-\frac{1}{2}(\underline{b}-\underline{b}_0)^T \underline{\underline{\Sigma}}^{-2}(\underline{b}-\underline{b}_0)\right], \qquad (4\text{-}1)$$

where $\det(\underline{\underline{\Sigma}}^2) = \sigma_1^2\sigma_2^2\cdots\sigma_m^2$ is the determinant of the matrix $\underline{\underline{\Sigma}}^2$. Clearly the equi-probability contours for this distribution are surfaces (in $\underline{b}$-space) such that

$$(\underline{b}-\underline{b}_0)^T \underline{\underline{\Sigma}}^{-2} (\underline{b}-\underline{b}_0) = \text{constant.}$$

For any given constant value $\mu^2 \geq 0$, there exists a probability level α, $0 \leq \alpha < 1$, such that the m-dimension region defined by

$$(\underline{b}-\underline{b}_0)^T \underline{\underline{\Sigma}}^{-2} (\underline{b}-\underline{b}_0) \leq \mu^2 \tag{4-2}$$

has an associated probability α of containing a given sample vector $\underline{b}$. Since the $\underline{b}$ vectors are normally distributed, the quantity $(\underline{b}-\underline{b}_0)^T \underline{\underline{\Sigma}}^{-2} (\underline{b}-\underline{b}_0)$ has a chi-square distribution with m-degrees of freedom, i.e.,

$$(\underline{b}-\underline{b}_0)^T \underline{\underline{\Sigma}}^{-2} (\underline{b}-\underline{b}_0) \sim \chi^2 (m).$$

Thus, the relationship between a given probability level α and the associated value μ^2 is given by

$$\alpha = \int_0^{\mu^2} \frac{\nu^{m/2-1} \exp(-\nu/2)}{\Gamma(m/2)\, 2^{m/2}} \, d\nu \tag{4-3}$$

where Γ represents the gamma function, and the whole integrand is just the expression for the $\chi^2(m)$ probability distribution function. This relationship between α and μ^2 is commonly tabulated in standard tables of the Chi-square distribution, (e.g. Beyer 1968, p. 294).

In practice one does not know the vector $\underline{b}_0$ but only that it satisfies

$$\underline{\underline{A}}\underline{x} = \underline{b}_0 \quad . \tag{4-4}$$

What one does know is an actual measured $\overline{\underline{b}}$. Substituting that quantity for $\underline{b}$ and $\underline{\underline{A}}\underline{x}$ for $\underline{b}_0$ in Eq. (4-2) gives

$$(\overline{\underline{b}}-\underline{\underline{A}}\underline{x})^T \underline{\underline{\Sigma}}^{-2} (\overline{\underline{b}}-\underline{\underline{A}}\underline{x}) \leq \mu^2 \; . \tag{4-5}$$

One cannot actually work with this expression either because the quantities $\sigma_1,\sigma_2,\cdots,\sigma_m$ are not known exactly. It is possible, however, to get estimates of the standard deviations from the measured $\underline{\bar{b}}$ vector. Since the elements of $\underline{\bar{b}}$ are obtained from a counting process, each of the standard deviations is approximately equal to the square root of the numbers of counts in the corresponding bin. The estimation of the counting errors is discussed in more detail in a later chapter. The FERDO and FERD codes both require an estimate of the σ_i for each of the $\bar{b}_i$. Denoting these estimates by s_i and defining the diagonal matrix $\underline{\underline{S}}$ by

$$\underline{\underline{S}} = \text{diag}\ (s_1,s_2,\cdots,s_m) \tag{4-6}$$

enables us to write the following approximation for Eq. (4-5):

$$(\underline{\bar{b}} - \underline{\underline{A}}\underline{x})^T \underline{\underline{S}}^{-2} (\underline{\bar{b}} - \underline{\underline{A}}\underline{x}) \leqslant \mu^2 , \tag{4-7}$$

where

$$\underline{\underline{S}}^{-2} = \text{diag}\left(\frac{1}{s_1^2}, \frac{1}{s_2^2}, \ldots, \frac{1}{s_m^2}\right). \tag{4-8}$$

Once the confidence level α is fixed and the corresponding value of μ^2 is determined from the χ^2 (m) distribution, then all of the quantities in Eq (4-7) are known except the solution vector $\underline{x}$. We can now rewrite the problems (3-19) in a form very close to the ones actually solved by the FERDO and FERD codes. This is done by substituting Eq. (4-7) for the constraints in (3-19), i.e.

$$\phi_k^{lo} = \min_{\underline{x}} \left\{ \underline{w}_k^T \underline{x} \;\middle|\; (\underline{\bar{b}} - \underline{\underline{A}}\underline{x})^T \underline{\underline{S}}^{-2} (\underline{\bar{b}} - \underline{\underline{A}}\underline{x}) \leqslant \mu^2 \right\} ,$$

$$\phi_k^{up} = \max_{x} \left\{ \underline{w}_k^T \underline{x} \;\middle|\; (\underline{\bar{b}} - \underline{\underline{A}}\underline{x})^T \underline{\underline{S}}^{-2} (\underline{\bar{b}} - \underline{\underline{A}}\underline{x}) \leqslant \mu^2 \right\} . \tag{4-9}$$

These expressions state that an α-level confidence interval $\left[\phi_k^{lo}\ ,\ \phi_k^{up}\right]$ could be obtained by determining the minimum and maximum values that the quantity $\underline{w}_k^T\ \underline{x}$ can obtain when $\underline{x}$ is constrained to lie in the region defined by the inequality

$$(\underline{\bar{b}} - \underline{\underline{A}}\ \underline{x})^T \underline{\underline{S}}^{-2} (\underline{\bar{b}} - \underline{\underline{A}}\ \underline{x}) \leq \mu^2 .$$

We will describe the geometry of this constraint region in more detail in the next chapter.

Using the χ^2 distribution to determine μ^2 from α gives very conservative confidence intervals which would, in fact, be unnecessarily large if Eqs. (4-9) were the final statements of the problems to be solved. There is, however, still one more constraint on the solution vector which is required by the FERDO and FERD codes. That constraint is that the elements of the vector $\underline{x}$ all be non-negative, i.e. $\underline{x} \geq \underline{0}$. This simple physical constraint is the key to the success of the whole procedure. The succeeding chapters will explain why this is true.

5. GEOMETRY OF THE CONSTRAINT REGION

The constraint region (4-7), which can also be written

$$(\underline{\underline{A}}\ \underline{x} - \underline{\bar{b}})^T \underline{\underline{S}}^{-2} (\underline{\underline{A}}\ \underline{x} - \underline{\bar{b}}) \leq \mu^2 \qquad (5\text{-}1)$$

defines a region in x-space which is designed to contain the unknown solution vector $\underline{x}$ with probability α. If a hypothetical experimenter were to repeat the measurements 1000 times, obtaining a different $\underline{\bar{b}}$ vector and different $\underline{\underline{S}}$ matrix each time, and construct the corresponding 1000 constraint regions defined by (5-1), then approximately 1000 α of those regions would contain the true but unknown $\underline{x}$.

Every $\underline{x}$ in x-space defines a vector $\underline{b}$ in b-space through the relation

$$\underline{b} = \underline{\underline{A}}\ \underline{x} . \qquad (5\text{-}2)$$

(The true but unknown $\underline{x}$ gives $\underline{b} = \underline{b}_o$). It is instructive to substitute (5-2) into (5-1) and consider the corresponding region in b-space, i.e. the region defined by

$$(\underline{b} - \underline{\bar{b}})^T S^{-2} (\underline{b} - \underline{\bar{b}}) \leq \mu^2 \qquad (5\text{-}3)$$

where $\underline{\bar{b}}$ is the given measured count vector and $\underline{b}$ is free to vary over all of b-space. The difference vector $(\underline{b}-\underline{\bar{b}})$ has elements $(\underline{b}-\underline{\bar{b}})_i$, and the inequality (5-3) can also be written

$$(b_1 - \bar{b}_1,\ b_2 - \bar{b}_2, \ldots, b_m - \bar{b}_m) \begin{pmatrix} 1/s_1^2 & & & \\ & 1/s_2^2 & & 0 \\ & & \ddots & \\ & 0 & & 1/s_m^2 \end{pmatrix} \begin{pmatrix} b_1 - \bar{b}_1 \\ b_2 - \bar{b}_2 \\ \vdots \\ b_m - \bar{b}_m \end{pmatrix} \leq \mu^2$$

or

$$\frac{(b_1 - \bar{b}_1)^2}{s_1^2} + \frac{(b_2 - \bar{b}_2)^2}{s_2^2} + \ldots + \frac{(b_m - \bar{b}_m)^2}{s_m^2} \leq \mu^2$$

Dividing through by μ^2 and replacing the inequality by a strict equality gives, as the equation defining the surface of the constraint region, the expression

$$\frac{(b_1 - \bar{b}_1)^2}{(\mu s_1)^2} + \frac{(b_2 - \bar{b}_2)^2}{(\mu s_2)^2} + \ldots + \frac{(b_m - \bar{b}_m)^2}{(\mu s_m)^2} = 1$$

which is immediately recognizable as the standard form for the equation of an m-dimensional ellipsoid centered at $(\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_m)$, with axes lengths $\mu s_1, \mu s_2, \ldots, \mu s_m$. This ellipsoid is shown in Figure 5.1 where m is taken to be 3 for illustrative purposes. Note that the axes of the ellipsoid are parallel to the coordinate axes. As μ^2 increases the size of the ellipsoid increases as does the probability α that it contains the $\underline{b}_o$ corresponding to the true $\underline{x}$.

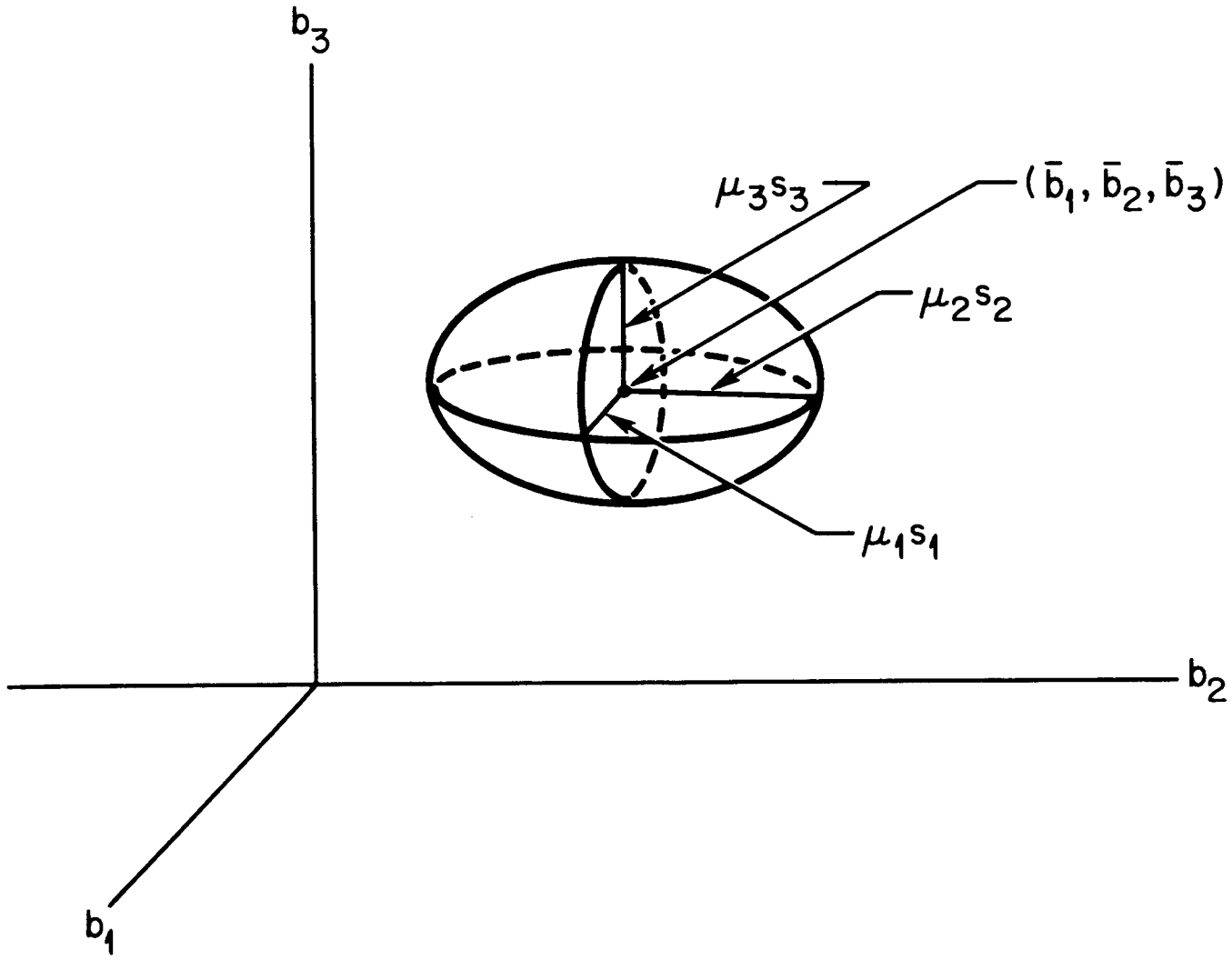


Fig. 5.1. The Constraint Ellipsoid in b-space.

In general, not every point in the b-space will be the unique image of a point in x-space since m and n are not generally equal to one another. In principle, the FERDO and FERD methods yield valid confidence interval estimates for any combination of m and n, but experience with a number of problems has shown us that one almost always obtains better (narrower) intervals if $n \leq m$. To illustrate the relationship between the constraint region in b-space and x-space we take $n=2$ in Figure 5.2. In the top portion of the figure is the same constraint ellipsoid shown in Fig. 5.1. Also shown is the plane defined by the transformation

$$\underline{b} = \underline{\underline{A}}\,\underline{x}$$

All of the points in the two-dimensional x-space map into this plane which is a two-dimensional subspace of the three-dimensional b-space. In particular, the point $\underline{b}_o$ corresponding to the true $\underline{x}$ lies somewhere in this plane, hopefully inside its intersection with the b-ellipsoid. Note that the measured count vector $\bar{\underline{b}}$, which is the center of the b-ellipsoid does not lie on the plane $\underline{b} = \underline{\underline{A}}\,\underline{x}$ because it contains a random measuring error.

The constraint surface in x-space is the ellipsoid shown in the bottom part of the figure, which corresponds to the intersection of the $\underline{b} = \underline{\underline{A}}\,\underline{x}$ plane with the b-ellipsoid. It is not obvious from the defining equation,

$$(\underline{\underline{A}}\,\underline{x} - \bar{\underline{b}})^T \underline{\underline{S}}^{-2} (\underline{\underline{A}}\,\underline{x} - \bar{\underline{b}}) = \mu^2,$$

that this constraint surface is actually an ellipsoid but if one rewrites the equation in terms of its center $\hat{\underline{x}}$ the ellipsoidal character becomes more apparent. We will do this in the next section.

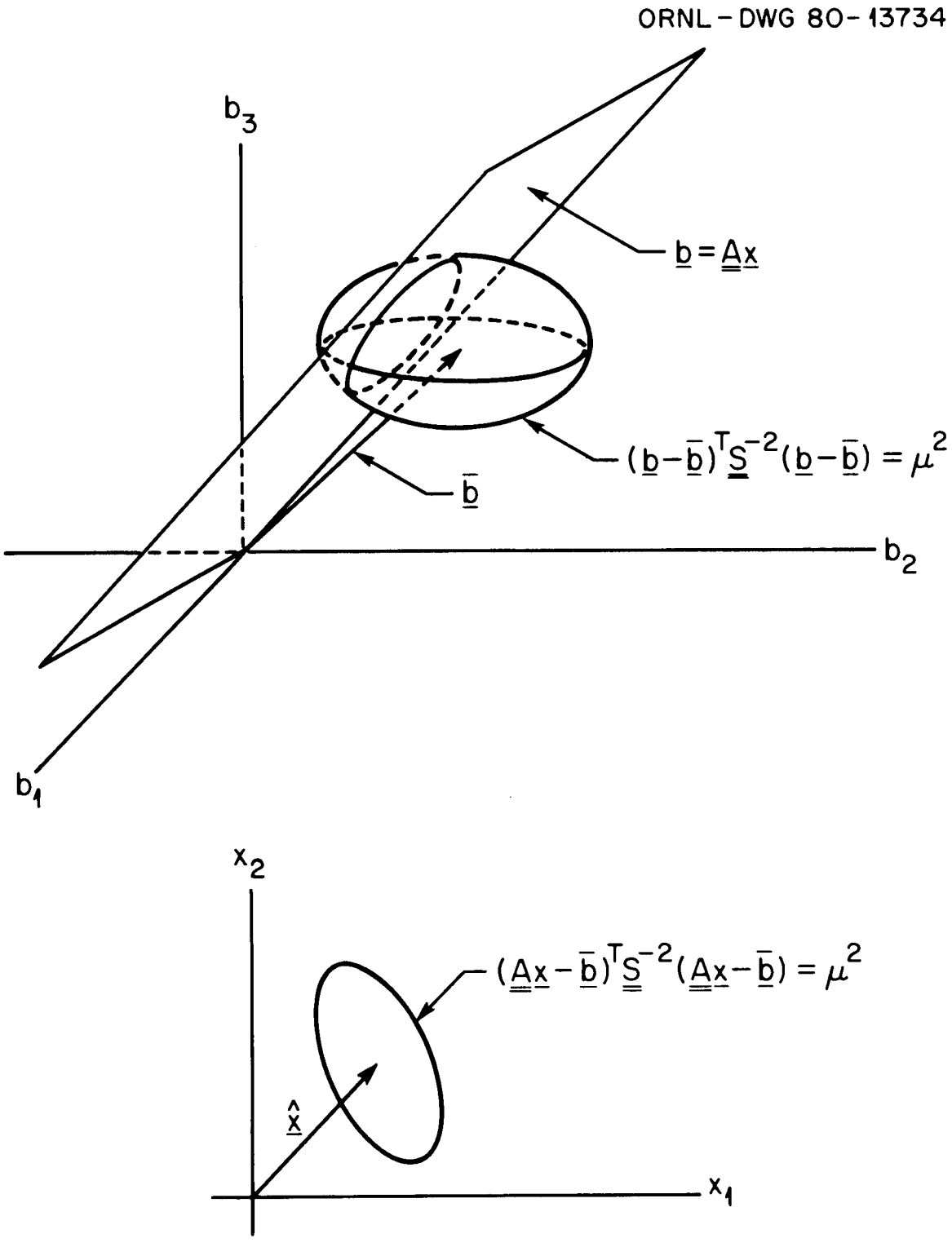


Fig. 5.2. Constraint Ellipsoids in b-space and x-space.

6. LEAST SQUARES ESTIMATION

Because of the random measuring error $\underline{\varepsilon}$ in the measured count vector $\underline{\bar{b}}$, the system of equations

$$\underline{\underline{A}}\underline{x} = \underline{\bar{b}}$$

will in general not have an exact solution. The common practice in such situations is to seek the least squares solution, i.e. the vector $\underline{x}$ which minimizes the weighted sum of squared residuals

$$\rho(\underline{x}) = \sum_{i=1}^{m} \frac{\left[(\underline{\underline{A}}\,\underline{x})_i - \bar{b}_i\right]^2}{s_i^2} .$$

The residuals are weighted inversely with the uncertainties s_i in the measured $\bar{b}_i$. The above expression can also be written in vector-matrix form as

$$\rho(\underline{x}) = (\underline{\underline{A}}\,\underline{x} - \underline{\bar{b}})^T\, \underline{\underline{S}}^{-2}\, (\underline{\underline{A}}\,\underline{x} - \underline{\bar{b}}).$$

In such problems m is taken to be greater than (or equal) to n and the columns of the matrix $\underline{\underline{A}}$ are assumed to be linearly independent vectors spanning an n-dimensional subspace of the m-dimensional b-space. When these assumptions are satisfied it is easy to show that the minimum value of $\rho(\underline{x})$ occurs when $\underline{x}$ is equal to

$$\underline{\hat{x}} = (\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}})^{-1}\,\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\bar{b}} \quad . \tag{6-2}$$

This least squares solution vector turns out to be the center of the constraint ellipsoid shown in the bottom part of Fig. 5-2, and the equation of that ellipsoid can be written

$$\rho(\underline{x}) = \mu^2. \tag{6-3}$$

If we denote the minimum value of $\rho(\underline{x})$ by ρ_o, i.e.

$$\rho_o = \rho(\underline{\hat{x}}) = (\underline{\underline{A}}\,\underline{\hat{x}} - \underline{\bar{b}})^T\underline{\underline{S}}^{-2}(\underline{\underline{A}}\,\underline{\hat{x}} - \underline{\bar{b}}),$$

it can be shown that

$$\rho(\underline{x}) = \rho_o + (\underline{x} - \hat{\underline{x}})^T \underline{\underline{A}}^T \underline{\underline{S}}^{-2} \underline{\underline{A}} (\underline{x} - \hat{\underline{x}})$$

and the constraint equation (6-3) can be written

$$(\underline{x} - \hat{\underline{x}})^T \underline{\underline{A}}^T \underline{\underline{S}}^{-2} \underline{\underline{A}} (\underline{x} - \hat{\underline{x}}) = \mu^2 - \rho_o \qquad (6\text{-}4)$$

which is the equation of an ellipsoid centered at $\underline{x}=\hat{\underline{x}}$. Since the matrix $\underline{\underline{A}}^T \underline{\underline{S}}^{-2} \underline{\underline{A}}$ is not a diagonal matrix, the axes of that ellipsoid are not aligned with the coordinate axes of the space.

The least squares solution vector $\hat{\underline{x}}$ is said to be a point estimate of the unknown solution vector. For most spectrum unfolding problems it is not, however, a very good estimate because of the basic instabilities of the underlying integral equations problem which were discussed in Chapter 2. Typical $\hat{\underline{x}}$ vectors obtained in this manner have elements which are extremely large in magnitude, and about half of them are negative. Such estimates are physically not acceptable. The FERDO and FERD methods modify this standard least squares procedure in order to take into account a non-negativity constraint. Before discussing those modifications, however, it is instructive to consider how one might obtain interval estimates from the least squares method.

For convenience, let us temporarily drop the subscript k from the window vector w_k and its corresponding response ϕ_k. If $\underline{x}$ is inside the confidence ellipsoid described by Eq. (6-4) then $\phi = \underline{w}^T \underline{x}$ must be in the interval defined by

$$\phi^{lo} = \min_{\underline{x}} \left\{ \underline{w}^T \underline{x} \;\middle|\; (\underline{x} - \hat{\underline{x}})^T \underline{\underline{A}}^T \underline{\underline{S}}^{-2} \underline{\underline{A}} (\underline{x} - \hat{\underline{x}}) \leq \mu^2 - \rho_o \right\},$$

$$\phi^{up} = \max_{\underline{x}} \left\{ \underline{w}^T \underline{x} \;\middle|\; (\underline{x} - \hat{\underline{x}})^T \underline{\underline{A}}^T \underline{\underline{S}}^{-2} \underline{\underline{A}} (\underline{x} - \hat{\underline{x}}) \leq \mu^2 - \rho_o \right\}.$$

Since $\underline{w}$ is an n-vector it can be plotted in x-space as shown in Fig. 6.1 for the same problem shown in Fig. 5.2. The surfaces of constant value

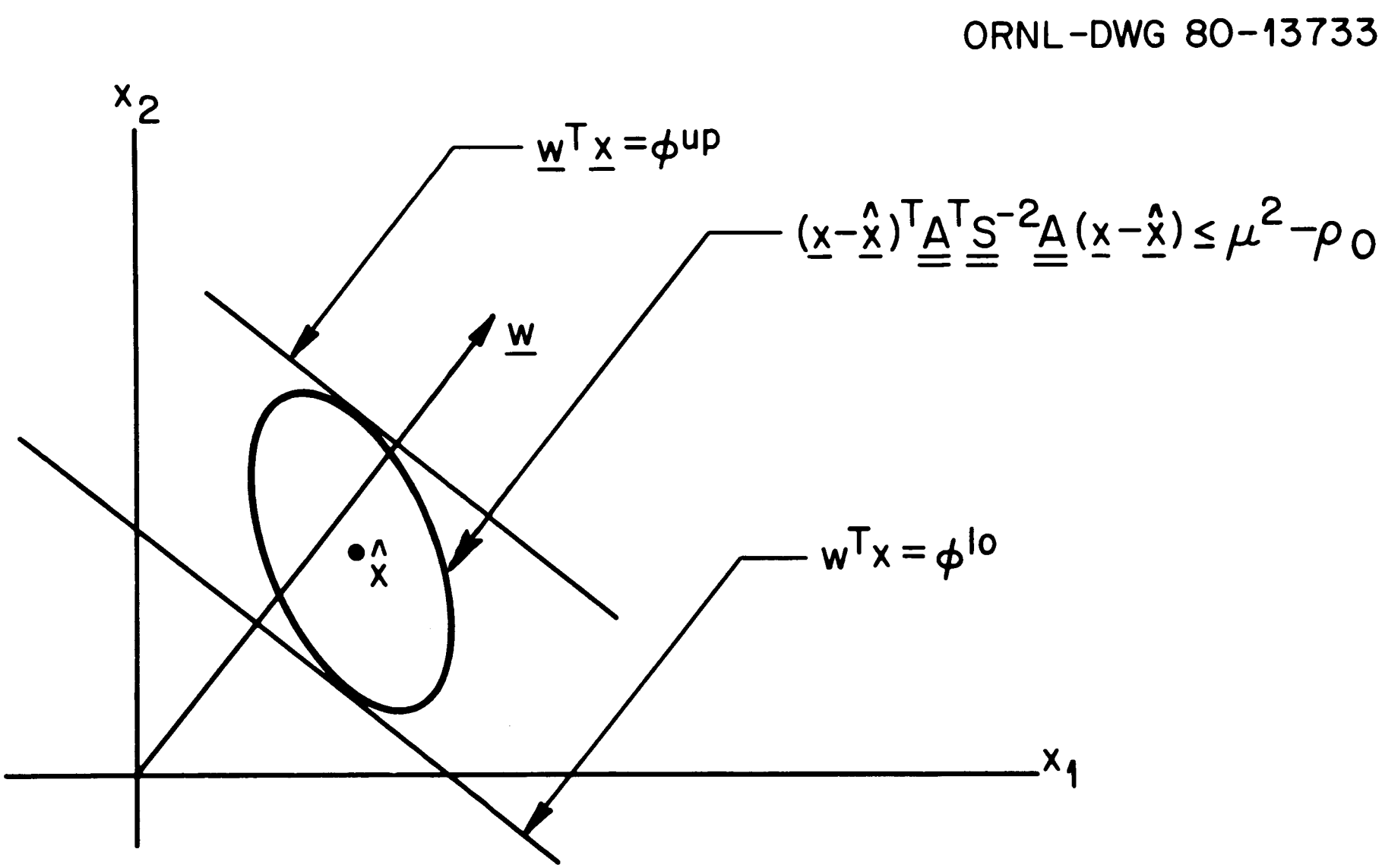


Fig. 6.1. Confidence Interval Estimates for $\phi = \underline{w}^T\underline{x}$.

for the function $\phi = \underline{w}^T\underline{x}$ are just hyperplanes orthogonal (perpendicular) to the vector $\underline{w}$. The two of these planes that are just tangent to the surface of the ellipsoid define the values ϕ^{lo} and ϕ^{up}. Since it is the tangent planes that define the desired interval, we can drop the inequality in the constraints and write the interval estimation problems as

$$\phi^{\left\{\begin{matrix}lo\\up\end{matrix}\right\}} = \underset{\underline{x}}{\left\{\begin{matrix}\min\\\max\end{matrix}\right\}} \left\{ \underline{w}^T\underline{x} \;\middle|\; (\underline{x} - \underline{\hat{x}})^T\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}}(\underline{x} - \underline{\hat{x}}) = \mu^2 - \rho_o \right\}.$$

These problems can be readily solved by the method of Lagrange multipliers to give

$$\begin{aligned} \phi^{lo} &= \underline{w}^T\underline{\hat{x}} - \left(\sqrt{\mu^2 - \rho_o}\right)\sqrt{\underline{w}^T(\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}})^{-1}\underline{w}} \\ \phi^{up} &= \underline{w}^T\underline{\hat{x}} + \left(\sqrt{\mu^2 - \rho_o}\right)\sqrt{\underline{w}^T(\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}})^{-1}\underline{w}} \end{aligned} \qquad (6\text{-}5)$$

The quantity $\underline{w}^T\underline{\hat{x}}$ is just the least squares point estimate of ϕ, and the confidence limits are obtained from it by adding and subtracting the quantity $\left(\sqrt{\mu^2 - \rho_o}\right)\sqrt{\underline{w}^T(A^TS^{-2}A)^{-1}w}$. Actually, since ϕ is a scalar function of $x_1, x_2, \ldots, x_n$, the confidence interval obtained by this method is wider than it really needs to be. This is because the confidence ellipsoid obtained by choosing μ^2 from the Chi-squared distribution (as described in Chapter 4) gives simultaneous confidence intervals for each of the x_i separately. Each of those intervals is wide enough so that all of them are simultaneously guaranteed with the given confidence level. Each of them could be smaller and still have the same degree of confidence if it were only required to be a confidence interval for the corresponding x_i alone without regard to the other components of $\underline{x}$. Similarly the interval for $\phi = \underline{w}^T\underline{x}$ alone does not need to be based on the full confidence ellipsoid. We shall see in the next chapter how to use the normal distribution, rather than the Chi-square, in order to obtain a narrower confidence interval for ϕ.

7. UNBIASED ESTIMATION

In order to explain the basic idea of the FERDO and FERD methods we will briefly consider the least squares interval estimation problem from another point of view. Again for convenience we drop the subscript k from the window vector and seek to estimate the quantity $\phi = \underline{w}^T \underline{x}$ using the measured count vector $\underline{\bar{b}}$. We seek a linear estimator of the form

$$\hat{\phi} = \sum_{i=1}^{m} u_i \bar{b}_i = \underline{u}^T \underline{\bar{b}}, \tag{7-1}$$

where the u_i are estimation coefficients to be determined. A very desirable property of such an estimator is that it be unbiased. That is, if the measurements are repeated many times, with the estimate $\hat{\phi}$ calculated each time, then the average value of all the $\hat{\phi}$'s thus obtained should approach the true but unknown value of ϕ. Using the expectation operator E we can write this criterion more succinctly as

$$E(\hat{\phi}) = \phi, \tag{7-2}$$

or as

$$E(\underline{u}^T \underline{b}) = \underline{w}^T \underline{x}. \tag{7-3}$$

Now

$$E(\underline{u}^T \underline{\bar{b}}) = \underline{u}^T E(\underline{\bar{b}}) = \underline{u}^T \underline{b}_o \tag{7-4}$$

where $\underline{b}_o$ is the unknown, errorless count vector corresponding exactly to the true solution $\underline{x}$, i.e.

$$\underline{b}_o = \underline{\underline{A}}\,\underline{x} \tag{7-5}$$

Combining (7-3,4,5) we can write the condition for an unbiased estimator as

$$\underline{w}^T \underline{x} = \underline{u}^T \underline{\underline{A}}\,\underline{x},$$

so that any vector $\underline{u}$ satisfying

$$\underline{u}^T \underline{\underline{A}} = \underline{w}^T \tag{7-6}$$

will produce an unbiased estimator $\hat{\phi}$.

Now since $\underline{u}$ is an m-vector, $\underline{w}$ is an n-vector, and we are assuming that $m > n$, Eq. (7-6) is a system of linear equations with more unknowns than equations. That means that there are infinitely many vectors $\underline{u}$ which solve the system and provide unbiased estimators. From all of those unbiased estimators we would like to pick the one most stable with respect to the random measuring error in $\underline{\bar{b}}$, because we would like to minimize the departure from the true value ϕ caused by the departure of $\underline{\bar{b}}$ from the true $\underline{b}_o$. The variance of the estimator $\hat{\phi}$ is given by

$$V(\hat{\phi}) = E[(\hat{\phi} - \phi)^2] = E[(\underline{u}^T\underline{\bar{b}} - \underline{w}^T\underline{x})^2]$$

$$= E[(\underline{u}^T\underline{\bar{b}} - \underline{u}^T\underline{b}_o)^2] = E\{[\underline{u}^T(\underline{\bar{b}} - \underline{b}_o)]^2\}$$

$$= E[\underline{u}^T(\underline{b} - \underline{b}_o)\underline{u}^T(\underline{b} - \underline{b}_o)]$$

$$= E[\underline{u}^T(\underline{\bar{b}} - \underline{b}_o)(\underline{\bar{b}} - \underline{b}_o)^T\underline{u}]$$

$$= \underline{u}^T E[(\underline{\bar{b}} - \underline{b}_o)(\underline{\bar{b}} - \underline{b}_o)^T]\underline{u} = \underline{u}^T\underline{\underline{\Sigma}}^2\underline{u},$$

where $\underline{\underline{\Sigma}}^2$ is the variance-covariance matrix for the measuring error. Since we don't know that matrix, it is necessary to use $\underline{\underline{S}}^2$ in its place and thus to approximate the variance of $\hat{\phi}$ by

$$V(\hat{\phi}) = \underline{u}^T\underline{\underline{S}}^2\underline{u}. \tag{7-7}$$

The problem then is to select from all the $\underline{u}$ that satisfy the constraint (7-6), the one which minimizes this variance. That $\underline{u}$ can easily be determined by the method of Lagrange multipliers to be

$$\underline{u} = \underline{\underline{S}}^{-2}\underline{\underline{A}}(\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}})^{-1}\underline{w}, \tag{7-8}$$

and the resulting estimator $\hat{\phi} = \underline{u}^T\underline{\bar{b}}$ is called the minimum variance unbiased estimator or sometimes the best linear unbiased estimator. It turns out to be the same as the least squares estimate since

$$\hat{\phi} = \underline{u}^T\underline{\bar{b}}$$

$$= \underline{w}^T(\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}})^{-1}\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\bar{b}} \tag{7-9}$$

$$= \underline{w}^T\hat{\underline{x}} \ ,$$

where $\hat{\underline{x}}$ is the least squares estimate of $\underline{x}$ given by Eq. (6-2).

The above expression for the best linear unbiased estimator does not assume anything about the probability distribution for the b_i. We are assuming that the $\bar{b}_i$ are independently normally distributed so it follows that $\phi=u^T\underline{\bar{b}}$ is normally distributed also with variance given by Eq. (7-7). The standard method for contructing confidence intervals for a normally distributed variable is given in most intermediate level statistics texts [cf. Mood & Graybill, 1963, Chapter 11.] The confidence limits are calculated from

$$\hat{\phi}^{lo} = \underline{u}^T\underline{\bar{b}} - \kappa \sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}}$$

$$\hat{\phi}^{up} = \underline{u}^T\underline{\bar{b}} + \kappa \sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}} \qquad (7\text{-}10)$$

where κ is a parameter whose value is determined by the probability level to be associated with the confidence level α and the value of κ is given by

$$\alpha = \frac{1}{\sqrt{2\pi}} \int_{-\kappa}^{\kappa} \exp[-\nu^2/2]d\nu. \qquad (7\text{-}11)$$

This relationship is tabulated in standard tables of the normal distribution and can be found in almost any book on statistics. The following short table gives some representative values:

α	0.68	0.90	0.95	0.99
κ	1.0	1.64	1.96	2.58

(7-12)

Comparing Eqs. (7-7) and (7-10) shows that the confidence interval $[\phi^{lo}, \phi^{up}]$ is constructed from the point estimate $u^T\underline{\bar{b}}$ and its standard deviation $\sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}}$. The FERDO and FERD codes are designed to give 68 per cent confidence intervals (i.e. plus and minus one standard deviation) so the appropriate value of κ would be 1. That is, if physically realistic intervals could be obtained for unfolding problems by the standard interval estimation technique outlined above, then the appropriate expressions for ϕ^{lo} and ϕ^{up} would be

$$\hat{\phi}^{lo} = \underline{u}^T\underline{b} - \sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}}\,,$$
$$\hat{\phi}^{up} = \underline{u}^T\underline{b} + \sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}}\,. \qquad (7\text{-}13)$$

As we shall see in the next chapter, intervals calculated from these equations for spectral unfolding problems are far too wide to give any useful information about the underlying spectrum. Before doing that, however, it is interesting to substitute the expression (7-8) for $\underline{u}$ into the equations (7-10) for the interval estimates. The final results are

$$\hat{\phi}^{lo} = \underline{w}^T\underline{\hat{x}} - \kappa\sqrt{\underline{w}^T(\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}})^{-1}\underline{w}}\,,$$
$$\hat{\phi}^{up} = \underline{w}^T\underline{\hat{x}} + \kappa\sqrt{\underline{w}^T(\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}})^{-1}\underline{w}}\,, \qquad (7\text{-}14)$$

which differ from those in Eq. (6-5) only in the value of the confidence level factor which multiples the standard deviation term. Intervals obtained from (7-14), which uses the normal distribution to determine κ from α, are narrower than those obtained from (6-5), which uses the Chi-square distribution to obtain μ^2 from α. As we shall see the FERDO codes do not use either of these methods exactly as they are explained in the preceding, because the intervals thus obtained would be too wide to be useful. As we stated at the end of Chapter 4, it is necessary to add the a priori physical constraint that $\underline{x}$ be non-negative in order to obtain reasonable estimates. Adding these constraints requires modification of both of the above described interval estimation procedures.

8. THE EFFECT OF ILL-CONDITIONING

The interval estimation procedure described in the preceding section is illustrated graphically in Fig. 8.1 where the independent variable ϕ is taken as the random variable for the associated normal probability density function. The estimate $\hat{\phi}$ of Eq. (7-9) is taken as the mean of the distribution and the variance is given by (7-7). The confidence interval, which is obtained from Eqs. (7-10), has a width $2\kappa\sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}}$ and an associated probability α of containing the true value of ϕ. That is, if the count vector is measured many times, and the corresponding estimates $\hat{\phi}$, $\hat{\phi}^{lo}$, $\hat{\phi}^{up}$ are computed each time, then approximately $100\alpha\%$ of the resulting intervals $[\hat{\phi}^{lo}, \hat{\phi}^{up}]$ will contain the true ϕ.

The situation illustrated in Fig. 8.1 applies to a well-conditioned problem. The value of $\hat{\phi}$ is positive and the interval $[\hat{\phi}^{lo}, \hat{\phi}^{up}]$ is well-separated from $\phi=0$. Unfortunately, most spectrum unfolding problems do not give such nice results. In Chapter 2 we described the basic difficulty of the unfolding problem which is the fact that the solution spectrum $x(E)$ is a highly unstable function of the measured count vector $\underline{b}$. This instability carries over to the discrete, vector-matrix estimation problem and manifests itself in two related ways: (1) the estimate $\hat{\phi}$ is a highly unstable function of $\underline{b}$, and (2) the variance $\kappa\sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}}$ becomes quite large so that the confidence interval $[\hat{\phi}^{lo}, \hat{\phi}^{up}]$ becomes very wide. Such problems are said to be ill-conditioned, and typically are similar to the one illustrated in Fig. 8.2 which shows a case in which the confidence interval does contain the true ϕ. The latter quantity is positive, as would be expected from physical considerations, but the estimate $\hat{\phi}$ is negative. If the measurements were repeated many times, then the resulting $\hat{\phi}$ might turn out to be positive some of the time, but the intervals $[\hat{\phi}^{lo}, \hat{\phi}^{up}]$ always turns out to be so wide, with $\hat{\phi}^{lo}$ almost always negative, that they yield almost no useful information about the true ϕ.

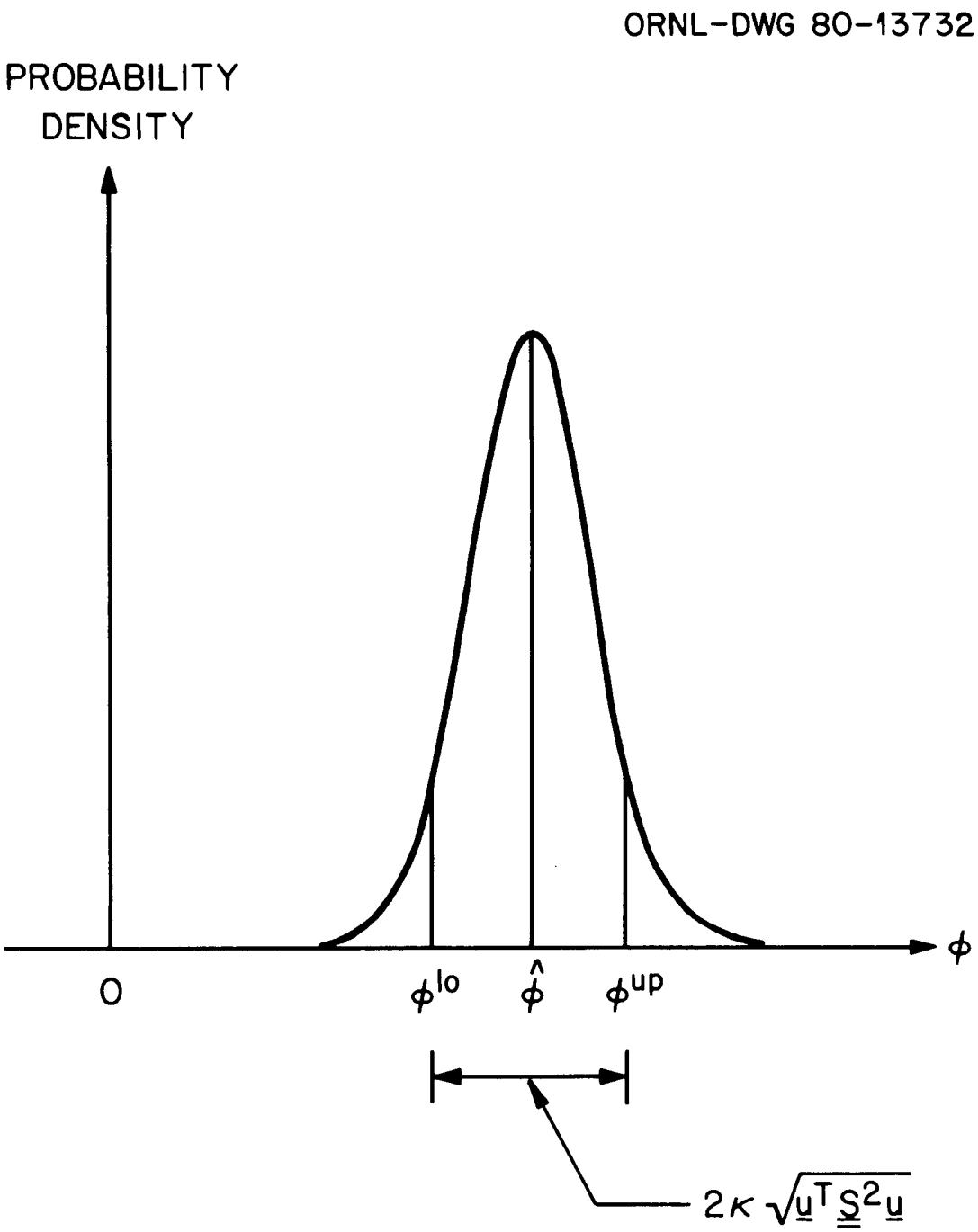


Fig. 8.1. Classical Interval Estimation For a Well-Conditioned Problem.

ORNL-DWG 80-13731

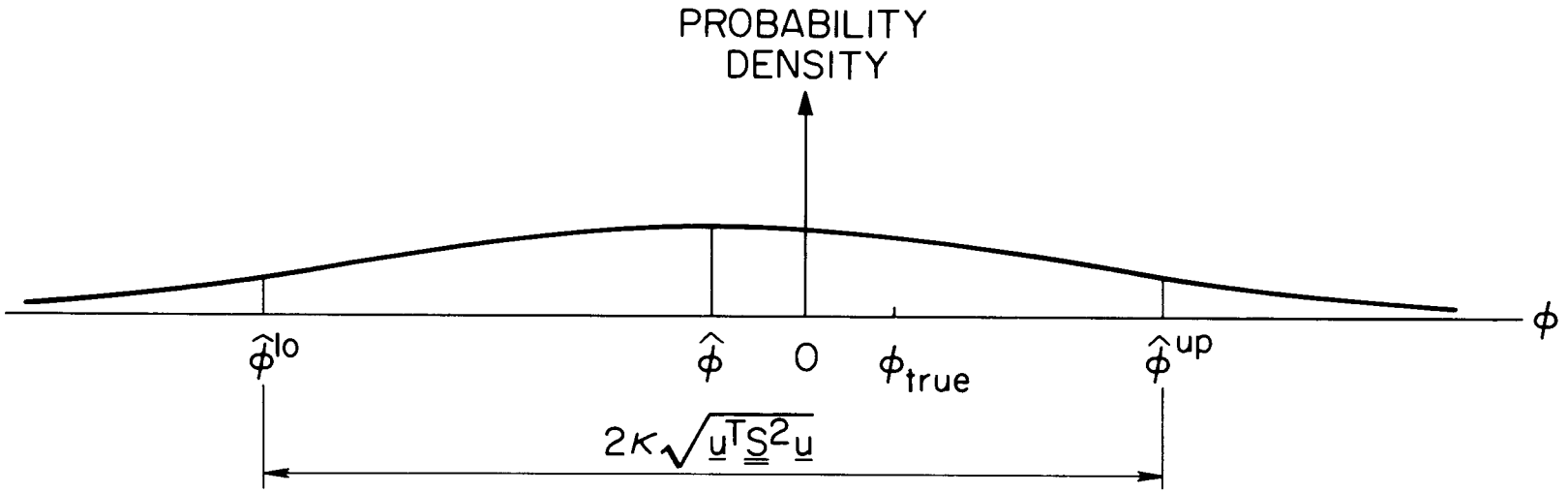


Fig. 8.2. Interval Estimation for an Ill-Conditioned Problem.

It is interesting to examine the geometry of the confidence ellipsoids discussed in Chapters 5 and 6 when the problem becomes ill-conditioned. This situation is illustrated in Fig. 8.3 which should be compared to Fig. 6.1. When the estimation problem becomes ill-conditioned, the confidence ellipsoid in x-space becomes greatly elongated in some directions, and the support planes become widely separated, unless the window vector is exactly orthogonal (perpendicular) to all of those directions. Since this almost never happens, the corresponding interval estimates for the window responses will all be too wide to be useful.

The reason for the elongation of the confidence ellipsoids in some directions is that the columns of the vector $\underline{\underline{A}}$ become nearly linearly dependent. We have been assuming that they span a n-dimensional subspace of the m-dimensional space of count vectors. The column vectors form a basis (set of coordinate axes) for this subspace. They are not mutually orthoganal vectors so they form a skewed coordinate system. When the problem is ill-conditioned this skewing effect is very pronounced and the vectors almost fail to define a full n-dimensional subspace. In some problems the columns of $\underline{\underline{A}}$ may, in fact, actually fail to span the full n-dimensions. In these cases the ellipsoids become open-ended and infinitely long in some directions. In such cases the least squares estimation procedures described in the preceding two chapters break down because the matrix $(\underline{\underline{A}}^T \underline{\underline{S}}^{-2} \underline{\underline{A}})$ is singular, i.e. (fails to have an inverse). The methods can be fixed up by using the <u>generalized inverse</u> matrix but the intervals obtained all turn out to be $(-\infty, +\infty)$ unless the window vector just happens to lie completely in the subspace that is spanned by the columns of $\underline{\underline{A}}$ -- a very rare occurence. Fortunately the FERDO and FERD methods will work in such cases as well as in the ill-conditioned full rank case. We will describe these methods in the next two chapters.

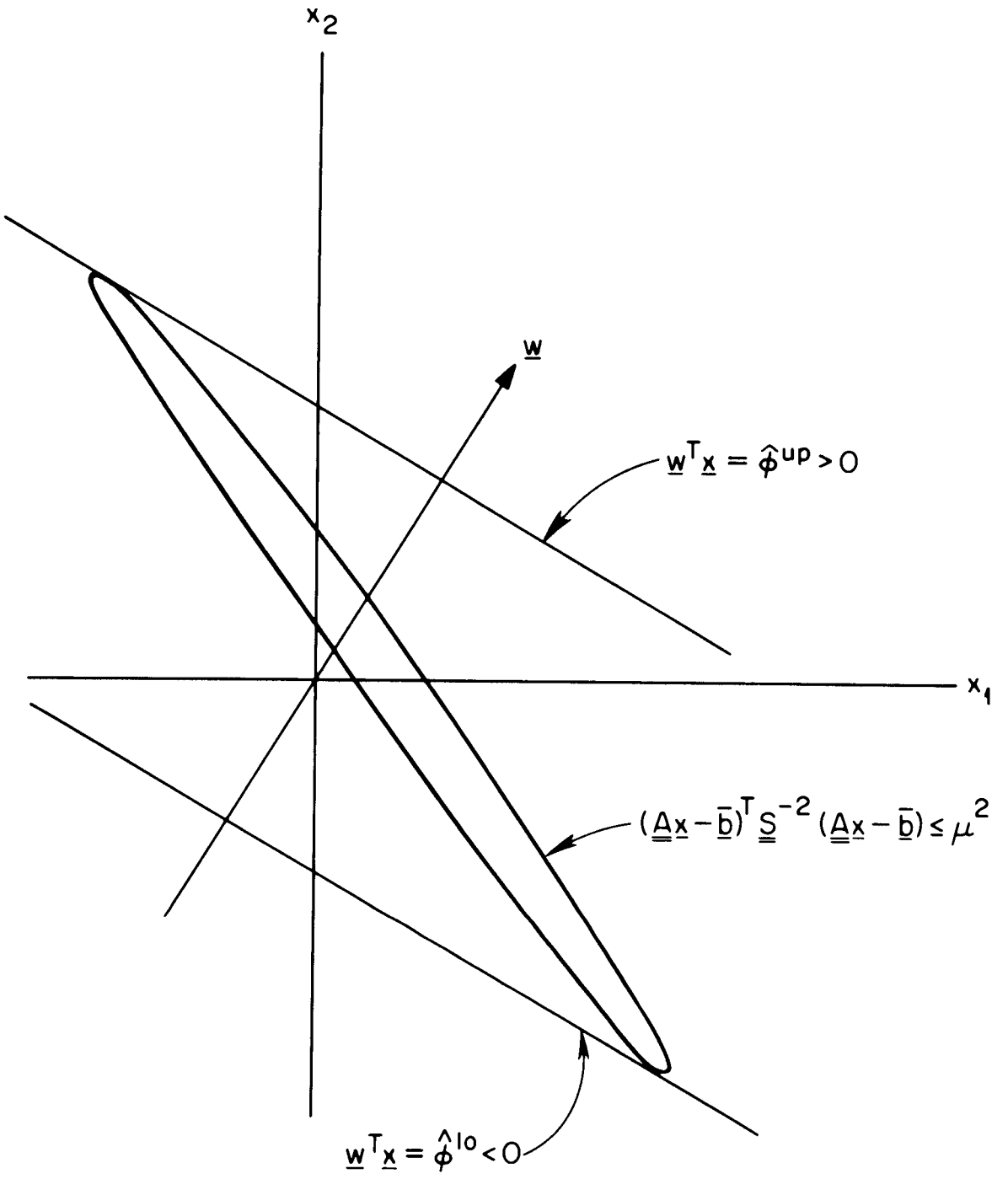


Fig. 8.3. Confidence Interval Estimates for an Ill-Conditioned Problem.

9. THE FERDO METHOD

The FERDO method avoids the problem of the instability of the best linear unbiased estimator by seeking a biased estimator

$$\tilde{\phi} = \tilde{\underline{u}}^T \underline{\bar{b}} \ , \tag{9-1}$$

i.e. an estimator such that

$$E(\tilde{\phi}) \neq \underline{w}^T \underline{x} \quad .$$

The bias ρ is defined by

$$\rho = \underline{w}^T \underline{x} - E(\tilde{\phi}) \ . \tag{9-2}$$

Now, since

$$E(\tilde{\phi}) = E(\tilde{\underline{u}}^T \underline{\bar{b}}) = \tilde{\underline{u}}^T E(\underline{\bar{b}})$$
$$= \tilde{\underline{u}}^T \underline{b}_o = \tilde{\underline{u}}^T \underline{\underline{A}} \underline{x} \ ,$$

the bias can also be written

$$\rho = (\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}}) \ \underline{x} \ . \tag{9-3}$$

We have not yet specified the vector $\tilde{\underline{u}}$, and before doing so we will derive the expressions for the bounds $\tilde{\phi}^{lo}$ and $\tilde{\phi}^{up}$ of the corresponding confidence interval. Since $\underline{\bar{b}} \sim N(\underline{b}_o, \underline{\underline{S}}^2)$ it follows that $\tilde{\phi} \sim N(\underline{u}^T \underline{b}_o, \tilde{\underline{u}}^T \underline{\underline{S}}^2 \tilde{\underline{u}})$ and that for a given confidence level α we can pick κ according to Eqs. (7-11, 12) so that

$$\Pr \left\{ \tilde{\phi} - \kappa \sqrt{V(\tilde{\phi})} \leq E(\tilde{\phi}) \leq \tilde{\phi} + \kappa \sqrt{V(\tilde{\phi})} \right\} \geq \alpha, \tag{9-4}$$

where

$$V(\tilde{\phi}) = \tilde{\underline{u}}^T \underline{\underline{S}}^2 \tilde{\underline{u}} \tag{9-5}$$

is just the variance of the estimator $\tilde{\phi}$. By (9-2), $E(\tilde{\phi}) = \underline{w}^T\underline{x} - \rho$, so we can rewrite the above confidence interval statement in the form

$$\Pr\left\{\underline{\tilde{u}}^T\underline{b} - \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}} \leq \underline{w}^T\underline{x} - \rho \leq \underline{\tilde{u}}^T\underline{b} + \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}}\right\} \geq \alpha$$

or

$$\Pr\left\{\underline{\tilde{u}}^T\underline{b} + \rho - \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}} \leq \underline{w}^T\underline{x} \leq \underline{\tilde{u}}^T\underline{b} + \rho + \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}}\right\} \geq \alpha .$$

A change in this statement which might cause the confidence interval to be wider, but never smaller, will certainly preserve the desired confidence level α. In particular, since $-|\rho| \leq \rho$ and $\rho \leq |\rho|$, we can write

$$\Pr\left\{\underline{\tilde{u}}^T\underline{b} - \left(|\rho| + \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}}\right) \leq \underline{w}^T\underline{x} \leq \underline{\tilde{u}}^T\underline{b} + \left(|\rho| + \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}}\right)\right\} \geq \alpha ,$$

or by (9-3)

$$\Pr\left\{\underline{\tilde{u}}^T\underline{b} - \left[\left|(\underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}})\underline{x}\right| + \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}}\right] \leq \phi \leq \underline{\tilde{u}}^T\underline{b} + \left[\left|(\underline{w}^T - \underline{u}^T\underline{\underline{A}})\underline{x}\right| + \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}}\right]\right\} \geq \alpha .$$

We have just shown that for any m-vector $\underline{\tilde{u}}$, a valid α-level confidence interval for $\phi = \underline{w}^T\underline{x}$ is given by the expressions

$$\begin{aligned}\tilde{\phi}^{lo} &= \underline{\tilde{u}}^T\underline{b} - \left[\kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}} + |(\underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}})\underline{x}|\right] ,\\ \tilde{\phi}^{up} &= \underline{\tilde{u}}^T\underline{b} + \left[\kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}} + |(\underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}})\underline{x}|\right] ,\end{aligned} \qquad (9\text{-}6)$$

when κ is determined by α according the relation (7-11) for normally distributed variables. It is easy to see that when $\underline{\tilde{u}}$ is chosen to give an unbiased estimator, i.e. from Eq. (7-8), then these bounds reduce to those of Eqs. (7-10). Since the bounds computed from the latter equations are too wide to be useful, the FERDO method seeks instead to determine $\underline{\tilde{u}}$ to minimize the width of the confidence interval obtained from the expressions (9-6). That width is just twice the quantity

$$L(\underline{\tilde{u}}) = \kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}} + |(\underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}})\underline{x}| \qquad (9\text{-}7)$$

which is just the sum of a variance term and a bias term. The idea of the FERDO method is to accept a non-zero bias in the estimate, in return for a reduction in the variance, in the hope that these two combined uncertainties will be less than the variance uncertainty for the unbiased estimator. The idea is illustrated in Fig. 9.1 which should be compared to Fig. 8.2. The former is drawn for a case in which the unknown quantity $E(\tilde{\phi}) = \underline{\tilde{u}}^T \underline{b}_o$ lies inside the α-level confidence interval about $\tilde{\phi}$ so that the unknown $\phi_{true} = \underline{w}^T \underline{x}$ lies inside the interval $[\tilde{\phi}^{lo}, \tilde{\phi}^{up}]$ as it should for at least $100\alpha\%$ of the measured count vectors.

Although the basic FERDO idea is to choose $\underline{\tilde{u}}$ to minimize the expression (9-7) and then to compute the confidence interval bounds by Eqs. (9-6), the code cannot actually do this because the vector $\underline{x}$ is unknown. To get around this difficulty we first note that since we know *a priori* that $\underline{x} \geq \underline{0}$, it is true that

$$|(\underline{w}^T - \underline{\tilde{u}}^T \underline{\underline{A}})\underline{x}| \leq |\underline{w}^T - \underline{\tilde{u}}^T \underline{\underline{A}}|\,\underline{x} \qquad (9\text{-}8)$$

The next step is to replace $\underline{x}$ by an upper bound that can be calculated from known quantities. Since the elements of $\underline{\underline{A}}$ are probabilities, it must satisfy $\underline{\underline{A}} \geq \underline{\underline{0}}$ where $\underline{\underline{0}}$ is the m x n zero-matrix. It follows then that for all i and j $(1 \leq i \leq m,\ 1 \leq j \leq n)$

$$A_{ij}\ x_j \leq \sum_{k=1}^{n} A_{ik} x_k = (\underline{\underline{A}}\underline{x})_i ,$$

so

$$x_j \leq \frac{(\underline{\underline{A}}\,\underline{x})_i}{A_{ij}}$$

Since this last equality holds for all i, it must hold for the particular i that minimizes the quantity on the right, i.e.,

ORNL-DWG 80-13729R

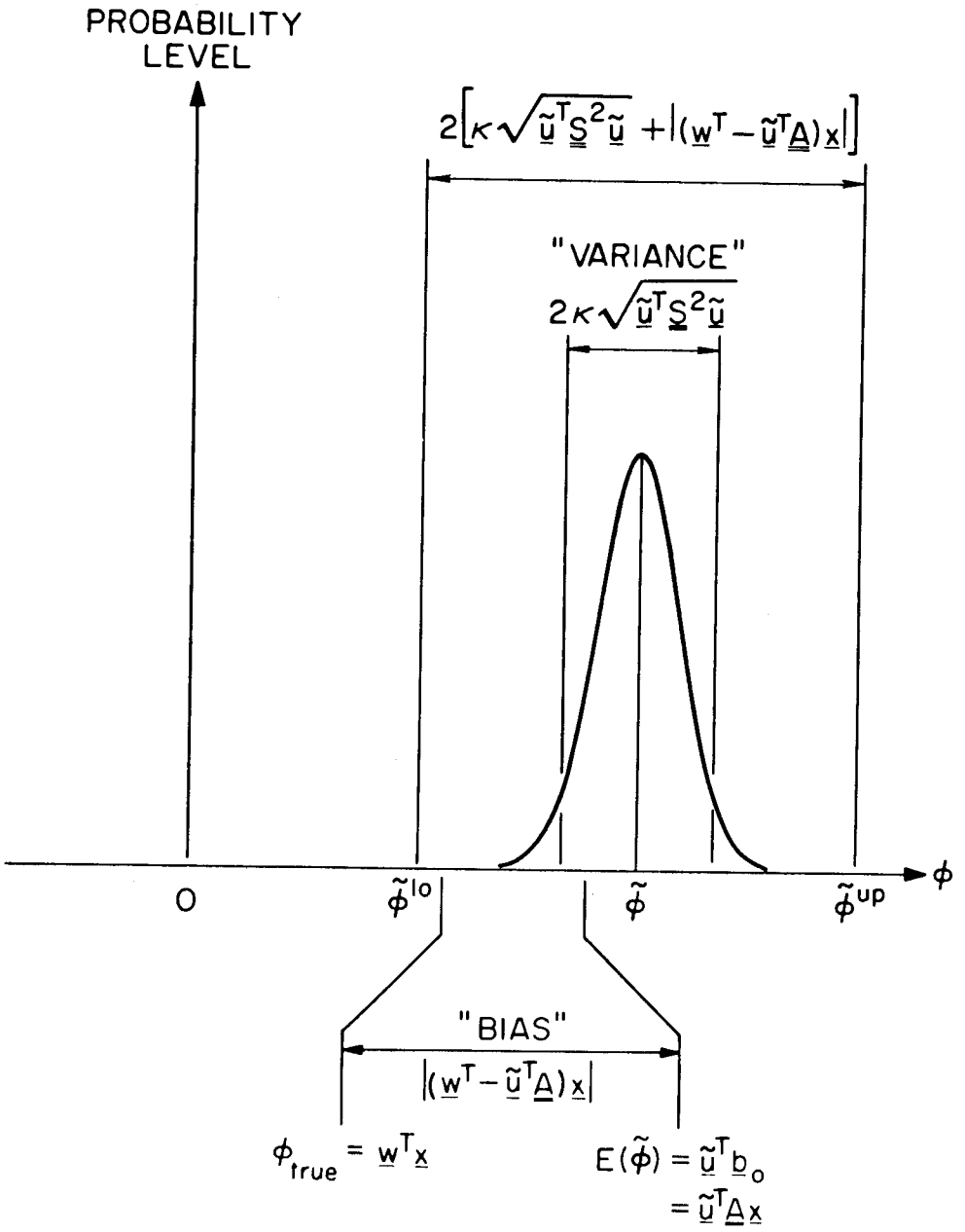


Fig. 9.1. The FERDO Interval Estimation Method

$$x_j \leq \min_{1 \leq i \leq m} \left\{ \frac{(\underline{\underline{A}}\underline{x})_i}{A_{ij}} \right\} , \quad j=1,\cdots,n . \qquad (9\text{-}9)$$

In order to turn this into a useable upper bound we must get rid of the $(\underline{\underline{A}}\underline{x})_i$ term. At this point it is necessary to make an approximation involving the confidence ellipsoids discussed in Chapters 4 and 5.

For any confidence level α', we can find, by means of the $\chi^2(m)$ relationship in Eq. (4-3), a corresponding constant μ such that

$$\Pr \left\{ (\underline{\underline{A}}\underline{x} - \underline{\bar{b}})^T \underline{\underline{S}}^{-2} (\underline{\underline{A}}\underline{x} - \underline{\bar{b}}) \leq \mu^2 \right\} \geq \alpha' .$$

Note that we use α' here to distinguish it from the probability level α corresponding to the confidence interval we are seeking to estimate. We want to use the ellipsoidal constraint

$$(\underline{\underline{A}}\underline{x} - \underline{\bar{b}})^T \underline{\underline{S}}^{-2} (\underline{\underline{A}}\underline{x} - \underline{\bar{b}}) \leq \mu^2 \qquad (9\text{-}10)$$

to get bounds on the elements $(\underline{\underline{A}}\underline{x})_i$, so ideally we would like to take $\alpha'=1$ in order to assure that the resulting bounds are 100% guaranteed. Unfortunately $\alpha'=1$ gives $\mu=\infty$ and the resulting bounds on $\underline{\underline{A}}\underline{x}$ are not very helpful. The approximation that must be made is to set α' to a value very near 1, determine the corresponding μ, and then assume that the resulting confidence ellipsoid is really a 100% ellipsoid. The FERDO code uses $\alpha'=.9995$ and automatically computes the value of μ corresponding to α' and m.

Assuming that the inequality (9-10) holds with 100% certainty we can rewrite it as

$$\sum_{k=1}^{m} \frac{(\underline{\underline{A}}\underline{x} - \underline{\bar{b}})_k^2}{s_k^2} \leq \mu^2$$

from which it certainly follows that for all i

$$(\underline{\underline{A}}\underline{x} - \underline{\bar{b}})_i \leq \mu s_i ,$$

and hence that

$$(\underline{\underline{A}}\underline{x})_i \leq \bar{b}_i + \mu s_i, \quad i=1,\cdots,m. \tag{9-11}$$

This inequality, together with (9-9) gives

$$x_j \leq \min_{1\leq i\leq m} \left\{ \frac{\bar{b}_i + \mu s_i}{A_{ij}} \right\}, \quad j=1,\cdots,n, \tag{9-12}$$

which gives computable upper bounds for the x_j. Although these bounds cannot in theory be rigorously guaranteed with 100% certainty, in practice they always turn out to be more than large enough. The main reason is the two steps of the derivation in which a sum was replaced by a single term, i.e.,

$$A_{ij}\, x_j \leq \sum_{k=1}^{n} A_{ik} x_k = (\underline{\underline{A}}\underline{x})_i ,$$

and

$$\frac{(\underline{\underline{A}}\underline{x} - \underline{\bar{b}})_i^2}{s_i^2} \leq \sum_{k=i}^{m} \frac{(\underline{\underline{A}}\underline{x} - \underline{\bar{b}})_k^2}{s_k^2} \leq \mu^2 .$$

In real unfolding problems these two steps introduce so much slack into the final bounds that the statistical contribution from Eq. (9-11) is small by comparison, even for very large values of α' and μ.

If we define the quantities q_j by

$$q_j = \min_{1\leq i\leq m} \left\{ \frac{\bar{b}_i + \mu s_i}{A_{ij}} \right\}, \quad j=1,\cdots,n, \tag{9-13}$$

and the matrix $\underline{Q}$ by

$$\underline{Q} \equiv \text{diag}\ (q_1, q_2, \ldots, q_n)\ , \tag{9-14}$$

then the inequalities (9-12) can be written in vector-matrix terms as

$$\underline{x} \leq \underline{Q}\ \underline{e}, \tag{9-15}$$

where the $\underline{e}$ is the n-vector of all 1's,

$$\underline{e} \equiv (1,1,\cdots,1)^T. \tag{9-16}$$

Combining the vector inequalities (9-15) and (9-8) gives

$$|\ (\underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}})\underline{x}\ | \leq |\ \underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}}\ |\underline{Qe}\ . \tag{9-16}$$

We can now replace the expressions (9-6) for the confidence interval bounds by

$$\begin{aligned} \tilde{\phi}^{lo} &= \underline{\tilde{u}}^T\underline{b} - \left[\kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}} \quad + \quad |\ \underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}}\ |\ \underline{Q}\ \underline{e}\right]\ , \\ \tilde{\phi}^{up} &= \underline{\tilde{u}}^T\underline{b} + \left[\kappa\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{\tilde{u}}} \quad + \quad |\ \underline{w} - \underline{\tilde{u}}^T\underline{\underline{A}}\ |\ \underline{Q}\ \underline{e}\right]. \end{aligned} \tag{9-17}$$

These are the expressions used by FERDO to calculate the final bounds. The interval which they give is wider than the one defined by Eqs. (9-6), but they have the decided advantage that, once $\underline{\tilde{u}}$ is specified all of the quantities on the right hand sides are known. No matter what $\underline{\tilde{u}}$ is chosen, the above equations give an interval whose associated confidence level is at least α.

The FERDO strategy for choosing $\underline{\tilde{u}}$ is based on the idea of minimizing the quantity (9-7), but it takes into account the fact that $\underline{x}$ is unknown, and that the final bounds are computed from upper bound approximations, by introducing a variable parameter which is then adjusted to make the final bounds as narrow as possible. Specifically, (9-7) is replaced by the expression

$$L_1(\underline{\tilde{u}},\tau) = \tau\sqrt{\underline{\tilde{u}}^T\underline{\underline{S}}^2\underline{u}} \quad + \quad |\ (\underline{w}^T - \underline{\tilde{u}}^T\underline{\underline{A}})\underline{x}\ | \tag{9-18}$$

where τ is a parameter chosen to lie between 0 and ∞. One way to justify this expression is to think of it as a weighted combination of the variance and bias errors with the relative weighting determined by the value chosen for τ. Once τ is chosen the idea is then to choose $\tilde{\underline{u}}$ to minimize this weighted combination. Of course it is still necessary to get rid of the unknown $\underline{x}$ in the expression to be minimized. To do this, first note that

$$|(\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}})\ \underline{x}| = |(\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}})\ \underline{\underline{Q}}\underline{\underline{Q}}^{-1}\underline{x}| \leq \|\underline{\underline{Q}}\ (\underline{w} - \underline{\underline{A}}^T \tilde{\underline{u}})\|\ \|\ \underline{\underline{Q}}^{-1}\ \underline{x}\| ,$$

where $\|a\|$ denotes the length of vector $\underline{a}$. The inequality follows from the definition of the dot product of two vectors. Using the definition of the length of a vector, we can rewrite this expression as

$$|(\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}})\ \underline{x}| \leq [(\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}})\underline{\underline{Q}}^2 (\underline{w} - \underline{\underline{A}}^T \tilde{\underline{u}})]^{1/2} [\underline{x}^T \underline{\underline{Q}}^{-2} \underline{x}]^{1/2}$$

and by Eq. (9-15),

$$[\underline{x}^T \underline{\underline{Q}}^{-2} \underline{x}]^{1/2} \leq [\underline{e}^T \underline{\underline{Q}}\underline{\underline{Q}}^{-2}\underline{\underline{Q}}\underline{e}]^{1/2} = [\underline{e}^T \underline{e}]^{1/2} = \sqrt{n} .$$

Thus we replace Eq. (9-18) with an upper bounding expression

$$L_2(\tilde{\underline{u}},\tau) = \tau \sqrt{\tilde{\underline{u}}^T \underline{\underline{S}}^2 \tilde{\underline{u}}} + (\sqrt{n}) \sqrt{(\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}})\underline{\underline{Q}}^2 (\underline{w} - \underline{\underline{A}}^T \tilde{\underline{u}})} .$$

FERDO does not try to minimize this expression either because of the difficulty of working with the sum of two square roots. Using the scalar inequality

$$(\alpha+\beta) \leq \sqrt{2(\alpha^2+\beta^2)}$$

we have

$$L_2(\tilde{\underline{u}},\tau) \leq \sqrt{2[\tau^2 \tilde{\underline{u}}^T \underline{\underline{S}}^2 \tilde{\underline{u}} + n(\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}})\underline{\underline{Q}}^2 (\underline{w} - \underline{\underline{A}}^T \tilde{\underline{u}})]} .$$

FERDO calculates the vector $\tilde{\underline{u}}$ which minimizes the quality in brackets, i.e.

$$L_3(\tilde{\underline{u}},\tau) = \tau^2 \tilde{\underline{u}}^T \underline{\underline{S}}^2 \tilde{\underline{u}} + n(\underline{w}^T - \tilde{\underline{u}}^T \underline{\underline{A}})\underline{\underline{Q}}^2 (\underline{w} - \underline{\underline{A}}^T \tilde{\underline{u}}).$$

Differentiating this expression with respect to $\tilde{\underline{u}}$ and equating the derivative obtained to zero gives

$$\tilde{\underline{u}} = (\underline{\underline{A}}\underline{\underline{Q}}^2\underline{\underline{A}}^T + \frac{\tau^2}{n}\underline{\underline{S}}^2)^{-1}\underline{\underline{A}}\underline{\underline{Q}}^2\ \underline{w} \quad ,$$

which can also be written

$$\tilde{\underline{u}} = \underline{\underline{S}}^{-2}\underline{\underline{A}}\ (\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}} + \frac{\tau^2}{n}\underline{\underline{Q}}^{-2})^{-1}\ \underline{w} \qquad (9\text{-}19)$$

This is the vector $\tilde{\underline{u}}$ calculated by FERDO which uses it then in Eqs. (9-17) to calculate the bounds for the desired confidence interval. Note the presence of the free parameter τ^2 which can in principle be adjusted to give the narrowest possible interval. Actual practice with many test problems has shown that a wide range of τ-values seems to give equally good results so we have set the τ-value in the program at TAU=1.0. A user can easily alter this value by changing one statement in the Subroutine SETLO.

It is interesting to compare the biased estimator obtained from (9-19) with the unbiased estimator (7-9). From the former expression we have

$$\tilde{\phi} = \tilde{\underline{u}}^T\overline{\underline{b}} = \underline{w}^T(\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\underline{\underline{A}} + \frac{\tau^2}{n}\underline{\underline{Q}}^{-2})^{-1}\underline{\underline{A}}^T\underline{\underline{S}}^{-2}\overline{\underline{b}}$$

which differs from the latter only in the diagonal terms of the inverted matrix. The form of the biased estimator shows that the FERDO method is very similar to the smoothing and regularization methods of Phillips, (1962) and Tikonov (1963) and the ridge regression method of Hoerl and Kennard (1970).

Note that in computing the bounds by Eq. (9-17), FERDO takes κ=1 so that the resulting interval is a 68.26% confidence interval.

10. THE FERD METHOD

The FERD method is really an extension of the FERDO method. For each window vector $\underline{w}_k$ it begins with the corresponding FERDO biased estimator $\tilde{\underline{u}}_k$ of Eq. (9-19) and seeks to improve the corresponding confidence interval $[\tilde{\phi}^{lo}_k, \tilde{\phi}^{up}_k]$ by an iterative sequence of modifications of the $\tilde{\underline{u}}_k$. Again, for convenience, we drop the window vector index k and also the tilde over the $\underline{u}$ and write the initial estimate as

$$\underline{u}^{(o)} = \underline{\underline{S}}^{-2}\underline{\underline{A}}\,(\underline{\underline{A}}^T\,\underline{\underline{S}}^{-2}\,\underline{\underline{A}} + \frac{\tau^2}{n}\,\underline{\underline{Q}}^{-2})^{-1}\,\underline{w}, \tag{10-1}$$

where the superscript (o) denotes the iteration number. Subsequent iterates will have superscripts (1), (2), and so on.

Before describing the iteration we note one other feature of the FERD code which is not shared by FERDO, and that is the ability to take into account uncertainties in the response matrix $\underline{\underline{A}}$. If we use $\underline{\underline{A}}^o$ to denote the estimate of $\underline{\underline{A}}$ then the allowed uncertainties have the form

$$\underline{\underline{A}}^{lo} \equiv \underline{\underline{A}}^o - \delta_1\,\underline{\underline{A}}^o - \delta_2\,\underline{\underline{E}} \leqslant \underline{\underline{A}}^o + \delta_1\,\underline{\underline{A}}^o + \delta_2\,\underline{\underline{E}} \equiv \underline{\underline{A}}^{up} \tag{10-2}$$

where $\underline{\underline{E}}$ is an mxn matrix all of whose element are one and δ_1 and δ_2 are scalar quantities specifying relative and absolute errors in $\underline{\underline{A}}$. Either or both can be taken equal to zero. When they are chosen to be non zero the resulting matrix interval is considered to be a guaranteed interval for $\underline{\underline{A}}$, i.e., guaranteed to contain $\underline{\underline{A}}$ with 100% certainty. The method for treating the matrix errors is based on simple programming tricks and is not a fundamental feature of the FERD iteration. Accordingly we will not explicitly take the matrix uncertainties into account in describing the iteration and will use the notation $\underline{\underline{A}}$ for the response matrix just as we have in the preceding chapters.

In order to motivate the FERD method we restate the estimation problems to be solved:

$$\begin{aligned}\phi^{lo} &= \min_{\underline{x}}\,[\underline{w}^T\underline{x}\;|\,(\underline{\underline{A}}\underline{x}-\underline{\bar{b}})^T\underline{\underline{S}}^{-2}(\underline{\underline{A}}\underline{x}-\underline{\bar{b}}) \leqslant \mu^2,\ \underline{x} \geq \underline{o}],\\ \phi^{up} &= \max_{\underline{x}}\,[\underline{w}^T\underline{x}\;|\,(\underline{\underline{A}}\underline{x}-\underline{\bar{b}})^T\underline{\underline{S}}^{-2}(\underline{\underline{A}}\underline{x}-\underline{\bar{b}}) \leqslant \mu^2,\ \underline{x} \geq \underline{o}].\end{aligned} \tag{10-3}$$

These problem statements differ from the ones of Eq. (4-9) only in the addition of the a-priori non-negativity constraints. The value of μ^2 can be chosen by means of the $\chi^2(m)$ distribution to make the constraint region

$$(\underline{\underline{A}}\underline{x} - \underline{\bar{b}})^T \underline{\underline{S}}^{-2} (\underline{\underline{A}}\underline{x} - \underline{\bar{b}}) \leqslant \mu^2$$

a $100\alpha\%$ confidence ellipsoid [cf. Eq. (4-3)]. Since x represents a physical quantity which cannot be negative, the positive orthant, $\underline{x} \geqslant \underline{o}$, is a 100% confidence region, and its intesection with the confidence ellipsoid is a $100\alpha\%$ confidence region. The extreme values of $\phi = \underline{w}^T\underline{x}$ in this region give the endpoints of a $100\alpha\%$ confidence interval.

It can be shown that the exact solutions of the problems (10-3) could be obtained by a parametric programming technique [cf. Rust and Burrus, 1972, Sections 5.3, 5.4]. This a complicated, expensive procedure which is also subject to numerical instabilities caused by the ill-conditioned nature of the problem. The FERD method avoids these difficulties by seeking a suboptimal approximation to the exact confidence limits. This approximate procedure is applied not to problems (10-3) but rather to two equivalent dual problems. One of the most basic relations of mathematical programming is a duality theorem that allows any given constrained optimization problem to be stated in two different, but equivalent, ways — one way as a constrained maximization and the other as a constrained minimization problem [cf. Rust and Burrus, 1972, section 4.4]. When this theorem is applied to problems (10-3) the resulting dual problems are

$$\begin{aligned} \phi^{lo} &= \max_{\underline{u}} [\underline{u}^T\underline{\bar{b}} - \mu\sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}} \mid \underline{u}^T\underline{\underline{A}} \leqslant \underline{w}^T], \\ \phi^{up} &= \min_{\underline{u}} [\underline{u}^T\underline{\bar{b}} + \mu\sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}} \mid \underline{u}^T\underline{\underline{A}} \geqslant \underline{w}^T]. \end{aligned} \tag{10-4}$$

The n-vector $\underline{x}$ of unknown variables is replaced by an m-vector $\underline{u}$ of dual variables. In general the same vector $\underline{u}$ will not solve both problems.

It is instructive to think of the elements of $\underline{u}$ as coefficients for combining the rows of the response matrix $\underline{\underline{A}}$ in order to approximate

the window vector w, i.e. as the coefficients for combining the m pulse height bin responses in order to model the desired window response. In chapter 7 we showed that the quantity $\hat{\phi} = \underline{u}^T\underline{b}$ is an unbiased estimator of ϕ if $\underline{u}$ is chosen to satisfy Eq. (7-6), i.e.

$$\underline{u}^T\underline{\underline{A}} = \underline{w}^T.$$

If such a strategy were practicable then the same $\underline{u}$ would solve both problems, and the confidence bounds could be calculated from the unbiased estimate by subtracting and adding the product of μ and $\sqrt{\underline{u}^T\underline{\underline{S}}^2\underline{u}}$. The latter quantity is just the standard deviation of the estimator [cf. Eq. (7-7)]. Unbiased estimation is not a practicable technique because the intervals obtained are too wide to be useful. The addition of the $\underline{x} \geq \underline{o}$ constraint produces narrower intervals but requires biased estimation, replacing the above condition for an unbiased estimator with the inequality constraints of Eqs. (10-4). The two constraints

$$\underline{u}^T\underline{\underline{A}} \leqslant \underline{w}^T \text{ and } \underline{u}^T\underline{\underline{A}} \geqslant \underline{w}^T \tag{10-5}$$

can be regarded respectively as lower and upper bias inequalities. The same $\underline{u}$ cannot solve both problems.

In general the initial estimate $\underline{u}^{(o)}$ obtained from Eq. (10-1) will not satisfy either of the bias constraints (10-5). The basic idea of the FERD method is to iteratively readjust $\underline{u}^{(o)}$ until the proper bias inequality is satisfied. Assume for the sake of definiteness that we are seeking the lower bound ϕ^{lo} and that after k steps we obtain $\underline{u}^{(k)}$ which satisfies the lower bias inequality, i.e.,

$$\underline{u}^{(k)T}\,\underline{\underline{A}} \leqslant \underline{w}^T.$$

Then a conservative estimate $\tilde{\phi}^{lo}$ is obtained from

$$\tilde{\phi}^{lo} = \underline{u}^{(k)T}\underline{b} - \mu\sqrt{\underline{u}^{(k)T}\underline{\underline{S}}^2\underline{u}^{(k)}}\,. \tag{10-6}$$

This estimate is conservative since it will always be less than or equal to the maximum value of the corresponding objective function in the first of Eqs. (10-4). Of course there is no guarantee that the above

$\tilde{\phi}$ is larger than the one obtained from the FERDO procedure [i.e. the one obtained by plugging $\underline{u}^{(o)}$ into Eqs. (9-17)], but in most problems to which the procedure has been applied it has done as well or better than the FERDO result. This last statement is subject to one important caveat which will be given at the end of this chapter. Before discussing that, however, we briefly describe the iteration used to obtain the lower biased estimation vector $\underline{u}^{(k)}$.

The initial estimate will in general be neither lower nor upper biased, and the bias discrepancy vector

$$\underline{e}^{(o)^T} = \underline{w}^T - \underline{u}^{(o)^T}\underline{\underline{A}} \tag{10-7}$$

will have both positive and negative components. We need to consider only the latter since the former already satisfy the lower bias inequality. The code picks a particular component JMIN satisfying

$$e^{(o)}_{JMIN} = (\underline{w}^T - \underline{u}^{(o)^T}\underline{\underline{A}})_{JMIN} < 0,$$

and then determines the row $i^* = i^*(JMIN)$ of the matrix $\underline{\underline{A}}$ for which

$$\frac{\bar{b}_{i^*} + \mu S_{i^*}}{A_{i^*,JMIN}} = \min_{1 \leq i \leq m} \frac{\bar{b}_i + \mu S_i}{A_{i,JMIN}} \quad . \tag{10-8}$$

A correction vector $\underline{\delta u}^{(o)}$ is defined by taking all components zero except the i* component which is taken to be

$$\delta u^{(o)}_{i^*} = \frac{e^{(o)}_{JMIN}}{A_{i^*,JMIN}} \quad .$$

The new estimate $\underline{u}^{(1)}$ is just

$$\underline{u}^{(1)} = \underline{u}^{(o)} + \underline{\delta u}^{(o)},$$

and it is easy to see that the new discrepancy vector is just

$$\underline{e}^{(1)^T} = \underline{e}^{(o)^T} - \underline{\delta u}^{(o)^T}\underline{\underline{A}}.$$

Since the elements of $\underline{\underline{A}}$ are non-negative and $e^{(o)}_{JMIN}$ is negative, it follows that

$$\underline{\delta u}^{(o)^T}\underline{\underline{A}} = \frac{e^{(o)}_{JMIN}}{A_{i^*,JMIN}} (A_{i^*,1}, A_{i^*,2}, \ldots, A_{i^*,n})$$

is everywhere nonpositive so the iteration does not change any of the components which already satisifed the desired lower bias inequality. Furthermore, the new JMIN component of the discrepancy is just

$$e^{(1)}_{JMIN} = e^{(o)}_{JMIN} - \frac{e^{(o)}_{JMIN}}{A_{i*,JMIN}} A_{i*,JMIN} = 0$$

so that the iteration has, in fact, given an estimate with one more component satisfying the lower bias inequality. This whole procedure is repeated until, after k steps, the discrepancy vector $\underline{e}^{(k)}$ is everywhere nonpositive, and the vector $\underline{u}^{(k)}$ gives a lower biased estimator. To get an upper biased estimate for computing $\tilde{\phi}^{up}$, one begins with the same $\underline{u}^{(o)}$ and applies a similar iteration to annihilate the negative elements of the discrepancy vector.

There is absolutely no guarantee that the iterative procedure described above will give an interval estimate $[\phi^{lo},\phi^{up}]$ that is narrower than the one obtained from the FERDO program. Also the particular strategies used by FERD in choosing JMIN and i* are by no means the only possible choices, and there is no guarantee that they are the best choices. A brief discussion of this strategy can be found in Burrus, et. al., (1980). The best justification that we can presently give for the procedure is that it has worked quite well on a large number of problems in the last ten years or so. FERD users should be aware, however, of the following fact: The FERD code assumes the value $\mu = 1$ and further that this value gives 68% confidence intervals $[\tilde{\phi}^{lo},\tilde{\phi}^{up}]$. This last assertion would follow directly if the quantity $(\underline{\bar{b}} - \underline{\underline{A}}\ \underline{x})^T\ \underline{\underline{S}}^{-2}\ (\underline{\bar{b}} - \underline{\underline{A}}\ \underline{x})$ were normally distributed, but we saw in Chapt. 4 that it follows a $\chi^2(m)$ distribution. This distinction has been discussed at greater length in Chapter 6 of Rust and Burrus, 1972. For the χ^2 distributions, larger values of μ are required to guarantee a given confidence level than are required for a normal distribution. One should not, however, immediately jump to the conclusion that the present version of the FERD code is returning interval estimates that are too narrow for the assumed confidence level. Using the full-fledged confidence ellipsoid machinery developed in Chapters 4-6 makes it quite easy to prove validity of the scalar confidence intervals $[\phi^{lo},\phi^{up}]$, but this may be a drastic overkill which gives intervals much wider than they need to be. Although we cannot at present give a rigorous mathematical, statistical proof that

the confidence interval obtained by using the normal distribution machinery, which is perfectly valid in the case of unconstrained estimation, is valid when the non-negativity constraint is imposed on the problem, we nevertheless believe that it is. Years of experience with FERD have not revealed any outstanding discrepancies or contradictions and have produced physically consistent solutions to many problems. Accordingly we continue to seek a rigorous proof that the extension to the constrained case is justified and use the method as if it really is.

11. PROGRAM DESCRIPTIONS

11.1 Overall Structure of the Two Codes

The FERDO and FERD program packages are almost identical except for the subroutines that do the mathematical calculations. The hierarchies of subroutine calls for the two packages are given in Figs. 11.1 and 11.2. Clearly most of the subroutines are identical in the two packages. In each case the MAIN program is used only to establish a vector array D large enough to contain all of the variable arrays used by the package. The user may want to recompile this program when the size of his problem changes. In doing so it is necessary to change the initialized value of the integer value LENGTH which must agree with the dimension of the array D. The comment cards at the beginning of the program give the formula for calculating that dimension using upper bounds for the values:

NR = number of rows in the response matrix,

NC = number of columns in the response matrix,

NW = number of window vectors to be used,

NLOC = number of words needed to store the unbinned pulse-height data.

These quantities are all input values described in the next chapter.

Having established the length of the D array, the FERDO/FERD MAIN program transfers to the subroutine SETLO/SETL which handles several other initialization chores, one of which is to set the value of the adjustable parameter

ORNL-DWG 83-13153

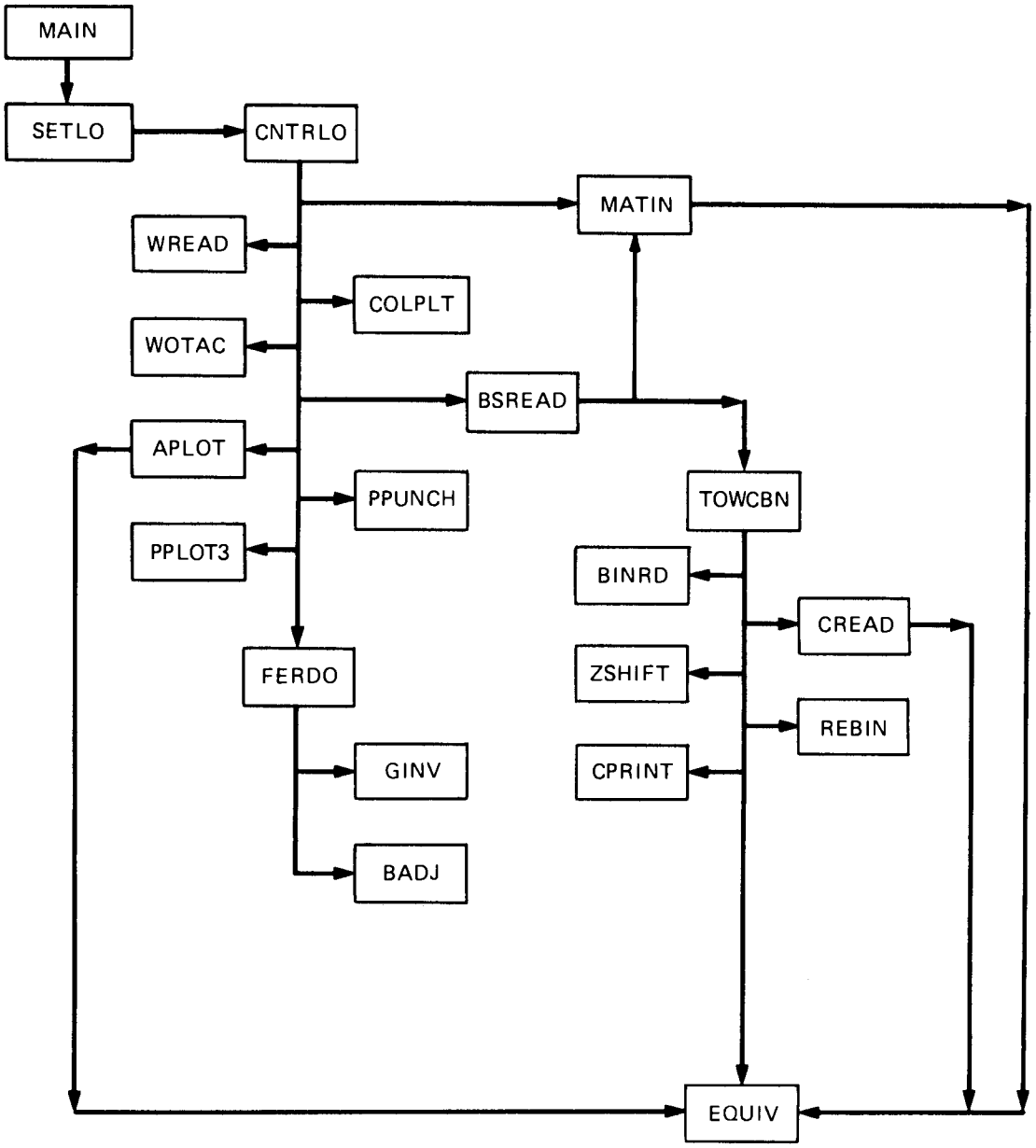


Fig. 11.1. Subroutine Hierarchy for FERDO Package

ORNL-DWG 83-13154

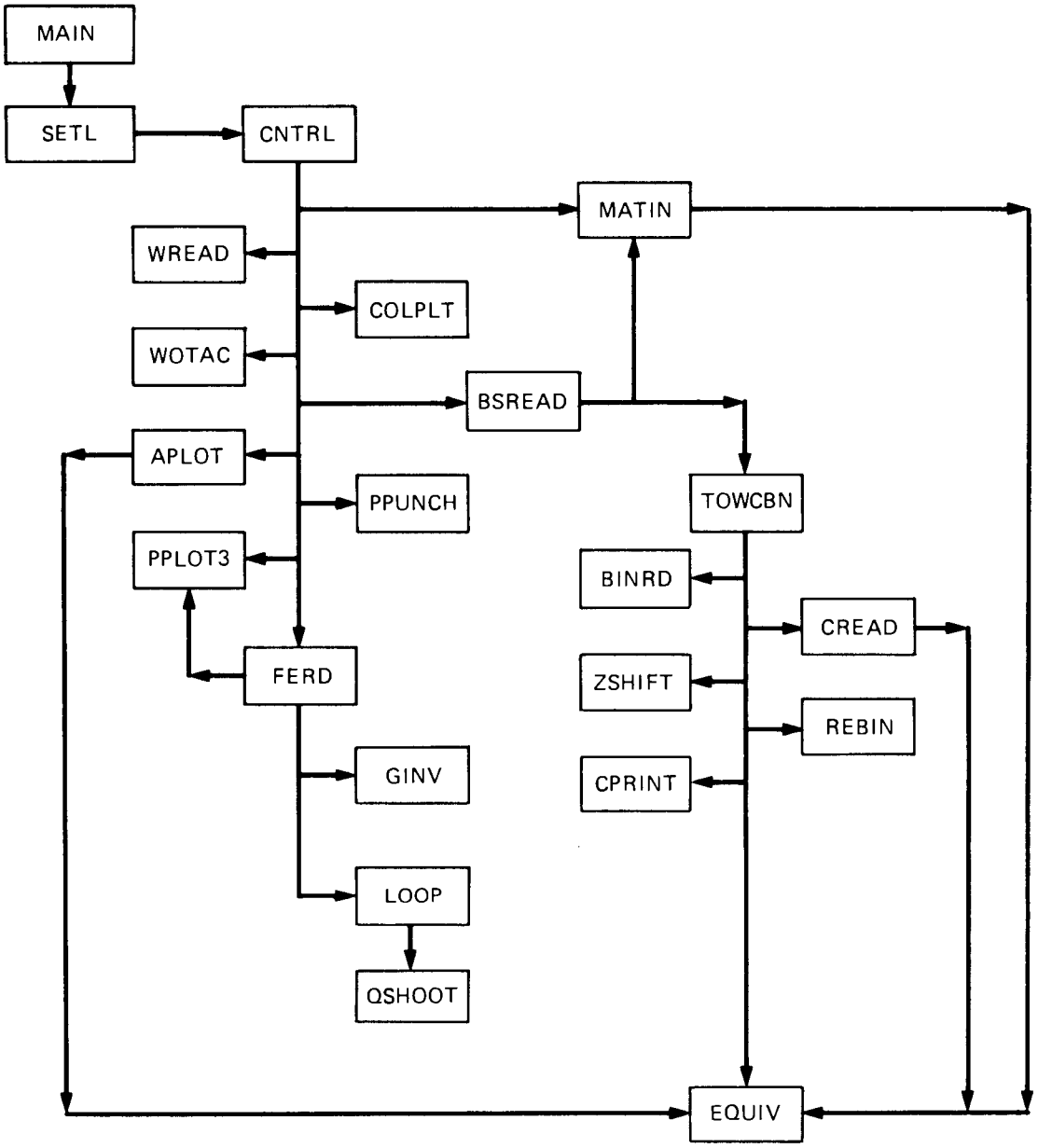


Fig. 11.2. Subroutine Hierarchy for FERD Package

τ defined in Chapter 9 [cf. Eqs. (9-18), (9-19), and the associated discussion]. The program currently sets TAU=1.0, but some users may want to experiment with other values in order to ensure that they obtain the sharpest possible interval estimates. SETLO/SETL also reads the input parameters NR, NC, NW, NLOC, and divides up the array D into subarrays corresponding to the array variables used in the actual calculations. The correspondence between the first location in each D-subarray and its variable array name is established by the call to the subroutine CNTRLO/CNTRL which is, in effect, the central program for the package. Table 11-1 gives a list of the variable names used in the FERDO/FERD packages together with the corresponding mathematical symbols used in this report, the definitions of the variables, and numbers of the equations in this report which define the quantities. The user may find the following additional comments useful in using the table and the program:

1. FERDO assumes no uncertainty in the response matrix A, but FERD allows uncertainties of the type described in Chapter 10 [cf. Eq. (10-2)]. Accordingly, FERDO uses a single response matrix A while FERD uses lower and upper bounding matrices ALO and AUP. The method of defining the bounds is described in the following and in the next chapter. If there is no uncertainty in the matrix, then FERD takes ALO and AUP to be identical.

2. Each column of the matrix/matrices A/ALO, AUP has an associated energy [cf. Eq. (3-6)] which is stored in the array ELAB. These energy values are not used in the calculations but they are used in printing and plotting the response functions and the window vectors.

3. Each row of the matrix/matrices A/ALO, AUP has an associated pulse height bin label stored in the array BLAB. These labels are used only to identify the associated pulse height bins and do not affect the calculations. They are used in plotting the observed counts and standard deviations.

TABLE 11-1

FERDO Variable	FERD Variable	Math. Symbol	Definition	Ref. Eq.
A		$\underline{\underline{A}}$	The response matrix for the detector	(3-6, 3-13)
	ALO	$\underline{\underline{A}}^{lo}$	Lower bound for response matrix	(10-2)
	AUP	$\underline{\underline{A}}^{up}$	Upper bound for response matrix	(10-2)
NR	NR	m	Number of rows in response matrix	
NC	NC	n	Number of columns in response matrix	
B	B	$\underline{b}$	Vector of observed counts (Length = NR)	(2-1, 3-12)
S	S	$\underline{\underline{S}}$	Vector of estimated standard deviations of the observed counts. The diagonal elements of the covariance matrix of the counts. (Length = NR)	(4-6)
NW	NW	p	Number of window vectors used to construct the estimated spectrum or number of energy points in the estimated spectrum.	
WINDOS	WINDOS	W_{kj}	Matrix of window vectors; k = 1, 2, . . ., NW; j = 1, 2, . . ., NC	(3-2)
W	W	$\underline{w}$, $\underline{w}_k$	Current window vector. (Length = NC)	(3-10)
WINW	WINW		Vector (Length = NW), used to generate Gaussian windows, containing the full widths at half maximum.	(11-1)
PLO	PLO	ϕ^{lo},$\phi^{lo}{}_k$	Vector of lower bounds for estimated spectrum (Length = NW)	(3-8, 3-19)

TABLE 11-1 (continued)

FERDO Variable	FERD Variable	Math. Symbol	Definition	Ref. Eq.
PUP	PUP	ϕ^{up}, ϕ_k^{up}	Vector of upper bounds for estimated spectrum (Length = NW)	(3-8, 3-19)
TAW	TAW	τ	Adjustable parameter which determines relative weighting of bias and variance in the estimation process (called TAU in SETLO/SETL).	(9-18, 9-19)
Q	Q	$\underline{\underline{Q}}$	Vector of initial upper bounds for the unknown "true" spectrum. The diagonal elements of matrix Q used in biased estimation process. (Length = NC).	(9-13, 9-14)
UT	UT	$\tilde{\underline{u}}$, $\underline{u}^{(o)}$	Vector of estimation coefficients. The coefficients of combination used to construct the linear estimates of ϕ from the observed counts $\underline{b}$. (Length = NR)	(9-1, 9-19, 10-1)
HT	HT	$(\underline{\underline{HT}})$	Biased inverse matrix (of order NR x NC) used to calculate the coefficients of combination UT from the window vector W.	(9-19, 11-7, 11-11 11-12)
ATEMP	ATEMP		A temporary working storage vector (Length = NC)	
	UTMP		A temporary working storage matrix (order NROW x 10, where NROW is maximum of NR, NC, NW).	
ISTAR	ISTAR	i*	An integer vector (Length = NC) which gives for each of the initial upper bounds Q the row of the matrix A used to obtain that value Q.	(9-13, 10-8)
MU	MU	μ	Chi-square value which guarantees that the upper bounds Q for the unknown solution hold with probability $\geq$ 99.95%.	(4-2, 4-3, 11-5)

TABLE 11-1 (continued)

FERDO Variable	FERD Variable	Math. Symbol	Definition	Ref. Eq.
(SK1)	SK1	δ_1	Maximum fractional relative uncertainty in elements of A (not used in FERDO)	(10-2)
(SK2)	SK2	δ_2	Maximum absolute uncertainty in elements of A (not used in FERDO)	(10-2)
ELAB	ELAB	E_j	Array of energy mesh points used in discretizing the response functions and the window functions. (Length = NC). The energies corresponding to the columns of A/ALO,AUP and WINDOS.	(3-2, 3-6)
PLAB	PLAB		Array of energies associated with the window vectors. (Length = NW). For each window vector W there is an associated energy which describes the location of the window on the energy scale. In the case of a symmetric window (e.g. a Gaussian window), the associated PLAB is usually taken to be the midpoint energy of the window.	
BLAB	BLAB		Array of pulse height energy labels in user defined units. (Length = NR)	
NBITS	NBITS		Number of binary bits in the mantissa of the computer's floating point word.	
SCOFF	SCOFF		Minimum allowed fractional error in the counts B(I)	(11-6)
IOVL	IOVL		First bin in the 4-bin overlap when B vector contains two gain runs.	
FUDGE	FUDGE		"Fudge factor" used to allow for statistical variations in adjusting overlapping gain runs.	
X		$\underline{\tilde{x}}$	Least squares solution vector for the biased estimation problem.	(11-8, 11-9)

TABLE 11-1 (continued)

FERDO Variable	FERD Variable	Math. Symbol	Definition	Ref. Eq.
BADJ	BADJ	($\underline{BADJ}$)	Computed right-hand side for the biased least-squares problem.	(11-10, 11-19)
UTB		$\tilde{\phi}$	Biased estimate of the desired spectral response $\underline{w}^T\underline{x}$	(9-1)
USSUM		$V(\tilde{\phi})$	Variance of the biased estimate $\tilde{\phi} = \underline{\tilde{u}}^T\underline{b}$	(9-5)
WUAQ			Upper bound on the bias for the estimate $\tilde{\phi} = \underline{\tilde{u}}^T\underline{b}$	(9-16)
PHILO		ϕ^{lo}	Lower bound for current spectral estimate w^Tx	(9-17)
PHIUP		ϕ^{up}	Upper bound for current spectral estimate w^Tx	(9-17)
	ULO	$\underline{u}^{(k)}$	Lower biased estimator vector.	(11-13)
	SYNWLO	($\underline{SYNWLO}$)	Synthetic window corresponding to lower biased estimator $\underline{u}^{(k)}$	(11-14)
	ULO	$\underline{u}^{(k)}$	Final converged lower biased estimator	(11-13, 11-16)
	ERR		Statistical part of the error in the lower biased estimate.	(11-16)
	ELO	$\underline{e}^{(k)}$	Vector difference between the given window vector and the synthetic window corresponding to the lower biased estimator.	(11-17)
	QSHOOT QUE QLO		A measure of the total possible overshoot in the lower biased windows.	(11-18)
	DELB	($\underline{DELB}$)	Vector difference between measured and computed work.	(11-20)

4. The count vector B used in the calculations is a single, normalized, calibrated pulse height spectrum binned into BLAB bins. The BLAB values are the pulse heights corresponding to the lower edges of the bins. The program will accept the data in this form, or in the form of raw foreground and background multichannel analyzer data. In the latter case the program will bin the data to produce the proper form of B. These binning procedures are described in the following sections, and descriptions of the two kinds of input are given in the next chapter.

5. If the user chooses to input an already binned pulse height spectrum B, he must also enter the standard deviations S associated with the counts. If he chooses to input the raw multichannel data, the binning programs will calculate properly normalized and calibrated standard deviations assuming square root errors for the raw counts. The program also contains a provision which prohibits the standard deviation estimates from falling below a certain fraction of the corresponding count values. This fraction is currently set to be 0.001, but the user can easily change this by altering the value of the parameter SCOFF defined in the subroutine BSREAD.

6. Each row of the NW × NC window matrix WINDOS corresponds to a separate window vector. Each window has an associated energy PLAB which is the energy level represented by that window (e.g. the median or mean energy of the window). Each PLAB value is also the energy associated with the estimated spectrum bounds PLO, PUP corresponding to that window. The columns of WINDOS are associated with the energies ELAB at which each window function is tabulated. These are the same values at which the response functions are tabulated. For narrow windows (high resolution in the estimated spectrum) the tabulated window values will be zero for most of the energies ELAB. The arrays ELAB and PLAB do not have to be identical and in fact it is recommended that the minimum and maximum PLAB values should be sufficiently inside the total range of the ELAB values so that none of the tabulated windows are incomplete (cut off before they go to zero).

7. The arrays BLAB, ELAB, and PLAB are not used in the calculations, but they are used as abcissas in plotting so they should be consistently ordered.

The subroutines CNTRLO/CNTRL are so nearly identical that they are outlined on the same flow chart in Fig. 11.3. Sections where they differ are indicated by dotted line flows diverging from a circle containing a slash symbol. The CNTRLO version is shown on the left branch and the CNTRL version on the right. Dotted line flows back into a circle containing a slash indicates the beginning of a segment where the two programs are again identical. Boxes with square corners indicate actions that performed unconditionally. Boxes with oval ends represent user-discretionary actions. The diamond shaped box represents a conditional branch and the arrow shaped boxes represent transfers to subroutines that will be discussed in considerable detail in the following sections.

11.2 Data Input and Initialization

At the very outset CNTRLO/CNTRL initializes two very important parameters, NBITS and NSCR. The former is the number of binary bits in the mantissa of the computer's floating point word (22 for IBM 360 single precision) and is used to calculate certain diagnostic indicators described in the following. The latter is the number of a temporary scratch output, input unit used by the program to conserve storage. The window matrix WINDOS is written on this file immediately after it is formed because it is subsequently destroyed in order to use the same space for the arrays A/AUP and HT. The window vectors are then retrieved as needed from this scratch unit. The response matrix A/ALO is also written on this scratch file because it is destroyed by the matrix inversion routine GINV and must be retrieved for subsequent calculations. Users who plan to implement these programs on other computers may want to change the values of NBITS and NSCR.

The next task performed by CNTRLO/CNTRL is to read the input, output option cards. Definitions of the input, output option parameters and detailed

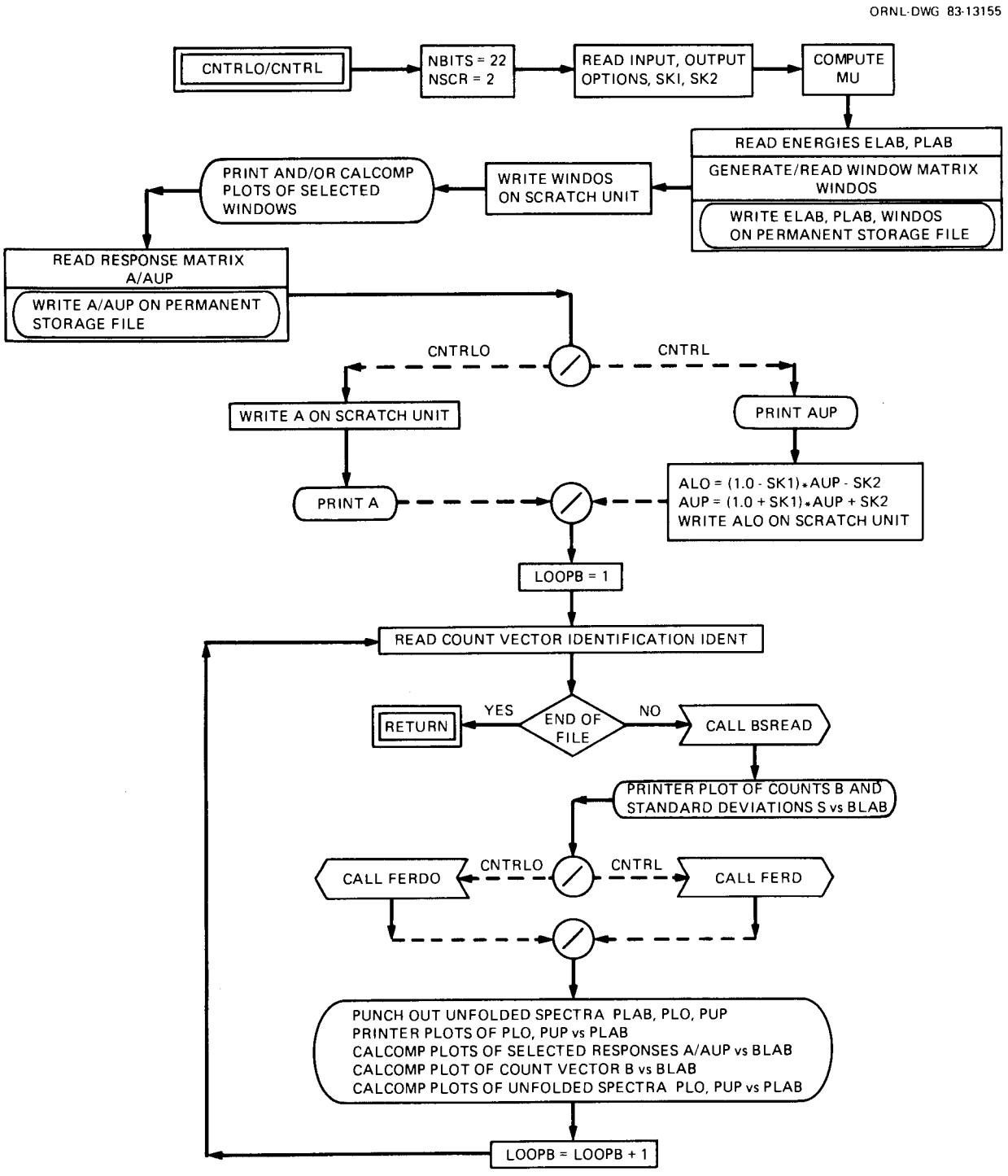


Fig. 11.3. Flow Chart of Subroutines CNTRLO/CNTRL

instructions for writing these cards are given in the next chapter. There are three ways in which the window matrix can be gotten into the program:

1. Calculated internally, in subroutine WREAD, from the card input arrays PLAB, WINW, and ELAB,

2. Read directly from cards, together with the arrays PLAB and ELAB, or

3. Read directly as unformatted input from a permanent storage file (e.g., a disk file), together with the arrays PLAB and ELAB.

Card input is accomplished in subroutine MATIN. If the first option is chosen, there is also an option which allows the user to alter all of the window widths WINW by a constant factor STRECH without changing the WINW values on the cards. The value of STRECH, which allows the user to experiment with the resolution demanded in the estimated spectrum, is read from one of the option cards. Each calculated window is a gaussian centered on the corresponding value of PLAB and normalized to have unit area under the curve. The widths WINW are full widths at half maximum expressed as a percentage of the corresponding mean energy PLAB. Subroutine WREAD converts these widths to energy units using the relation

$$\Delta E_k = 0.01*WINW(K)*PLAB(K),\ k=K=1,2,\ldots,NW \tag{11-1}$$

and to the normal gaussian dispersion parameters σ_k by the standard relation

$$\sigma_k = \frac{\Delta E_k}{2\sqrt{2\ell n(2)}} \quad . \tag{11-2}$$

The tabulated values for window k are then calculated from the standard formula

$$W_k(J) = \frac{1}{\sqrt{2\pi}\,\sigma_k} \exp\left\{-\frac{1}{2\sigma_k^2}[ELAB(J) - PLAB(K)]^2\right\} \quad , \tag{11-3}$$

$$J = 1, 2, \ldots, NC,$$

and these values become the kth row of the matrix WINDOS. The programs contain options for printing the window matrix and for plotting as many as seven selected windows on the CALCOMP plotter. There is also an option allowing the matrix WINDOS to be written on a permanent storage file. Thus, a user might use option 1 or 2 above only once, write the resulting WINDOS on the storage file as unformatted data and then use option 3 to obtain the windows in all future runs.

There are two ways in which the response matrix A/AUP can be gotten into the program:

1. From cards, using subroutine MATIN, or

2. From a permenent storage file as unformatted data.

If the first option is chosen, there is a provision for writing the resulting A on a permanent storage file so that option 2 can be used in subsequent runs. It should be noted that for the FERD package the response matrix AUP which is read in by one of these two options is the "midpoint" matrix $\underline{\underline{A}}^{o}$ of Eq. (10-2). Later in the program it is used to generate the bounding matrices $\underline{\underline{A}}^{lo}$, $\underline{\underline{A}}^{up}$ in arrays ALO, AUP the latter being overwritten by the upper bounds. The parameters δ_1, δ_2 used to compute these bounds are also read from the option cards as SK1, SK2. If both these values are set to zero, there are no uncertainties in $\underline{\underline{A}}$ so FERD uses ALO = AUP = $\underline{\underline{A}}^{o}$. FERDO also reads the parameters SK1, SK2 from the option cards but does not use them in the calculations. This provision was included to allow the use of the same data cards, with no alterations, to run both programs.

After reading in all of the input, output option cards, CNTRLO/CNTRL calculates the value MU = μ used in computing the upper bounds Q = q of Eqs. (9-13, -14). Recall that μ is chosen from a chi-square distribution with NR = m degrees of freedom so that the inequality (9-10) is guaranteed with 99.95% certainty. For values of $m \geqslant 30$, the percentage points for a $\chi^2(m)$ distribution are quite accurately approximated by

$$\chi^2_\alpha = (1/2)[\kappa_\alpha + \sqrt{2m-1}]^2 \qquad (11\text{-}4)$$

where α is the desired confidence level (.9995 in our case) and κ_α is the α-percentage point of the standard normal distribution [cf. Beyer (1968), p. 293]. For $\alpha = .9995$ the tables of the normal distribution give $\kappa_\alpha = 3.270$; so the required value of μ is

$$\mu = \frac{1}{\sqrt{2}}(3.270 + \sqrt{2m-1}) \quad . \qquad (11\text{-}5)$$

It should be noted that this expression is accurate only for $m \geqslant 30$. Users who want to run smaller problems are advised to remove this calculation from CNTRLO/CNTRL and to read in the value of μ (obtained from a χ^2-table) which is appropriate for their problem.

After establishing the value of μ, CNTRLO/CNTRL reads in and writes out the arrays ELAB, PLAB, WINDOS, and A/AUP using the options specified on the input, output option cards. Card input is read in subroutine MATIN using the format described in the next chapters. Printed output is done in WOTAC, a subroutine designed to print 1, 2, or 3 dimensional arrays. Plotting is done in subroutine COLPLT which utilizes the DISSPLA package to produce high quality CALCOMP plots. Users who contemplate changing these packages should be aware that some of the input, output routines require that certain combinations of the arrays be stored contiguously in core. In particular, the arrays PLAB and WINW must be stored together, in that order. This is accomplished in SETLO/SETL by assigning those variables to contiguous subarrays of the master storage array D. Other combinations of arrays which must satisfy this same sort of ordering restriction are {BLAB, B, S} and {PLO, PUP}.

As noted above, both WINDOS and A/ALO are written on a temporary scratch unit (NSCR = 2) for later retrieval. If the user elects to print out the response matrix in CNTRL, he gets the matrix AUP before it is converted into an upper bound matrix, i.e. while it still contains the "midpoint" matrix $\underline{A}^0$. The program then generates ALO, overwrites AUP [cf. Eq. (10-2)], and writes ALO on the scratch unit.

After the response and window matrices are in place CNTRLO/CNTRL is ready to start processing observed count vectors B into estimated spectra PLO, PUP. This is done in a loop so the user can unfold any number of pulse height spectra simply by stacking their input cards, which are described in the next chapter. At the top of the loop the program reads a card containing identification and output title information associated the count vector to follow. If an end of file is encountered on this read, CNTRLO/CNTRL immediately returns control to SETLO/SETL which closes down the DISSPLA plots and immediately returns control to the MAIN program which immediately terminates execution. If identification information is encountered, rather than an end of file, CNTRLO/CNTRL calls subroutine BSREAD to obtain the count vector B, the standard deviations S, and the pulse height energy labels BLAB. If the user chooses to input already binned data, then BSREAD calls subroutine MATIN to read these data from cards. If the user chooses to input raw multichannel analyzer count data, then BSREAD calls subroutine TOWCBN to read and bin the data.

The subroutine TOWCBN and the associated routines BINRD, ZSHIFT, CREAD, CPRINT, and REBIN serve the purpose of generating a single pulse-height spectrum with statistical errors from the "raw" foreground and background data produced by a multichannel analyzer (MCA). To perform this, the routines must, of course, read the channel data and corresponding calibration data, and also read and apply a binning structure which dictates how the channel data is to be grouped into pulse-height bins. In addition, the routines allow for certain flexibilities in the input formats and provide sufficient output edits to document the data manipulations.

The heart of the TOWCBN scheme is the binning table, which specifies how the MCA data are to be grouped to give the same pulse-height boundaries established by the response matrix. Before binning can be performed, the MCA data must first be "conditioned" so that zero pulse-height appears at the lower edge of the first channel (done in subroutine ZSHIFT), and the actual gain is shifted to the nominal gain (done in subroutine REBIN).

Once the MCA data are shifted to the appropriate nominal gain, the data are grouped according to the binning table to yield a measured spectrum with pulse-height boundaries which exactly correspond to the pulse-height boundaries associated with the response matrix. At the same time that the conditioned MCA data are grouped they are also normalized, and the background data are subtracted from the foreground data. Also, the variances of the data are calculated.

The binning table is read by the subroutine BINRD which reads three vectors: KLO, KUP, and KDEL. The vectors are read in the order KLO(1), KUP(1), KDEL(1), KLO(2), KUP(2), KDEL(2),..., KLO(NGRPS), KUP(NGRPS), KDEL(NGRPS). KLO and KUP specify the lower and upper channels of a range of MCA channels (after REBIN has shifted the gain and intercept) which are to be grouped into bins containing KDEL channels per bin. For example:

$$\begin{aligned} KLO(K) &= 1017 \\ KUP(K) &= 1022 \\ KDEL(K) &= 3 \end{aligned}$$

means that between channels 1017 and 1022 the data will be grouped with three channels per bin, or

Bin Number	Channels
I	1017, 1018, 1019
I + 1	1020, 1021, 1022

The binning table, which is specified in ascending pulse-height, must also reflect the appropriate overlap between multiple gain cases. Normally, the nominal gains (and hence the approximate actual gains) of a multiple gain case differ by a factor of 10 so that 10 channels of the first-gain spectrum give the same pulse-height range as one channel of the second-gain

spectrum. For example, if two gains were used and data from the first gain run is specified as 1000 + actual channel number and data from the second gain run is specified as 2000 + actual channel number, then specifying:

K	KLO(K)	KUP(K)	KDEL(K)
N	1151	1160	10
N + 1	2016	2016	1
N + 2	1161	1170	10
N + 3	2017	2017	1

in the binning table would yield a binning structure:

Bin Number	Channels	
I	151, 152,...160	gain 1
I + 1	16	gain 2
I + 2	161, 162,...170	gain 1
I + 3	17	gain 2

Therefore, there would be a 2-bin overlap region. In this example, the binned pulse-height spectrum and the response matrix would have two pairs of identical pulse-height bins. Generally, four overlapping bins are preferred.

The measured foreground and background data are read by CREAD using the FORMAT specified on a DYNMAT card. The form of the DYNMAT card is described in the next chapter.

Upon return from TOWCBN, BSREAD checks the vector S of standard deviations for elements S(I) which are smaller than 0.1% of the corresponding count B(I). All such errors are reset by

$$S(I) = SCOFF * B(I) \quad , \qquad (11\text{-}6)$$

where SCOFF = 0.001 is previously set by the program. The purpose of this procedure is to prevent inconsistent problems which might arise if the errors are underestimated. Some users may want to change the value of SCOFF.

If the counts and standard deviations are read from cards, in subroutine MATIN then, upon return, BSREAD prints a table of the input data, including percentage errors in the B(I), i.e. the quantities 100*S(I)/B(I).

Once the vectors B and S are obtained, either through TOWCBN or MATIN, BSREAD adjusts multiple gain runs for agreement in the overlap region. It calculates normalizations on high gain and low gain so that counts agree in an overlap range of 4 bins beginning with bin number IOVL, a number which is input by the user on card 7 (see next chapter). In making the adjustment BSREAD takes into account the fact that even with the utmost care, it is nearly impossible to make two runs at differing gains and obtain exact agreement between binned counts in the region of overlap. The routine therefore adjusts one or the other of the runs by a "fudge factor," FUDGE, basing its decision upon the relative magnitudes of the differences between values in overlapping bins and the errors associated with the data.

Upon return from BSREAD, CNTRLO/CNTRL is ready to compute the estimated spectrum. This is done by calling subroutine FERDO/FERD.

11.3 Subroutine FERDO

The mathematical arguments underlying the FERDOR subroutine are summarized earlier in this report. Briefly, the Fredholm integral equation problem is represented by a matrix equation with non-negativity constraints. The equation contains a parameter TAW, which may be adjusted by the user. If TAW is set to zero, the method reduces essentially to ordinary least squares regression. However, for poorly conditioned or undetermined systems of equations, the errors "blow up" when TAW is zero. When TAW is set to infinity, on the other hand, the method almost completely ignores

the input data and its associated errors, and forces the results to become all zero. For intermediate values of TAW (values from about 0.1 to 10.0 are appropriate) both the "a priori" non-negativity information and the "a fortiori" observational data are utilized about equally. The result, under these conditions, resembles that obtained by ordinary least squares methods but does not experience "error blow-up" with poorly conditioned or underdetermined systems.

FERDO begins by computing the matrix $\underline{Q}$ defined by Eqs. (9-13, -14). It then uses $\underline{Q}$ and the error vector $\underline{s}$, or more precisely the error matrix $\underline{S}$ = diag $(s_1, s_2, \ldots, s_m)$, to scale the response matrix $\underline{A}$, producing a matrix $(\underline{HT}) = \underline{S}^{-1}\underline{A}\underline{Q}$. It then calls the subroutine GINV to compute a "working inverse" of $(\underline{HT})$. GINV uses a Gramm-Schmidt orthogonalization procedure to replace the matrix $(\underline{HT})$ with the transpose of a scaled working inverse, i.e., $\underline{S}^{-1}\underline{A}(\underline{A}^T\underline{S}^{-2}\underline{A} + \frac{\tau^2}{n}\underline{Q}^{-2})^{-1}\underline{Q}^{-1}$. In the process it uses the space of the original matrix $\underline{A}$ as working storage. Therefore, the first action of FERDO, upon return from GINV, is to restore the matrix $\underline{A}$ by reading it from the scratch unit. then unscales the working inverse matrix using the same scaling transformations applied before the call to GINV, i.e.,

$$\underline{S}^{-1}(\underline{HT})\underline{Q} = \underline{S}^{-2}\underline{A}(\underline{A}^T\underline{S}^{-2}\underline{A} + \frac{\tau^2}{n}\underline{Q}^{-2})^{-1} \quad . \tag{11-7}$$

This last result is the matrix of Eq. (9-19) which is used to convert each window vector $\underline{w}$ into a biased estimation vector $\tilde{\underline{u}}$.

Before calculating the spectral bounds for each window vector, FERDO calls subroutine BADJ to calculate several diagnostic quantities. The first of these is the biased solution vector

$$\tilde{\underline{x}} = (\underline{A}^T\underline{S}^{-2}\underline{A} + \frac{\tau^2}{n}\underline{Q}^{-2})^{-1}\underline{A}^T\underline{S}^{-2}\underline{b} \tag{11-8}$$

whose elements could be combined with those of the window vectors $\underline{w}$ to give a point estimate $\underline{w}^T\tilde{\underline{x}}$ of the corresponding spectral response $\tilde{\phi}$. The vector $\tilde{\underline{x}}$ is the solution of the augmented least squares problem

$$\rho_\tau = \min_{\underline{x}} \left\| \begin{pmatrix} \underline{\underline{A}} \\ \frac{\tau}{\sqrt{n}}\underline{\underline{I}} \end{pmatrix} \underline{x} - \begin{pmatrix} \underline{b} \\ \underline{o} \end{pmatrix} \right\|^2_{S^2} \tag{11-9}$$

where

$$\| f - g \|^2_{\underline{\underline{S}}^2} \equiv (\underline{f} - \underline{g})^T \underline{\underline{S}}^{-2} (\underline{f} - \underline{g}) \quad .$$

The program calls this vector the "dual solution vector" or the "dual vector," and prints its elements along with the corresponding upper bounds q_j which were obtained in FERDO using Eq. (9-13). Subroutine BADJ then computes and prints a vector

$$(\underline{\text{BADJ}}) = \underline{\underline{A}}\, \tilde{\underline{x}} \tag{11-10}$$

which is, in effect a computed "right hand side" of Eq. (3-15). It is the code's estimate of the count vector $\underline{b}$ and can be used to check for consistency between the biased estimator and the observed counts (see below). The program also computes and prints weighted percentage deviations between $(\underline{\text{BADJ}})_i$ and the observed counts b_i, with the weights equal to the inverses of the counting errors s_i, i.e.

$$(\underline{\text{PCTDEV}}) = 100 * \underline{\underline{S}}^{-1}[\underline{b} - (\underline{\text{BADJ}})] \quad .$$

The program then computes and prints out three indicators which are useful in evaluating the quality of the results. If the sum of these indicators is greater than about 5.0, then there may be some problem with the run and the individual indicators should be carefully checked.

The first of these indicators is called DICA1. It is a measure of the conditioning of the response matrix $\underline{\underline{A}}$. It measures the difficulty in inverting the matrix relative to the number of bits of accuracy of the floating point arithmetic of the particular computer being used. Roughly speaking, 6-decimal accuracy in the arithmetic (21 bits) is able to deal with problems in which the NR direction spans about 50 "resolution widths." For more resolution widths, the problem becomes more difficult and greater machine precision is required. Normally, if DICA1 is greater than about 1.0, it suggests that the matrix $\underline{\underline{A}}$ has been set up inappropriately, with the wrong interval between points, or that the problem posed is too ambitious. We are not aware of any practical, well-posed problem tested with FERDO in which the value of DICA1 has been greater than about 0.2.

Indicator No. 2 (DICA2) is a measure of the consistency of the input data in the sense of the ability of the fitting procedures to solve the equations under the constraint of non-negativity. If for some reason the input data contains blunders, such as might be caused by channel overflow in the pulse height analyzer in low channels, the resulting curve will contain a discontinuity, which, under the assumption of non-negativity, is physically impossible. Such as accident will result in an excessively high value of DICA2. In the usual case, with reasonably consistent input data the value of Indicator 2 is less than 1.0.

We have already described the computations of the quantity called "BADJ" ($\underline{b}$ adjusted). This quantity is, in effect, the code's estimate of what the input data should be. Indicator DICA3 is a measure of the weighted residuals between BADJ, the computed value, and $\underline{b}$, the input value. It is comparable to the ordinary "weighted sum of squared residuals" in ordinary least squares regression, except that subroutine GINV appends a $\underline{\underline{Q}}$ matrix based upon non-negativity considerations. The value of DICA3 is distributed statistically somewhat like a Chi-square distribution. If it is greater than about 3.0, it indicates that the input data, $\underline{b}$, and the assigned errors, $\underline{\underline{S}}$, are inconsistent with the kernel of the equation, $\underline{\underline{A}}$. In practical cases, a large value usually signals either a spurious data point or an unrealistically

small error estimate. A common cause is that sometimes, because of instrumental difficulty, a data value will be recorded as zero. FERDOR will tolerate such values if an appropriately large error is assigned, but is intolerant of input such a 0 ± 0, unless such a value is genuinely possible.

After printing the error indicators, BADJ returns control to FERDO which enters the window loop where, for each window vector $\underline{w}$, it successively calculates:

1. The vector UT = $\tilde{\underline{u}}$ according to Eq. (9-19),

2. UTB = $\tilde{\underline{u}}^T \underline{b}$, cf. Eq. (9-1),

3. USSUM = $\tilde{\underline{u}}^T \underline{\underline{S}}^2 \tilde{\underline{u}}$, cf. Eq. (9-5),

4. WUAQ = $| \underline{w} - \underline{\underline{A}} \tilde{\underline{u}} |^T \underline{\underline{Q}} \underline{e}$, cf. Eq. (9-16),

5. PHIUP = UTB + [κ * $\sqrt{\text{USSUM}}$ + WUAQ],
   PHILO = UTB - [κ * $\sqrt{\text{USSUM}}$ + WUAQ],

with the value κ implicitly assumed to be 1 in order to give 68.2% confidence intervals. The last two quantities are, by Eq. (9-17), the required upper and lower spectral bounds corresponding to the window vector $\underline{w}$. The quantities USSUM and WUAQ are recognized as the variance and an upper bound on the bias for the spectral estimate UTB. FERDO prints these two values, under the labels ERR1 and ERR2, along with the estimate and the bounds themselves. ERR1 measures the statistical portion of the final error which can be attributed to the statistical error, $\underline{\underline{S}}$, in the vector of observations, $\underline{b}$. When TAW is set to zero (ordinary least squares) ERR1 is the only source of error which is minimized. ERR2 measures the bias portion of the final error which stems from the fact that the "synthetic" window functions generated by the code by linear combinations of the response functions do not exactly fit the desired windows input by the user. In the limit of TAW = infinity, ERR2 is the only source of error which is minimized. Numerical arithmetic errors are not explicitly accounted for, and

tend to confuse the final interpretation if TAW is very small or very large. Thus, even if TAW = 0, ERR2 will not be exactly zero, and if TAW is very large, ERR1 will not be exactly zero, because of numerical error accumulation.

For each window $\underline{w}$ FERDO also calculates and prints two other quantities SP and RUNSUM. SP is a percent uncertainty in the spectral estimate and is calculated by

$$SP = 100 * \frac{\sqrt{USSUM} + WUAQ}{UTB} \quad .$$

RUNSUM is a running estimate of the integrated spectrum, integrated from the energy corresponding to the first window, i.e. PLAB(1), to the energy corresponding to the current window, i.e. PLAB(K). It is calculated by the formula

$$RUNSUM_K = \sum_{I=1}^{K} UTB_I * [PLAB(I) - PLAB(I-1)] \quad .$$

On each pass through the window loop, FERDO stores the current spectral bounds in the arrays PLO and PUP, i.e.

$$PLO(K) = PHILO_K \quad ,$$
$$PUP(K) = PHIUP_K \quad .$$

When the window loop is completed, FERDO returns the arrays to CNTRLO for further output processing.

11.4 Subroutine FERD

Like FERDO, subroutine FERD begins by computing the diagonal upper bound matrix $\underline{\underline{Q}}$ defined by Eqs. (9-13, -14). The only difference is that FERD takes errors in the matrix $\underline{\underline{A}}$ into account by using elements of the lower bound matrix ALO = $\underline{\underline{A}}^{lo}$ in the denominator of Eq. (9-13). It then computes the "effective matrix errors" by the equation

$$\underline{\delta}(\underline{\underline{A}}) = (\underline{\underline{A}}^{up} - \underline{\underline{A}}^{lo})\ \underline{\underline{Q}}\ \underline{e} \quad ,$$

storing the result temporarily in the vector UT. These errors are used in scaling the response matrix before calling GINV to calculate its working inverse. If we define the diagonal matrix $\underline{\underline{\Delta}}$ by

$$\underline{\underline{\Delta}} = \text{diag}[\delta_1(\underline{\underline{A}}),\ \delta_2(\underline{\underline{A}}),\ \ldots,\ \delta_m(\underline{\underline{A}})] \ ,$$

then the scaling formula is

$$(\underline{\underline{HT}}) = (\underline{\underline{S}} + \underline{\underline{\Delta}})^{-1}\ \underline{\underline{A}}_0\ \underline{\underline{Q}} \qquad (11\text{-}11)$$

where $\underline{\underline{S}}$ is the diagonal matrix of measuring errors and $\underline{\underline{A}}_0$ is the average response matrix,

$$\underline{\underline{A}}_0 = \frac{1}{2}[\underline{\underline{A}}^{lo} + \underline{\underline{A}}^{up}] \ .$$

The scaling in Eq. (11-11) is completely analogous to that applied in FERDO before calling GINV. FERD also calculates a matrix condition indicator for the scaled matrix using the formula

$$\text{CONIND} = \frac{10 \cdot \| (\underline{\underline{S}} + \underline{\underline{\Delta}})^{-1}\ \underline{\underline{A}}_0\ \underline{\underline{Q}} \|}{2^{\text{NBITS}}} = \frac{10 \| (\underline{\underline{HT}}) \|}{2^{\text{NBITS}}} \ ,$$

where NBITS is the number of bits in the fraction of the machine floating point word, and the matrix norm $\| (\underline{\underline{HT}}) \|$ is the square root of the sum of the squares of the elements of $(\underline{\underline{HT}})$. If CONIND is larger than the value of the parameter TAW = τ, which was set in SETL (see above), then TAW is reset to the value of the condition indicator. FERD then calls subroutine GINV to calculate the scaled working inverse of $(\underline{\underline{HT}})$. This is the same subroutine used in the FERDO code, and the result is the same. The matrix $(\underline{\underline{HT}})$ is replaced by the transpose of its working inverse and the matrix $\underline{\underline{A}}^{lo}$ is destroyed in the process. Upon return from GINV, FERD immediately restores $\underline{\underline{A}}^{lo}$ by reading it from the scratch unit and unscales the scaled working inverse using the inverse of the scaling transformation (11-11). Before this inverse transformation the matrix $(\underline{\underline{HT}})$ is given by

$$(\underline{\underline{HT}}) = (\underline{\underline{S}} + \underline{\underline{\Delta}})^{-1}\ \underline{\underline{A}}_o[\underline{\underline{A}}_o{}^T(\underline{\underline{S}} + \underline{\underline{\Delta}})^{-2}\ \underline{\underline{A}}_o + \frac{\tau^2}{n}\ \underline{\underline{Q}}^{-2}]^{-1}\ \underline{\underline{Q}}^{-1} \quad .$$

Afterwards it is

$$(\underline{\underline{HT}}) = (\underline{\underline{S}} + \underline{\underline{\Delta}})^{-2}\ \underline{\underline{A}}_o[\underline{\underline{A}}_o{}^T(\underline{\underline{S}} + \underline{\underline{\Delta}})^{-2}\ \underline{\underline{A}}_o + \frac{\tau^2}{n}\ \underline{\underline{Q}}^{-2}]^{-1} \quad . \tag{11-12}$$

This is the matrix actually used in Eq. (10-1) to compute the initial estimate for each window vector.

After unscaling the working inverse $(\underline{\underline{HT}})$, FERD enters the window loop. On each pass through this loop a new window vector $\underline{w}$ is read from the scratch unit at the top of the loop. The initial estimator $\underline{u}^{(o)}$ is calculated by

$$\underline{u}^{(o)} = (\underline{\underline{HT}})\ \underline{w} \quad ,$$

with the result being stored in the array UT, cf. Eqs. (10-1), (11-12). FERD then calls subroutine LOOP which converts $\underline{u}^{(o)}$ into a vector $\underline{u}^{(k)}$ which gives a lower biased estimator, i.e.

$$\underline{u}^{(k)^T}\ \underline{\underline{A}} \leqslant \underline{w}^T \quad . \tag{11-13}$$

The iteration for accomplishing this result is outlined in Chapter 10, cf. Eqs. (10-6) - (10-8) and the accompanying explanatory text. At each place where a matrix element A_{ij} is required the program chooses either A_{ij}^{lo} or A_{ij}^{up} depending on which will maximize the final value of $\underline{u}^{(k)T}\ \underline{\underline{A}}$. This will assure that the matrix uncertainty is taken into account in the final lower biased estimate. The final estimator is stored in the array ULO.

Once the final estimator is obtained, LOOP calculates the "synthetic window" corresponding to that window. The formula for that window is

$$(\underline{\text{SYNWLO}}) = \underline{u}^{(k)^T}\ \underline{\underline{A}} \quad , \tag{11-14}$$

where the elements A_{ij} are taken to be A_{ij}^{lo} or A_{ij}^{up} as explained above. Each element of this synthetic window should be less than the corresponding element of the window vector $\underline{w}$. LOOP next calculates the lower bound estimate for $\phi = \underline{w}^T \underline{\chi}$ according to the formula

$$\phi^{lo} = \underline{u}^{(k)T} \underline{b} - \sqrt{\underline{u}^{(k)T} \underline{\underline{S}}^2 \underline{u}^{(k)}} \quad , \tag{11-15}$$

ref. Eq. (10-4). This result is stored and returned to FERD in the FORTRAN variable PHILO. LOOP also returns the quantities UT, ULO, SYNWLO, ERR, ELO, and QUE. The first three of these have already been explained. The quantity ERR is given by

$$\text{ERR} = \sqrt{\underline{u}^{(k)T} \underline{\underline{S}}^2 \underline{u}^{(k)}} \tag{11-16}$$

and ELO is the n-vector

$$\underline{e}^{(k)} = \underline{w} - \underline{\underline{A}}^T \underline{u}^{(k)} \quad , \tag{11-17}$$

where the elements A_{ij} are chosen as explained above. Thus, ELO is the vector difference between the given window vector and the synthetic window vector. All of its elements should be positive. The vector $\underline{u}^{(k)}$ is chosen to assure that the synthetic window is smaller than the given window at the energy mesh points E_j = ELAB(J), J = j = 1, 2, ..., n. Both of these window vectors are discrete point representations of underlying continuous window functions. LOOP checks whether the synthetic window function might exceed the desired window function at intermediate energies, between the mesh points. To do this it calls the function subroutine QSHOOT which examines the difference vector ELO. QSHOOT assumes that the energy mesh points are equally spaced and analyzes every set of three adjacent points. For each set it fits a quadratic polynomial through the three corresponding elements of the $\underline{e}^{(k)}$ vector. It then checks the resulting polynomial to see if its minimum value falls within the energy range corresponding to the outer two points. If it does, it checks whether that minimum value is negative. Let $e_{min}(i)$ denote

the minimum value for the segment defined by energy mesh points E_{i-1}, E_i, E_{i+1} and let the energy at which it occurs be $E_{min}(i)$. Then for i = 2, 3, ..., n - 1, QSHOOT checks whether $E_{i-1} \leqslant E_{min}(i) \leqslant E_{i+1}$ and, if so, whether $e_{min}(i) < 0$. These negative excursions are called overshoots. Using all such values $e_{min}(i)$ it calculates the quantity

$$QSHOOT = \frac{4}{9} \sum_{i=2}^{n-1} e_{min}(i) * q_i \quad , \tag{11-18}$$

where each term is included in the sum only if

$$E_{i-1} \leqslant E_{min}(i) \leqslant E_{i+1} \ ,$$

$$e_{min}(i) < 0 \ .$$

The q_i are the diagonal elements of the upper bound matrix $\underline{Q}$ which were computed at the beginning of FERD. LOOP stores the value QSHOOT in the variable QUE and returns to FERD which in turn stores it in a variable called QLO. If QSHOOT exceeds 20% of the confidence interval width then the energy mesh points may need to be taken closer together. FERD also stores PHILO in the element of array PLO corresponding to the current window vector, i.e. in PLO(K) for the Kth window. It also stores some of the other results returned by LOOP in various columns of an array called UTMP whose elements are later used to form various diagnostic quantities for output. We will not here try to explain all of the manipulations of the array UTMP but rather will try to identify the various quantities that are finally written as output.

To compute the upper bound for the current window FERD simply reverses the signs of all of the elements of the window vector $\underline{w}$ and the initial estimator $\underline{u}^{(o)}$ (i.e. UT) and again calls subroutine LOOP. Upon return it reverses the sign of the bound obtained and stores it in the appropriate element of the array PUP of upper bounds. The synthetic window that is returned also has

its sign reversed so that all its elements are then larger than the corresponding elements of w (it becomes an upper biased window). The value returned in QUE is a measure of possible undershoot between mesh points and is stored in QUP.

FERD next tests the output option indicator NOPT (described in the next chapter under the heading Card 5). If the plot option is indicated, FERD calls subroutine PPLOT3 to produce a line printer plot of the current window vector $\underline{w}$ and of the lower and upper synthetic windows on the same graph. At this point the program has reached the bottom of the window loop and goes back to the top to read in a new window from the scratch unit.

After emerging from the window loop FERD again tests the parameter NOPT to see whether to calculate a "computed right hand side" vector ($\underline{BADJ}$), called BADJ in the code, analogous to the one computed in FERDO (see above). This calculation is considerably more complicated than the one in FERDO, however, because of the errors in the matrix $\underline{\underline{A}}$. If the calculation is indicated, FERD enters a loop which calculates each element

$$(\underline{BADJ})_L = BADJ(L),\ L = 1, 2, \ldots, m$$

separately. At the top of the loop it calculates and stores in the vector array UT an initial estimator vector

$$UT = \underline{u}_L^{(o)} = \underline{e}_L^T\, \underline{\underline{A}}^o [\underline{\underline{A}}^{o\ T} (\underline{\underline{S}} + \underline{\underline{\Delta}})^{-2}\, \underline{\underline{A}}^o + \frac{\tau^2}{n}\, \underline{\underline{Q}}^{-2}]^{-1}\, \underline{\underline{A}}^{o\ T} (\underline{\underline{S}} + \underline{\underline{\Delta}})^{-2}$$

where the subscript L on the $\underline{u}_L^{(o)}$ vector means the Lth $\underline{u}^{(o)}$ vector rather than the Lth element of $\underline{u}^{(o)}$. The vector $\underline{e}_L$ is just the unit vector whose Lth component is 1 and whose other components are all zero. Clearly, $\underline{u}_L^{(o)}$ is just the Lth row of the matrix product of the matrix $\underline{\underline{A}}^o$ and its working inverse computed previously by GINV. FERD then calls subroutine LOOP, using in place of the window vector the Lth row of the matrix $\underline{\underline{A}}^{lo}$, i.e. $\underline{e}_L^T\, \underline{\underline{A}}^{lo}$. LOOP begins with the vector $\underline{u}_L^{lo}$ and iterates until it obtains a vector $\underline{u}_L^{lo}$ which satisfies

$$(\underline{u}_L^{lo})^T\, \underline{\underline{A}} \leqslant \underline{e}_L^T\, \underline{\underline{A}}^{lo} \quad .$$

Note that we can write the matrix $\underline{\underline{A}}$ on the left of this inequality because LOOP always chooses between A_{ij}^{lo} and A_{ij}^{up} in such a way that the inequality is guaranteed. Thus, for all $\underline{\underline{A}}$ in the matrix interval $[\underline{\underline{A}}^{lo}, \underline{\underline{A}}^{up}]$, the vector $(\underline{u}_L^{lo})^T\underline{\underline{A}}$ is guaranteed to be less than the Lth row of $\underline{\underline{A}}^{lo}$. Multiplying both sides of the inequality by $\underline{x}$ and substituting on the left from Eq. (3-15) gives

$$(\underline{u}_L^{lo})^T(\underline{b} + \underline{\varepsilon}) \leqslant (\underline{\underline{A}}^{lo}\ \underline{x})_L \quad .$$

The quantity on the left is thus guaranteed to be less than the Lth element of $\underline{\underline{A}}\ \underline{x}$. It involves the statistical error $\underline{\varepsilon}$ but LOOP takes this into account when it returns

$$\text{PHILO} = (\underline{u}_L^{lo})^T\ \underline{b} - \sqrt{(\underline{u}_L^{lo})^T\ \underline{\underline{S}}^2(\underline{u}_L^{lo})}$$

which can be regarded as a lower bound for the Lth component of (BADJ). FERD next obtains an upper bound for this quantity by calling LOOP with the vector $-\underline{e}_L^T\ \underline{\underline{A}}^{up}$ used in place of the window vector. Beginning with $\underline{u}_L^{(o)}$, LOOP iterates to get an upper biased estimator vector $\underline{u}_L^{up}$ and returns (after a sign change) the upper bound

$$\text{PHIUP} = (\underline{u}_L^{up})^T\ \underline{b} + \sqrt{(\underline{u}_L^{up})^T\ \underline{\underline{S}}^2\ \underline{u}_L^{up}} \quad .$$

The desired Lth component of (BADJ) is then computed as the average of PHILO and PHIUP, i.e.

$$\text{BADJ(L)} = (\underline{\text{BADJ}})_L$$

$$= \frac{1}{2}\left[(\underline{u}_L^{lo})^T\ \underline{b} - \sqrt{(\underline{u}_L^{lo})^T\ \underline{\underline{S}}^2\ \underline{u}_L^{lo}} + (\underline{u}_L^{up})^T\ \underline{b} + \sqrt{(\underline{u}_L^{up})^T\ \underline{\underline{S}}^2\ \underline{u}_L^{up}}\right] \quad . \quad (11\text{-}19)$$

FERD does not print out this estimate of $(\underline{\text{BADJ}})$, but instead prints, for each L, the upper and lower bounds PHILO and PHIUP (which might more properly be called BLO and BUP) and the difference

$$(\text{DELB})_L = \underline{b}_L - (\underline{\text{BADJ}})_L \qquad (11\text{-}20)$$

between the measured and computed $\underline{b}$. It also prints a normalized percentage deviation

$$(\mathrm{PCTDEV})_L = 100 * \frac{\underline{b}_L - (\underline{\mathrm{BADJ}})_L}{s_L}$$

where the normalization factor s_L is just the standard deviation of the observed counts.

When the (BADJ) loop has been completed, FERD calls subroutine PPLOT3 to produce a line printer plot of the count vector $\underline{b}$ and the lower and upper bounds (PHILO) and (PHIUP) described in the preceding paragraph, all on the same graph. It then writes out the following norm of the vector (DELB),

$$\left\| \underline{b} - (\underline{\mathrm{BADJ}}) \right\|_{\underline{\underline{S}}^2} = \sqrt{[\underline{b} - (\underline{\mathrm{BADJ}})]^T \, \underline{\underline{S}}^{-2} [\underline{b} - (\underline{\mathrm{BADJ}})]}$$

which it calls the "inconsistency factor" in the output.

The last task performed by FERD before returning to CNTRL is to print a table which gives, for each window, the following quantities:

ENERGY = PLAB,
PLO = ϕ^{lo}, , cf. Eq. (11-15),
PUP = ϕ^{up},
PAVG = $1/2(\phi^{lo} + \phi^{up})$,
SP(%) = $100 * (\phi^{up} - \phi^{lo})/(\phi^{lo} + \phi^{up})$,
a measure of the percentage uncertainty represented by the width of the confidence interval,
ERR = total statistical error in the interval, i.e. sum of the two ERR components, cf. Eq. (11-16).
E PCT = total statistical error expressed as a percentage of the width of the confidence interval, i.e.

$$\mathrm{E\ PCT} = \frac{\mathrm{ERR}}{\phi^{up} - \phi^{lo}} \times 100 \ ,$$

Q PCT = total undershoot and overshoot error expressed as a percentage of the width of the confidence interval, i.e. sum of the two QSHOOT values, cf. Eq. (11-18), divided by $(\phi^{up} - \phi^{lo})$ and multiplied by 100.

11.5 The Final Output

Upon return from FERDO/FERD, CNTRLO/CNTRL first checks the punch option parameter IPPUN (see next chapter under heading Card 7). If indicated, the program will call subroutine PPUNCH to write the window energies PLAB and the corresponding lower and upper bounds PLO and PUP on unit 7. These quantities can then be used later as input to other programs.

Next CNTRLO/CNTRL checks whether the user has requested line printer plots of the calculated bounds. If so, it calls subroutine PPLOT3 to give linear plots of the unfolded spectral estimates PLO and PUP versus the window energy PLAB. It then calls subroutine APLOT to plot the same quantities on a pseudo log scale (which allows negative values).

CNTRLO/CNTRL next does the DISSPLA plots selected by the user (see next chapter). These are all done by subroutine COLPLT. Three different plots are possible: (1) a plot of selected response functions (column of the response matrix $\underline{\underline{A}}$) plotted as histograms versus pulse height, (2) a plot of the count vector $\underline{b}$ and its standard deviations plotted as a histogram versus pulse height, and (3) a plot of the lower and upper bounds for the unfolded spectrum as a function of energy.

When the DISSPLA plots are completed CNTRLO/CNTRL loops back to read in another count vector to be analyzed using the same response matrix and window vectors. If the input file contains the data for a new count vector, it will be analyzed as described above. If the program encounters an end of file while looking for the new count vector, CNTRLO/CNTRL returns to SETLO which calls the DISSPLA subroutine DONEPL to close the plot file. SETLO then returns to the MAIN program which then stops.

12. INPUT REQUIREMENTS FOR FERDO/FERD

Input for the FERDO/FERD codes comprise title cards, numerous options parameters, detector response functions and window functions, and the measured pulse-height data. A detailed description of input cards and data blocks is given below followed by a description of MATIN and CREAD input formats.

Card 1 (20A4)

JDENT(I),I=1,20 - General problem title.

Card 2 (4I10)

NR — Number of rows (pulse-height bins) in detector response matrix.

NC — Number of columns (energies) in response matrix and window matrix.

NW — Number of rows (energies) in window matrix.

NLOC — Number of words needed to store the unbinned pulse-height data - normally NGAIN*NPOINT (# gains x # channels per gain). Use NLOC = 0 if IBIN = 0.

Card 3 (2I10,F10.3,2I10)

IWRD
- -1 - Calculate Gaussian window functions using energies (PLAB) and %-FWHM's (WINW) read from cards - Block 1a.
- 0 - Read row energies (PLAB) and 2-dimensional window matrix from cards - Blocks 1b and 1c.
- N - Read row energies (PLAB), 2-dimensional window matrix, and column energies (ELAB) from device on Logical Unit N.

IWWRT — 0/N - No effect. / Write PLAB, 2-D window matrix, and ELAB onto device on Logical Unit N.

STRECH — Multiplication factor for %-FWHM's (WINW). Not used if IWRD $\geq$ 0.

IWPRT — 0/1 - No effect. / Print 2-D window matrix.

NWPLT — 0/N - No effect. / Plot N different window functions on one graph (N $\leq$ 7). Specific window functions are identified by column on the following input card.

Card 4 (7I10) Input only if NWPLT > 0.

Column numbers of specific window functions to be plotted. NWPLT entires are read.

Card 5 (5I10,2F10.0)

IARD
- 0 - Read 2-dimensional response matrix from cards - Block 3.
- N - Read 2-dimensional response matrix from device on Logical Unit N. N can be same unit as IWRD.

IAWRT
- 0/N - No effect. / Write response matrix onto device on Logical Unit N. N may be same unit as IWWRT.

IAPRT
- 0/1 - No effect. / Print response matrix.

NAPLT
- 0/N - No effect. / Plot N different response functions on one graph (N $\leq$ 7). Specific response functions are identified by column on the following input card.

NOPT5 Output edit option (not used in FERDO):
- 1 - Print only summary information for PLO-PUP solution.
- 2 - Print only summary information for PLO-PUP solution and adjusted B-vector.
- 3 - Print summary information and plot every 20^{th} synthetic window function.
- 4 - Print summary information and plot every synthetic window function.

SK1 Relative (fractional) error factor for response matrix. Not used in FERDO.

SK2 Absolute (normalization) error factor for response matrix. Not used in FERDO.

Card 6 (7I10) Input only if NAPLT > 0.

Column numbers of specific response functions to be plotted. NAPLT entries are read.

Card 7 (7I10)

IBIN 0/1 - Read previously binned data and corresponding statistical errors. / Read actual channel data and calculate binned spectrum and errors.

IDIV 0/1 - No effect. / Divide internally-binned pulse-height data by bin width. (Used only when IBIN=1)

NGAIN Number of overlapping gains in measured data. NGAIN = 1 or 2 only.

IOVL First bin of 4-bin overlap between different gains. Only required if NGAIN=2.

IBPLT 0 - No effect.
1 - Output printer plot of binned spectrum.
2 - Output Calcomp plot of binned spectrum.
3 - Output both printer and Calcomp plots of binned spectrum.

IPPLT Same as IBPLT except for unfolded spectrum upper and lower bounds.

IPPUN 0/1 - No effect. / Punch unfolded spectrum upper and lower bounds.

Block 1a (MATIN format) Input if IWRD = -1.

PLAB(I),I=1,NW - Energies (in units of MeV) corresponding to rows of window matrix. Unfolded spectrum is calculated at these energies.

WINW(I),I=1,NW - FWHM (in per cent) of Gaussian window functions to be calculated by FERDO/FERD.

Block 1b (MATIN format) Input if IWRD = 0.

PLAB(I),I=1,NW - Energies (in units of MeV) corresponding to rows of window matrix. Unfolded spectrum is calculated at these energies.

Block 1c (MATIN format) Input if IWRD = 0.

WINDOS(I,J),I=1,NW;J=1,NC - Two-dimensional window matrix.

Block 2 (MATIN format) Input if IWRD $\leq$ 0.

ELAB(I),I=1,NC - Energies (in units of MeV) corresponding to columns of window matrix and response matrix.

**NOTE: If IWRD > 0, Blocks 1a-c and 2 are not entered, and the corresponding arrays are read (unformatted) from device on Unit IWRD.

Block 3 (MATIN format) Input if IARD = 0.

A(I,J),I=1,NR;J=1,NC - Two-dimensional detector response matrix.

**NOTE: If IARD > 0, Block 3 is not entered, and the response matrix is read (unformatted) from device on Unit IARD.

The following input may be repeated as many times as desired to analyze multiple sets of similar pulse-height data.

Card 8 (20A4)

IDENT(I),I=1,20 - Individual case title.

Block 4a (MATIN format) Input if IBIN = 0.

BLAB(I),I=1,NR - Pulse heights (normally in units of Light Units) corresponding to lower edge of pulse-height bins for which the measured data has been defined. Rows of response matrix must also be defined by BLAB.

B(I),I=1,NR - Measured pulse-height spectrum properly normalized, calibrated, and binned into BLAB bins.

S(I),I=1,NR - Statistical errors associated with B-vector.

**NOTE: If IBIN = 0, no further input is required unless multiple cases are to be analyzed. If IBIN = 1, the following cards and blocks should be input instead of Block 4a.

Card 9 (19F4.3)

FAF(I),I=1,19 - Forte Acceptance Factors: Specifies the relative efficiency of the first 19 bins of the measured pulse-height spectrum.

Card 10 (2I10)

NPOINT Number of channels in one pulse-height spectrum.

NGRPS Number of channel groups in binning table.

Block 4b (12I6 - Repeated as needed)

KLO(I),KUP(I),KDEL(I),I=1,NGRPS - Binning table. KLO and KUP specify the lower and upper channels of a range of MCA channels which are to be grouped into pulse-height bins containing KDEL channels per bin.

Block 5 (7E10.4 - Repeated NGAIN times)

UC(J,1) Foreground normalization (time) for gain J.

UC(J,2) Background normalization (time) for gain J.

ZI(J,1) Foreground zero-intercept channel for gain J.

ZI(J,2) Background zero-intercept channel for gain J.

GN(J,1) Foreground gain (normally in Light Units per channel) for gain J.

GN(J,2) Background gain (normally in Light Units per channel) for gain J.

GNOM(J) Nominal (reference) gain for gain J.

Block 6 (CREAD format)

C(I,J),I=1,NPOINT;J=1,NGAIN - Foreground channel data.

Block 7 (CREAD format)

C(I,J),I=1,NPOINT;J=1,NGAIN - Background channel data.

MATIN Format

Each data block which is input using the MATIN format represents a single call to the subroutine MATIN. The first card in the block must be a FORMAT card of the form:

FORMAT I N (...)

which is read with a (A4,3X,A3,I3,1X,16A4) format. The entries represent:

'FORMAT '	-	designates card as FORMAT card
I	-	'ROW' or 'COL' to signify whether data is in row or column format
N	-	Number of data entries per card
(...)	-	Actual format of data entries.

Each data card must begin with two interger-type indices which indicate the matrix position of the first data entry on the card. The remaining data entries on the card are then stored in succeeding locations within the same row if I = 'ROW' or within the same column if I = 'COL'.

MATIN allows changing formats in midstream through the use of negative indices. If MATIN encounters a card with both indices negative, then no data are read from that card and MATIN expects the next card to be a new FORMAT card. Reading of the input block is terminated when a blank card is encountered.

CREAD Format

Unbinned channel spectra (IBIN=1) are read by subroutine CREAD which uses a style similar to MATIN. Two calls to CREAD are made - one for foreground data and one for background data, if any. The input sequence for a single call to CREAD is as follows:

1. a DYNMAT card (described below)
2. a title card in 20A4 format
3. the data cards in a format specified on the DYNMAT card.

The DYNMAT card has the form:

DYNMAT IREL NWORD (...)

which is read using a (A4,2X,I6,I3,1X,10A4) format. The entries represent:

'DYNMAT'	-	designates the card as a DYNMAT card
IREL	-	offset parameter used to locate multiple gain data within a single 1-dimensional array
NWORD	-	Number of data entries per card
(...)	-	Actual format of data entries.

Each data card which is read using the (...) format must begin with a literal-type title field followed by an integer-type index which gives the channel number of the first data entry on the card. The remaining data entries are stored in succeeding locations.

If NGAIN=2, a second set of DYNMAT, title, and data cards is expected immediately following the first set. Since both sets of channel data are stored in the same 1-D array, the value of IREL for the second gain set, i.e. IREL(2), must exceed the value of IREL(1)+NPOINT. If IREL(2) > IREL(1)+NPOINT, then the value of NLOC (Card 2) must be increased a corresponding amount. The actual value needed for NLOC is IREL(2)+NPOINT -IREL(1).

A CREAD data block is terminated by a blank card. If no background data are available, a single blank card must be input to properly terminate CREAD.

13. SAMPLE PROBLEMS AND OUTPUT DESCRIPTION

This section describes three sample problems where FERDO/FERD have been used to unfold pulse-height data from an NE-213 spectrometer. An abreviated listing of the input is given for each sample problem and a thorough discussion of the output is given for the first problem only. Differences in the output of the other two problems are highlighted. An attempt has been made to demonstrate a variety of input and output options. In the discussion below, several input parameters and arrays are referred to by names which are identified and described in Sec. 12.

13.1 Sample Problem 1

The first sample problem is a FERDO case, although the identical input could also be used with FERD. The pulse-height spectrum to be unfolded resulted from an NE-213 measurement of a modified 14-MeV neutron spectrum degraded by a steel and polyethylene shield. The problem demonstrates a case in which all arrays are input explicitly - a situation which should occur only when a new response matrix or window matrix is tried. A listing of the Job Control Language (JCL) and input data is given in Fig. 13.1 with some of the lengthy arrays abreviated for succinctness.

As seen in the input listing in Fig. 13.1, this problem specifies pre-binned pulse-height data (IBIN=0), so that NLOC could be input as 0 thus saving some memory space. The window functions are input as a two-dimensional array preceded by the PLAB vector and followed by the ELAB vector and the A array (response matrix.). All four of these input blocks are written onto the logical unit 27 device for use in subsequent problems. For the sake of this example, 5 window functions (columns of WINDOS) and 4 response functions (columns of A) were selected for CALCOMP, ie., external plotting. Likewise, the input pulse-height spectrum and the unfolded solution bounds were selected for external plotting as well as for normal line printer plotting.

```
//DTISAMP1 JOB (20106),'SAVE6522,55 DTI 6025'
//*CLASS CPU33=040S,REGION=380K,IO=3.C
/*ROUTE  PRINT LOCAL
/*ROUTE  PUNCH REMOTE46
//SCRT EXEC PGM=IEHPROGM,REGION=64K
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=3330-1,VOL=SER=ZX8888,DISP=OLD
//SYSIN DD *
 SCRATCH DSNAME=X.DTIGTCF1.N685555,VOL=3330-1=ZX8888
//FERDO EXEC PGM=FERDO,REGION=380K
//GO.SYSPLIB DD DSN=X.DTI46103.UNFOLD,DISP=SHR
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=(A,,1001)
//GO.FT07F001 DD SYSOUT=B
//GO.COMPOUT DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=&&PLTOUT,
// SPACE=(4000,(50,50),RLSE)
//GO.FT02F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(TRK,(100,10)),
//   DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6447,BUFNO=1)
//GO.FT27F001 DD UNIT=3330-1,VOL=SER=ZX8888,DISP=(NEW,KEEP),
// DSN=X.DTIGTCF1.N685555,SPACE=(TRK,(10,1),RLSE),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6447,BUFNO=1)
SAMPLE 1:  FERDO NEUTRON PROBLEM WITH FULL ARRAY INPUT
        69         55         55          0
         0         27       1.0          0          5
         5         15         25         35         50
         0         27          0          4          0        0.0        0.0
         5         12         20         40
         0          0          1          0          3          3          0
FORMAT ROW 01 (6X,I4,I2,1P15.8)
         1   0.23470575E+02
         2   0.22560175E+02
         3   0.21579701E+02
         4   0.20555088E+02
         5   0.19498314E+02
         6   0.18520989E+02
         7   0.17532757E+02
         8   0.16539378E+02
         9   0.15628767E+02
        10   0.14791910E+02
                .
                .
                .
        50   0.13095893E+01
        51   0.12047316E+01
        52   0.11068935E+01
        53   0.10066275E+01
        54   0.90693686E+00
        55   0.81135350E+00

FORMAT ROW 07 (1X,2I3,3X,7E10.6)   ( 55, 55)  AS PUNCHED
  1  1  1 239831+00 206500+00 125770+00 519224-01 139907-01 286424-02 408266-03
  2  1  2 211904+00 249991+00 206512+00 114245+00 401928-01 103654-01 180329-02
  3  1  3 121841+00 211869+00 259673+00 209311+00 103849+00 357253-01 804276-02
  4  1  4 380777-01 107414+00 213499+00 271812+00 209415+00 103458+00 322710-01
  5  1  5 502063-02 258916-01 942134-01 214178+00 285693+00 223330+00 105613+00
  6  1  6 295859-03 298304-02 212931-01 925515-01 228608+00 299854+00 227342+00
  7  1  7 503968-05 114694-03 185909-02 178268-01 936723-01 232264+00 315663+00
  8  1  8 0      +  0      +  472243-04 118655-02 156775-01 848743-01 236732+00
  9  1  9 0      +  0      +  0      +  235148-04 105917-02 138016-01 866989-01
 10  1 10 0      +  0      +  0      +  0      +  270660-04 942486-03 147325-01
 11  1 11 0      +  0      +  0      +  0      +  0      +  265128-04 113432-02
 12  1 12 0      +  0      +  0      +  0      +  0      +  0      +  201897-04
  1  8  1 408352-04 359956-05 0      +  0      +  0      +  0      +  0      +
  2  8  2 214152-03 215733-04 0      +  0      +  0      +  0      +  0      +
  3  8  3 119532-02 143683-03 148889-04 0      +  0      +  0      +  0      +
  4  8  4 638640-02 964442-03 119747-03 132676-04 0      +  0      +  0      +
  5  8  5 302129-01 613835-02 967054-03 129323-03 108635-04 0      +  0      +
                                .
                                .
                                .
```

Fig. 13.1. Input listing for sample problem 1.

```
 43 50 43 189884-02 268884-03 348152-04 0     +  0     +  0     +  0     +
 44 50 44 161310-01 276698-02 416727-03 471956-04 0     +  0     +  0     +
 45 50 45 672028-01 137511-01 238504-02 304771-03 295817-04 0     +  0     +
 46 50 46 206095+00 507265-01 102150-01 148562-02 160367-03 0     +  0     +
 47 50 47 532902+00 162346+00 388966-01 658650-02 837132-03 806993-04 0     +
 48 50 48 116365+01 458065+00 135428+00 276122-01 395782-02 447202-03 0     +
 49 50 49 202434+01 108598+01 414526+00 106095+00 184618-01 243240-02 0     +
 50 50 50 259882+01 205676+01 108403+01 370491+00 826933-01 133034-01 0     +
 51 50 51 211485+01 274635+01 218724+01 108482+01 334531+00 699528-01 0     +
 52 50 52 975485+00 225078+01 290186+01 222734+01 100613+01 287585+00 0     +
 53 50 53 198631+00 956024+00 228729+01 308352+01 228592+01 983385+00 0     +
 54 50 54 129380-01 159222+00 841224+00 234098+01 329426+01 241390+01 0     +
 55 50 55 209462-03 820056-02 115393+00 789410+00 247013+01 353005+01 0     +

FORMAT ROW 01 (6X,I4,I2,1E15.8)
          1   0.23470575E+02
          2   0.22560175E+02
          3   0.21579701E+02
          4   0.20555988E+02
          5   0.19498314E+02
          6   0.18520989E+02
          7   0.17532757E+02
          8   0.16539378E+02
          9   0.15628767E+02
         10   0.14791910E+02
         11   0.14020996E+02
         12   0.13200995E+02
         13   0.12392783E+02
         14   0.11751374E+02
         15   0.11247053E+02
                  .
                  .
                  .
         48   0.15106974E+01
         49   0.14121000E+01
         50   0.13095893E+01
         51   0.12047316E+01
         52   0.11068935E+01
         53   0.10066275E+01
         54   0.90693686E+00
         55   0.81135350E+00

FORMAT COL 08 (2X,2I3,8E9.3)
    1   1 .181E-01 .211E-01 .163E-01 .160E-01 .158E-01 .149E-01 .303E-01 .278E-01
    9   1 .269E-01 .231E-01 .214E-01 .205E-01 .217E-01 .190E-01 .302E-01 .331E-01
   17   1 .315E-01 .302E-01 .306E-01 .271E-01 .281E-01 .306E-01 .294E-01 .309E-01
   25   1 .582E-01 .612E-01 .618E-01 .582E-01 .525E-01 .499E-01 .430E-01 .412E-01
   33   1 .347E-01 .307E-01 .481E-01 .440E-01 .406E-01 .419E-01 .420E-01 .356E-01
   41   1 .306E-01 .312E-01 .291E-01 .281E-01 .409E-01 .387E-01 .357E-01 .344E-01
   49   1 .315E-01 .283E-01 .253E-01 .272E-01 .259E-01 .245E-01 .458E-01 .466E-01
   57   1 .438E-01 .441E-01 .457E-01 .439E-01 .430E-01 .462E-01 .470E-01 .424E-01
   65   1 .292E-01 .151E-01 .681E-02 .173E-02
    1   2 .204E-01 .166E-01 .173E-01 .171E-01 .175E-01 .147E-01 .270E-01 .292E-01
    9   2 .239E-01 .219E-01 .242E-01 .246E-01 .234E-01 .205E-01 .379E-01 .380E-01
   17   2 .382E-01 .379E-01 .345E-01 .368E-01 .336E-01 .372E-01 .378E-01 .414E-01
   25   2 .735E-01 .729E-01 .671E-01 .614E-01 .547E-01 .454E-01 .372E-01 .346E-01
   33   2 .322E-01 .298E-01 .478E-01 .431E-01 .454E-01 .431E-01 .375E-01 .340E-01
   41   2 .310E-01 .272E-01 .249E-01 .239E-01 .396E-01 .377E-01 .353E-01 .363E-01
   49   2 .300E-01 .312E-01 .337E-01 .276E-01 .319E-01 .266E-01 .484E-01 .514E-01
   57   2 .506E-01 .514E-01 .515E-01 .496E-01 .540E-01 .523E-01 .437E-01 .322E-01
   65   2 .142E-01 .739E-02 .151E-02 .959E-03
    1   3 .221E-01 .205E-01 .199E-01 .167E-01 .174E-01 .155E-01 .302E-01 .284E-01
    9   3 .276E-01 .253E-01 .259E-01 .292E-01 .256E-01 .249E-01 .414E-01 .438E-01
                                  .
                                  .
                                  .
```

Fig. 13.1. -continued-

```
    17 51 .316E-04 .483E-05 .148E-04 .213E-04 .209E-04 .904E-05 .148E-05 .719E-05
    25 51 .709E-05 .394E-05 .000E+00
     1 52 .821E+00 .775E+00 .676E+00 .575E+00 .476E+00 .338E+00 .261E+00 .335E-01
     9 52 .565E-02 .280E-02 .151E-02 .681E-03 .227E-03 .700E-04 .377E-04 .203E-04
    17 52 .262E-04 .849E-05 .000E+00
     1 53 .915E+00 .771E+00 .624E+00 .454E+00 .254E+00 .100E+00 .380E-01 .507E-02
     9 53 .222E-02 .820E-03 .312E-03 .877E-04 .222E-04 .140E-04 .317E-04 .155E-04
    17 53 .640E-05 .551E-05 .145E-04 .145E-04 .552E-05 .483E-05 .619E-05 .281E-05
    25 53 .449E-05 .000E+00
     1 54 .910E+00 .676E+00 .402E+00 .157E+00 .412E-01 .878E-02 .444E-02 .163E-02
     9 54 .541E-03 .138E-03 .538E-04 .298E-04 .495E-05 .497E-05 .000E+00
     1 55 .676E+00 .302E+00 .783E-01 .139E-01 .296E-02 .139E-02 .113E-02 .392E-03
     9 55 .855E-04 .242E-04 .191E-04 .176E-04 .649E-05 .123E-04 .579E-05 .000E+00

PREBINNED PULSE-HEIGHT DATA FOR SAMPLE 1
FORMAT ROW 03 (6X,I4,I2,3E15.8)
          1   0.10750000E+00 0.58200051E-07 0.10095450E-09
          2   0.12250000E+00 0.46992510E-07 0.90714840E-10
          3   0.13750000E+00 0.39095937E-07 0.82742728E-10
          4   0.15250000E+00 0.33146102E-07 0.76186890E-10
          5   0.16750000E+00 0.28439675E-07 0.70571010E-10
          6   0.18250000E+00 0.24637228E-07 0.65684052E-10
          7   0.20500000E+00 0.40221739E-07 0.83925597E-10
          8   0.23500000E+00 0.32011980E-07 0.74872147E-10
          9   0.26500000E+00 0.25825781E-07 0.67249761E-10
         10   0.29500000E+00 0.21539085E-07 0.61415452E-10
         11   0.32500000E+00 0.18094870E-07 0.56291361E-10
         12   0.35500000E+00 0.15123394E-07 0.51462235E-10
         13   0.38500000E+00 0.12935422E-07 0.47594232E-10
         14   0.41500000E+00 0.11200632E-07 0.44287920E-10
         15   0.45500000E+00 0.15734942E-07 0.52492419E-10
                             .
                             .
                             .
         50   0.52350000E+01 0.17839630E-08 0.17674899E-10
         51   0.55450000E+01 0.17262967E-08 0.17386883E-10
         52   0.58550000E+01 0.16002113E-08 0.16739893E-10
         53   0.61650000E+01 0.14562958E-08 0.15969408E-10
         54   0.64750000E+01 0.12918850E-08 0.15040974E-10
         55   0.69100000E+01 0.16950716E-08 0.17228919E-10
         56   0.74700000E+01 0.63469839E-09 0.10542600E-10
         57   0.80300000E+01 0.11417395E-09 0.44714413E-11
         58   0.85900000E+01 0.00000000E+00 0.44714413E-11
         59   0.91500000E+01 0.00000000E+00 0.44714413E-11
         60   0.97100000E+01 0.00000000E+00 0.44714413E-11
         61   0.10270000E+02 0.00000000E+00 0.44714413E-11
         62   0.10830000E+02 0.00000000E+00 0.44714413E-11
         63   0.11390000E+02 0.00000000E+00 0.44714413E-11
         64   0.11950000E+02 0.00000000E+00 0.44714413E-11
         65   0.12510000E+02 0.00000000E+00 0.44714413E-11
         66   0.13070000E+02 0.00000000E+00 0.44714413E-11
         67   0.13630000E+02 0.00000000E+00 0.44714413E-11
         68   0.14190000E+02 0.00000000E+00 0.44714413E-11

//GLD EXEC GLDPLT,REGION=110K
//COMPIN DD DSN=&&PLTOUT,DISP=(OLD,DELETE)
//FT05F001 DD *
DRAW=1-END$
//
ENDINPUT
```

Fig. 13.1. -continued-

Considerable printed output is generated by FERDO (and even more by FERD) so that the output from the sample problems will not be listed but rather merely described. The first output gives an edit of input parameters and a statement of required computer memory. A DISSPLA-generated summary of the window function plot is given next, followed by an edit of the input pulse-height spectrum. This edit includes five columns representing the pulse-height bin numbers, the pulse-height values associated with the lower edge of the bins (PLAB vector), the binned count-rate (B vector), the corresponding errors (S vector), and the errors calculated as percent of B. This edit is completed with a single statement which identifies the values of SCOFF, FUDGE, annd RELFU as determined in subroutine BSREAD. If IBIN=1, BSREAD uses the value of SCOFF (presently set equal to 0.001) to calculate a minimum permissible error for any B_i which is given by: S_i = SCOFF * B_i. When NGAIN=2, BSREAD must account for the inevitable disagreement which occurs in the overlap region of the two gain runs. It does so by first calculating the ratio of the total binned counts in the overlap region of the high gain run relative to the low gain run (FUDGE parameter), and then adjusts one gain or the other by an appropriate factor (RELFU) so as to force agreement in the overlap region. The determination of which gain run gets scaled is based on their relative errors in the overlap region. Following the tabular edit of B and S is a printer plot of B and S generated by APLOT (pseudo-log plot with 2 negative cycles and 3 positive cycles).

Next a tabular edit of the dual vector (X vector) and the corresponding uppper-bounds (Q vector) is given. The X vector is a sort of pseudo solution in that it gives the variation of the PLO and PUP solution bounds as the window functions are varied. Typically the X vector is highly oscillatory about zero. Following the edit of the X and Q vectors, the calculated B-ADJ vector is given which represents the product, $\underline{\underline{A}} * \underline{x}$, i.e., it is the computed pulse-height spectrum which is consistent with the unfolded energy spectrum. Included with the edit of B-ADJ is a tabular listing of the original B and S vectors and the computed percent difference between B and B-ADJ, weighted by S.

Three indicators are printed next which are intended to provide guidance in judging the "quality" of the unfolded solution. These indicators have been passed down through the generations of FERDO/FERD precursors, as has the descriptive text given below:

> Indicator No. 1. A measure of the conditioning of the A matrix (response matrix). It measures the difficulty in inverting the matrix relative to the number of bits of accuracy of the floating point arithmetic of the particular computer being used. Roughly speaking, 6-decimal accuracy in the arithmetic (21 bits) is able to deal with problems in which the NR direction spans about 50 "resolution widths." Larger NR requires correspondingly higher accuracy. Normally, if Indicator No. 1 is greater than about 1.0, it suggests that the matrix A has been set up inappropriately, with the wrong interval between points, or that the problem posed is too ambitious. In most well-posed problems, Indicator No. 1 will be much less than 1.0.
>
> Indicator No. 2. A measure of the consistency of the input data with the constraint of non-negativity. If the input data should contain some uncorrected blunder such as channel overflows or "dropped" channels, a discontinuity in the spectrum may exist which is physically impossible under the assumption of non-negativity. Such an input will result in an excessively high value of Indicator No. 2. Normally, the value of Indicator No. 2 should be less than 1.0.
>
> Indicator No. 3. A measure of the consistency of the computed B-ADJ vector and the input B vector. The value of Indicator No. 3 is distributed statistically somewhat like a Chi-square distribution, and is similar to the "weighted sum of squared residuals" in ordinary least squares regression. If it is greater than about 3.0, it indicates that the input data and the assigned errors (B and S) are inconsistent with the response matrix, A. In practical cases, a large value usually suggests either a spurious data point or an unrealistically small error estimate. A common cause is that sometimes a data value will be input as zero, which can be tolerated by FERDO if an appropriately large error is assigned. Experience has shown that for the upper end of the input spectrum, where the count data often converge to zero, the error assigned to the zero-count channels (or bins) should be set equal to the error assigned to the last non-zero channel. This seems to give a stable solution at the upper energy end of the spectrum without also excessively wide solution bounds.

The final output relates directly to the unfolded solution bounds. First, a table is printed which includes: (1) the energies associated with the unfolded spectrum (PLAB), (2) the lower and upper bounds (PLO

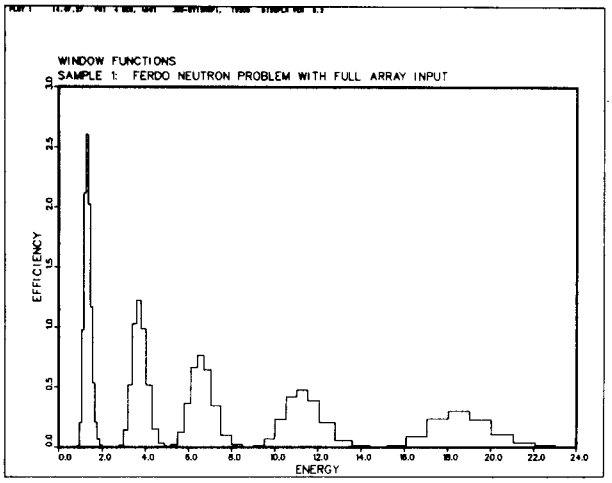
and PUP) of the unfolded spectrum, (3) the computed average of PLO and PUP (labeled PAVE), (4) a pseudo standard deviation for PAVE given in per cent, (5) a running integral of PAVE (labeled RUNSUM), and (6) two columns labeled ERR1 and ERR2. ERR1 represents the statistical portion of the final error which can be attributed to the statistical errors in the original pulse-height data. ERR2 on the other hand represents the "bias" portion of the final error, and results from the fact that the implicit window functions generated within FERDO to minimize total errors do not exactly match the original window functions input by the user. These two error terms are related to TAW, since if TAW=0, then ERR1 is the only source of error (an ordinary least squares problem), and if TAW goes to infinity, then ERR2 will become the only source of error.

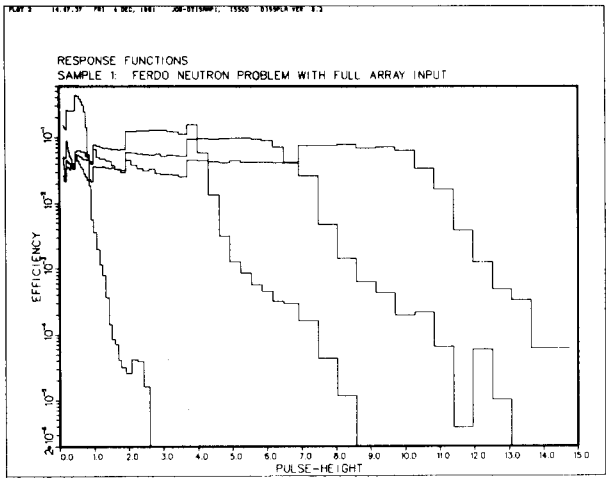
The tabular edit of PLO and PUP is followed by two line printer plots of PLO and PUP (a linear plot and a pseudo-log plot) if IPPLT equals 1 or 3. At the same time, external plots are made of: selected response functions (if NWPLT is greater than zero), B and S (if IBPLT equals 2 or 3), and PLO and PUP(if IPPLT equals 2 or 3). DISSPLA-generated summaries are printed for each plot. The external plots generated from sample problem 1 are shown in Fig. 13.2. The plots were produced on a GOULD electrostatic plotter using the JCL procedure listed at the end of Fig. 13.1

13.2 Sample Problem 2

The second sample problem is also a FERDO case and is identical to the first problem except that the PLAB, WINDOS, ELAB, and A arrays are read from an external device rather than from cards. Also, no external plots are selected, but a punched output of the PLO-PUP solution is produced.

A listing of the JCL and input data is given in Fig. 13.3. The same data set saved in sample problem 1 is used in this case to provide the bulky window and response matrices, as well as the PLAB and ELAB





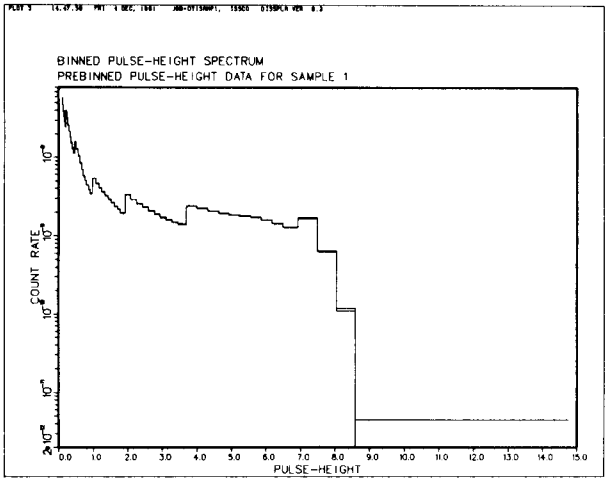
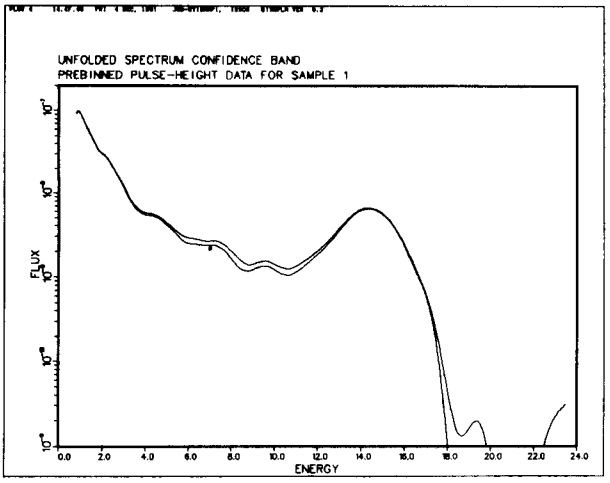
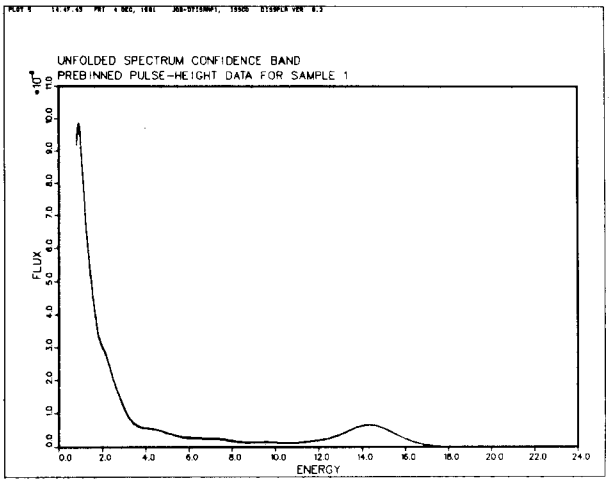


Fig. 13.2. Plot output from sample problem 1.





```
//DIISAMP2 JOB (20106),'SAVE6522,55 DTI 6025'
//*CLASS CPU33=040S,REGION=380K,IO=3.0
/*ROUTE  PRINT LOCAL
/*ROUTE  PUNCH REMOTE45
//FERDO EXEC PGM=FERDO,REGION=380K
//GO.STEPLIB DD DSN=X.DTI46103.UNFOLD,DISP=SHR
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=(A,,1001)
//GO.FT07F001 DD SYSOUT=B
//GO.COMPOUT DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=&&PLTOUT,
// SPACE=(4000,(50,50),RLSE)
//GO.FT02F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(TRK,(100,10)),
//  DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6447,BUFNO=1)
//GO.FT27F001 DD UNIT=3330-1,VOL=SER=ZX8888,DISP=SHR,DCB=BUFNO=1,
// DSN=X.DTIGTCF1.N685555
SAMPLE 2:  FERDO NEUTRON PROBLEM WITH MINIMUM INPUT
        68         55         55          0
        27          0        1.0          0          0
        27          0          0          0          0        0.0        0.0
         0          0          1          0          1          1          1
PREBINNED PULSE-HEIGHT DATA FOR SAMPLE 1
FORMAT ROW 03 (6X,I4,I2,3E15.8)
         1   0.10750000E+00 0.58200051E-07 0.10095450E-09
         2   0.12250000E+00 0.46992510E-07 0.90714840E-10
         3   0.13750000E+00 0.39195937E-07 0.82742728E-10
         4   0.15250000E+00 0.33146102E-07 0.76186890E-10
         5   0.16750000E+00 0.28439675E-07 0.70571010E-10
         6   0.18250000E+00 0.24637228E-07 0.65684052E-10
         7   0.20500000E+00 0.40221739E-07 0.83925597E-10
         8   0.23500000E+00 0.32011980E-07 0.74872147E-10
         9   0.26500000E+00 0.25825781E-07 0.67249761E-10
        10   0.29500000E+00 0.21539085E-07 0.61415452E-10
        11   0.32500000E+00 0.18094870E-07 0.56291361E-10
        12   0.35500000E+00 0.15123394E-07 0.51462235E-10
        13   0.38500000E+00 0.12935422E-07 0.47594232E-10
        14   0.41500000E+00 0.11200632E-07 0.44287920E-10
        15   0.45500000E+00 0.15734942E-07 0.52492419E-10
                            .
                            .
                            .
        50   0.52350000E+01 0.17839630E-08 0.17674899E-10
        51   0.55450000E+01 0.17262967E-08 0.17386883E-10
        52   0.58550000E+01 0.16002113E-08 0.16739893E-10
        53   0.61650000E+01 0.14562958E-08 0.15969408E-10
        54   0.64750000E+01 0.12918850E-08 0.15040974E-10
        55   0.69100000E+01 0.16950716E-08 0.17228919E-10
        56   0.74700000E+01 0.63469839E-09 0.10542600E-10
        57   0.80300000E+01 0.11417395E-09 0.44714413E-11
        58   0.85900000E+01 0.00000000E+00 0.44714413E-11
        59   0.91500000E+01 0.00000000E+00 0.44714413E-11
        60   0.97100000E+01 0.00000000E+00 0.44714413E-11
        61   0.10270000E+02 0.00000000E+00 0.44714413E-11
        62   0.10830000E+02 0.00000000E+00 0.44714413E-11
        63   0.11390000E+02 0.00000000E+00 0.44714413E-11
        64   0.11950000E+02 0.00000000E+00 0.44714413E-11
        65   0.12510000E+02 0.00000000E+00 0.44714413E-11
        66   0.13070000E+02 0.00000000E+00 0.44714413E-11
        67   0.13630000E+02 0.00000000E+00 0.44714413E-11
        68   0.14190000E+02 0.00000000E+00 0.44714413E-11

//
ENDINPUT
```

Fig. 13.3. Input listing for sample problem 2.

vectors. This should be the normal input procedure when the code is being used in a "production" manner. Even though the arrays are input from an external device, all the print and plot options allowed by IWPRT, NWPLT, IAPRT, and NAPLT are still available. The pulse-height spectrum listed in Fig. 13.3 is the same as used for sample problem 1, and the printed output is the same as described in the previous section. Figure 13.4 lists the punched output produced by this case. The format of the punched cards can be easily changed by modifying subroutine PPUNCH.

13.3 Sample Problem 3

The third and final sample problem is a FERD case. The pulse-height spectrum, which is input as raw channel data, resulted from an NE-213 measurement of the gamma-ray flux generated from neutron transmission through a fast reactor blanket and shield mockup. As with sample problem 2, the PLAB, WINDOS, ELAR, and A arrays are read from an external device.

An abbreviated listing of the input is given in Fig. 13.5. Since binning of the input channel data is required (IBIN=1), NLOC is input sufficiently large to accommodate the single gain data (actually, NLOC has been input excessively large, since 1024 would have been sufficient). Five response functions were selected for external plotting, as well as the binned pulse-height spectrum and the unfolded solution bounds. IDIV was input as zero which causes the binned pulse-height spectrum not to be divided by the bin width. This option requires consistency with the response matrix being used, which for this case, also does not include the division by bin width. The pulse-height data input occurs in several data blocks as described in Sec. 12.

As with FERDO, FERD printed output starts with an edit of the input parameters. This edit is followed by a summary of the binning table which dictates how the individual channels of the input spectrum are to be grouped into larger pulse-height bins. An edit of the foreground and background normalizations and gains is also given. Next an edit of the

```
PREBINNED PULSE-HEIGHT DATA FOR SAMPLE 1
        55
 0.23471E 02-0.14962E-10 0.30660E-10 0.22560E 02-0.44489E-10 0.11950E-10
 0.21580E 02-0.90138E-10-0.13138E-10 0.20556E 02-0.76570E-10-0.61435E-11
 0.19498E 02-0.29531E-10 0.18432E-10 0.18521E 02-0.34345E-10 0.13805E-10
 0.17533E 02 0.12699E-09 0.19391E-09 0.16539E 02 0.11340E-08 0.11998E-08
 0.15629E 02 0.34727E-08 0.35726E-08 0.14792E 02 0.59940E-08 0.61281E-08
 0.14021E 02 0.62193E-08 0.63887E-08 0.13201E 02 0.40975E-08 0.42827E-08
 0.12393E 02 0.23117E-08 0.25210E-08 0.11751E 02 0.16321E-08 0.18172E-08
 0.11247E 02 0.12709E-08 0.14703E-08 0.10777E 02 0.10729E-08 0.12674E-08
 0.10259E 02 0.11078E-08 0.13044E-08 0.97411E 01 0.13150E-08 0.15141E-08
 0.92613E 01 0.13097E-08 0.14936E-08 0.87578E 01 0.11769E-08 0.13929E-08
 0.82364E 01 0.13589E-08 0.17312E-08 0.77365E 01 0.19279E-08 0.22823E-08
 0.72352E 01 0.23421E-08 0.26585E-08 0.68390E 01 0.23561E-08 0.26304E-08
 0.65557E 01 0.23830E-08 0.27211E-08 0.62542E 01 0.24247E-08 0.28405E-08
 0.59380E 01 0.24904E-08 0.29203E-08 0.56453E 01 0.27967E-08 0.31668E-08
 0.53459E 01 0.33630E-08 0.36141E-08 0.50422E 01 0.39660E-08 0.42138E-08
 0.47521E 01 0.45941E-08 0.48571E-08 0.44540E 01 0.51476E-08 0.53965E-08
 0.41528E 01 0.53530E-08 0.56383E-08 0.39061E 01 0.55513E-08 0.58647E-08
 0.37074E 01 0.59647E-08 0.63104E-08 0.35009E 01 0.66478E-08 0.71046E-08
 0.32965E 01 0.79852E-08 0.83230E-08 0.30975E 01 0.10089E-07 0.10632E-07
 0.28984E 01 0.13136E-07 0.13658E-07 0.27026E 01 0.16428E-07 0.16809E-07
 0.24988E 01 0.20374E-07 0.20786E-07 0.22961E 01 0.24865E-07 0.25379E-07
 0.21000E 01 0.28522E-07 0.29103E-07 0.19323E 01 0.30968E-07 0.31514E-07
 0.18125E 01 0.33578E-07 0.34282E-07 0.17095E 01 0.37477E-07 0.38431E-07
 0.16100E 01 0.42720E-07 0.43524E-07 0.15107E 01 0.48217E-07 0.49228E-07
 0.14121E 01 0.53693E-07 0.55281E-07 0.13096E 01 0.60887E-07 0.62015E-07
 0.12047E 01 0.70067E-07 0.70891E-07 0.11069E 01 0.80199E-07 0.81194E-07
 0.10066E 01 0.90676E-07 0.91506E-07 0.90694E 00 0.97262E-07 0.98722E-07
 0.81135E 00 0.91726E-07 0.93550E-07
```

Fig. 13.4. Listing of punched card output generated by sample problem 2.

```
//DIISAMP3 JOB (20106),'SAVE6522,55 DTI 6025'
//*CLASS CPU33=020S,REGION=380K,IO=1.5
/*ROUTE  PRINT LOCAL
/*ROUTE  PUNCH REMOTE46
//FERD EXEC PGM=FERD,REGION=380K
//GO.STEPLIB DD DSN=X.DTI46103.UNFOLD,DISP=SHR
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=(A,,1001)
//GO.FT07F001 DD SYSOUT=B
//GO.COMPOUT DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=&&PLTOUT,
// SPACE=(4000,(50,50),RLSE)
//GO.FT02F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(TRK,(100,10)),
//   DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6447,BUFNO=1)
//GO.FT27F001 DD UNIT=3330-1,VOL=SER=ZX8888,DISP=SHR,DCB=BUFNO=1,
// DSN=X.DTICAT01.G878883
SAMPL3:  FERD GAMMA-RAY PROBLEM WITH MINIMUM INPUT
        87         88         88       2000
        27          0        1.0          0          0
        27          0          0          5          3        0.0        0.0
         8         26         46         59         79
         1          0          1          0          3          3          0
SAMPL3:  SM+UO2+15BG=15SS  RUNS 7792A,B  RAW CHANNEL DATA
.001.001.001.001.001.014.134.243.330.420.510.639.742.816.903.948.9881.001.00 FAF
      1024         10      *** RUNS 7792A,7792B SM+UO2+BG+SS ***
  0043  0072      3  0073  0112      4  0113  0172      6  0173  0252      8
  0253  0352     10  0353  0502     15  0503  0702     20  0703  0952     25
  0953  1012     30  1013  1212     40
   162000.    82414.5       12.4       12.8  0.007769  0.007763       0.01
DYNMAT     0  8 (1A4,I4,8F8.0)
 FILE: 7792A.DAT  DATE: 30-Sep-80  CHANNELS: 1024
       1          0          0          0          0          0          0          0
       9          0          0          0          0          0          0          2          1
      17         29        122        233        672       2571      82164    1068887     989419
      25     907809     840701     779283     732765     686941     645309     613066     578144
      33     550324     525883     502449     482420     465713     451373     437431     425956
      41     415282     404512     392177     379961     364796     349521     334525     320077
      49     306746     296872     286200     277625     269450     262202     254673     247685
      57     241172     236277     229828     227275     222501     218399     216636     211388
      65     208010     206021     201536     198135     194658     192082     188379     184679
      73     183198     180830     177937     175369     172491     169606     167543     164521
      81     162219     160063     157204     155715     153365     152266     149705     147518
      89     145235     143372     141283     139473     137289     135329     134058     132785
      97     130633     129237     127738     126225     124798     123410     121769     119703
     105     117441     117331     115706     114645     113016     111676     110353     109510
     113     108459     106640     105059     104639     102829     102091     100613      99333
     121      98661      97427      95751      95203      93646      92752      92068      90766
     129      90331      89794      88532      87678      86204      85731      84480      83501
     137      82790      82002      80884      80321      79988      78586      77694      77403
     145      76028      75285      74297      73792      72877      72633      71931      70535
     153      70470      69919      69004      68379      67271      66942      66216      65567
     161      65070      64447      63488      63275      62757      61856      61482      60097
     169      59963      59447      59176      58877      58247      57387      56515      56571
     177      55752      55082      54449      53956      53779      53863      52614      52227
     185      51975      51488      51117      50370      50313      49852      49853      48760
     193      49348      48326      47315      47381      47218      47118      46316      45573
     201      45730      45321      45265      44353      44415      43973      43441      43286
     209      42867      42749      41702      41481      41101      40622      40203      40115
     217      39371      38635      38461      38300      36825      36411      36742      36562
     225      35817      35216      34993      34605      34917      34104      33760      32999
     233      32771      32769      31806      31906      31475      30938      30608      30055
     241      30039      29749      29260      29311      29491      29326      28541      28771
     249      28400      28016      27771      27165      27154      25813      26769      26369
     257      26208      26132      25837      25643      25481      25246      25102      24674
```

Fig. 13.5. Input listing for sample problem 3.

```
     953       22       33       18       25       26       30       22       27
     961       23       26       17       21       19       25       12       15
     969       15       13       19       18       20       14       16       14
     977       15       20       15       13       21       12       13       11
     985       15       15       13       14       15       11       13       10
     993       15       12       11        7       12        7       10        4
    1001        5       14        8        5        7        6       11        8
    1009        7        3        5       10       11        4        4       11
    1017        3        3        3        4        4        9        5        5

DYNMAT     0  8 (1A4,I4,8F8.0)
 FILE: 7792B.DAT  DATE: 30-Sep-80  CHANNELS: 1024
       1        0        0        0        0        0        0        0        0
       9        0        0        0        0        0        0        0        0
      17        0        0        3        7       12     1864    61114    57876
      25    54670    51672    48578    46782    44525    42476    40812    39232
      33    37687    36564    35024    34436    33233    32625    32202    31077
      41    30763    29958    29429    28465    27525    26928    25526    24625
      49    23439    23104    22178    21575    21100    20253    19737    19454
      57    18778    18273    17887    17647    17291    16910    17005    16534
      65    16351    16066    15601    15886    15176    15080    15138    14636
      73    14332    14132    14262    13952    13673    13386    13252    13128
      81    12858    12842    12637    12373    12278    12040    11995    11724
      89    11890    11551    11389    11076    11021    10960    10725    10576
      97    10377    10503    10206    10054     9958    10005     9665     9733
     105     9614     9340     9348     9167     9212     8904     8846     8854
     113     8548     8688     8235     8291     8329     8197     7998     7877
     121     7755     7797     7426     7572     7358     7351     7295     7212
     129     7245     6925     7061     6929     6980     6616     6475     6389
     137     6518     6183     6183     6198     6269     6084     6055     5941
     145     5937     5570     5815     5567     5541     5491     5538     5395
     153     5327     5333     5076     5307     5035     4901     4929     4820
     161     4838     4744     4702     4756     4653     4661     4549     4430
     169     4316     4392     4218     4291     4168     4215     4194     4195
     177     4127     3993     4052     3994     3889     3783     3791     3741
                                .
                                .
                                .
     905        3        5        2        7        4        5        6        4
     913        4        6        2        0        3        3        2        3
     921        4        3        4        3        1        1        4        1
     929        1        2        2        0        1        1        3        1
     937        6        2        3        3        2        5        0        1
     945        2        2        2        3        0        1        3        1
     953        0        2        2        3        0        0        1        2
     961        0        2        2        0        0        1        1        0
     969        3        4        0        1        0        1        0        1
     977        1        4        0        1        1        2        1        0
     985        2        2        0        0        2        1        1        0
     993        1        3        0        1        0        2        2        1
    1001        0        0        3        0        3        0        1        0
    1009        0        0        1        1        1        0        0        0
    1017        0        0        0        0        1        1        0        0

//GLD EXEC GLDPLT,REGION=110K
//COMPIN DD DSN=&&PLTOUT,DISP=(OLD,DELETE)
//FT05F001 DD *
DRAW=1-END$
//
ENDINPUT
```

Fig. 13.5. -continued-

actual input foreground data is given followed by an edit of the "adjusted" foreground data, i.e., the foreground data which has been shifted for zero intercept and which has been re-channeled into the "nominal" gain structure. The same two edits are repeated for the background data, followed by the summary listing of the final binned pulse-height spectrum. This tabular listing gives the bin number, the normalized counts per bin, the corresponding errors is the binned counts, the range of channels used to construct the bin, the pulse-height values associated with the lower and upper boundaries of the bin, and the Forte Acceptance Factor (FAF) for each bin. The listing is completed with an edit of SCOFF, FUDGE, and REFLU, as described for sample problem 1. If IBPLT equals 1 or 3, an APLOT-type printer plot of the binned spectrum is given next.

For this sample problem, NOPT5 was input as 3 which results in a PPLOT3-type plot of every 20th window function. Each plot includes the upper and lower synthetic windows as well as the input window function. One should be prepared for considerable printed output if NOPT5=4.

A tabular edit similar to the B-ADJ edit in FERDO is given next. The columns of the table include: (1) the bin number and lower pulse-height value of the bin, (2) the original pulse-height spectrum and errors (B and S vectors), (3) the computed lower and upper bounds of B (BLO and BUP) which are consistent with the unfolded solution and the response matrix, (4) the difference between B and the average of BLO and BUP, and (5) the per cent difference, weighted by S. A PPLOT3-type plot of B-ADJ is given next.

The final edit is quite similar to the corresponding output from FERDO, and gives the PLO-PUP solution bounds with the relevant energy labels and a computed average, PAVE. A percent error in PAVE is also given along with the actual error in PLO and PUP. The last two columns of the edit are labelled "E PCT" and "Q PCT" which are analogous to the "ERR1" and ERR2" columns in the FERDO edit. This edit is followed by two line printer plots of PLO and PUP, and the DISSPLA-generated summaries of the external plots. The actual plots produced by this sample problem are shown in Fig. 13.6.

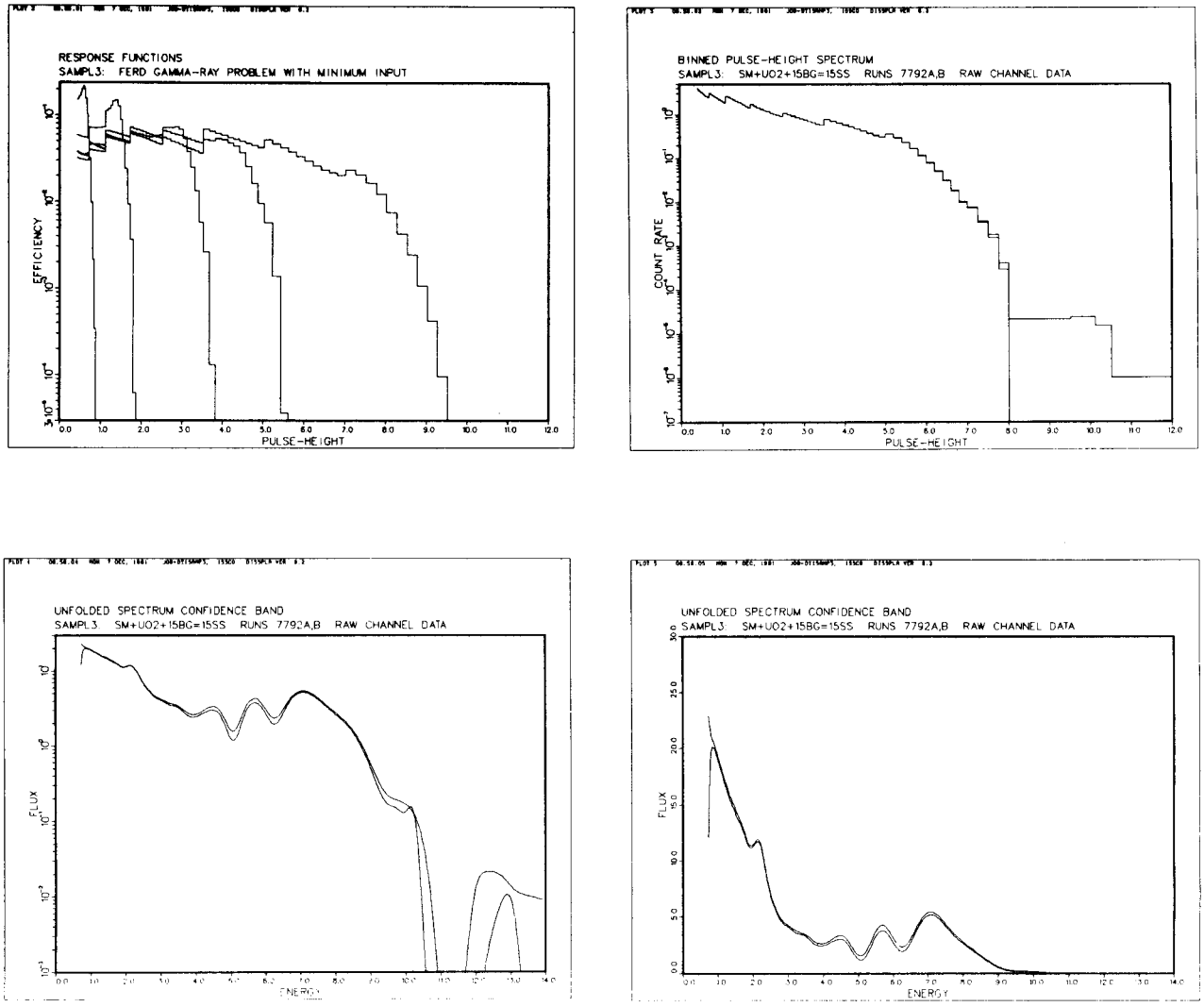


Fig. 13.6. Plot output from sample problem 3.

ACKNOWLEDGEMENTS

The authors would like to thank Drs. J. E. Cope and D. P. O'Leary for their advice and encouragement.

REFERENCES

Beyer, W. H., ed.: 1968, CRC Handbook of Tables for Probability and Statistics, 2nd Edition, The Chemical Rubber Co., Cleveland.

Burrus, W. R.: 1960, "Unscrambling scintillation spectrometer data," IRE Trans. on Nuc. Sci., NS-7, pp 102-111.

Burrus, W. R.: 1961a, "Unscrambling of continuous scintillation spectra," Aircraft Nuclear Propulsion Project Semiannual Progress Report, ORNL-3144, pp 100-112.

Burrus, W. R.: 1961b, "Unscrambling of scintillation spectra," Neutron Physics Div. Annual Progress Report for Period Ending Sept. 1, 1961, ORNL-3193, pp 44-52.

Burrus, W. R.: 1962, "Use of mathematical programming in unfolding instrument response," Bull. Amer. Phys. Sec. II, 7, p. 9.

Burrus, W. R.: 1963, "Inequality method of unfolding for nuclear spectroscopy," Trans. Amer. Nuc. Soc., 6, pp. 173-174.

Burrus, W. R.: 1965, Utilization of A Priori Information by Means of Mathematical Programming in the Statistical Interpretation of Measured Distributions, ORNL-3743.

Burrus, W. R.: 1976, "FERD and FERDOR type unfolding codes," in A Review of Radiation Spectra Unfolding (D. K. Trubey, ed.), ORNL/RSIC-40, pp. 1-22.

Burrus, W. R., Rust, B. W., and Cape, J. E.: 1980, "Constrained interval estimation for linear models with ill-conditioned equations," in Information Linkage Between Applied Mathematics and Industry II (A. L. Schoenstadt et al., eds.) Academic Press, New York, pp. 1-38.

Burrus, W. R. and Verbinski, V. V.: 1965, "Recent developments in the proton-recoil scintillation neutron spectrometer," ANS-SD-2, pp. 148-185.

Burrus, W. R. and Verbinski, V. V.: 1969, "Fast-neutron spectroscopy with thick organic scintillators," Nucl Instr. and Methods, 67, pp. 181-196.

Bogert, V. D. and Burrus, W. R.: 1962, "The SLOP code for unfolding of instrument response," Neutron Physics Div. Annual Progress Report for Period Ending Sept. 1, 1962, ORNL-3360, pp. 22-31.

Hoerl, A. E. and Kennard, R. W.: 1970, "Ridge regression: Biased estimation for nonorthogonal problems," Technometrics, 12, pp. 55-67.

Kendrick, H. and Sperling, S. M.: 1970, An Introduction to the Principles and Use of the FERDOR Unfolding Code, GA-9882.

Mood, A. M. and Graybill, F. A.: 1963, Introduction to the Theory of Statistics, McGraw-Hill, New York.

Peelle, R. W.: 1971, "Techniques Used at Oak Ridge National Laboratory for Unfolding Neutron and Gamma-Ray Pulse-Height Spectra," ORNL/TM-3463.

Phillips, D. L.: 1962, "A technique for the numerical solution of certain integral equations of the first kind," Jour. of the ACM, 9, pp. 84-97.

Rust, B. W.: 1976, "Mathematical foundations of the Burrus techniques for spectral unfolding," in A Review of Radiation Energy Spectra Unfolding (D. K. Trubey, ed.) ORNL/RSIC-40, pp. 23-32.

Rust, B. W. and Burrus, W. R.: 1971, Suboptimal Methods for Solving Constrained Estimation Problems, DASA-2604.

Rust, B. W. and Burrus, W. R.: 1972, Mathematical Programming and the Numerical Solution of Linear Equations, American Elsevier Publ. Co., New York.

Tikonov, A. N.: 1963, "Solution of incorrectly formulated problems and the regularization method," Soviet Mathematics Dokl., 4, pp. 1035-1038.

ORNL/TM-8720

INTERNAL DISTRIBUTION

1-5. L. S. Abbott
6. R. G. Alsmiller
7. D. E. Bartine
8. G. T. Chapman
9. J. K. Dickens
10. G. F. Flanagan
11. C. Y. Fu.
12. T. A. Gabriel
13. R. Gwin
14. J. A. Harvey
15-19. D. T. Ingersoll
20. J. O. Johnson
21. D. C. Larson
22-26. F. C. Maienschein
27. B. F. Maskewitz
28. B. L. McGill
29. F. J. Muckenthaler
30. R. W. Peelle
31. F. G. Perey
32. R. W. Roussin
33. R. T. Santoro
34. D. K. Trubey
35. C. R. Weisbin
36. L. W. Weston
37. A. Zucker
38. P. W. Dickson, Jr. (Consultant)
39. H.J.C.Kouts (Consultant)
40. W. B. Loewenstein (Consultant)
41. R. Wilson (Consultant)
42-43. Central Research Library
44. Y-12 Document Reference Section
45-46. Laboratory Records
47. Laboratory Records ORNL RC
48. ORNL Patent Office
49-53. EPD Reports Office

EXTERNAL DISTRIBUTION

54. Office of Assistant Manager for Energy Research and Development, DOE-ORO, Oak Ridge, TN 37830
55-59. Walter R. Burrus, Science Applications, Inc., 800 Oak Ridge Turnpike, Suite 801, Oak Ridge, TN 37830
60. Richard Bellman, Depts. of Math. and Eng., University of Southern California, University Park, Los Angeles, CA 90007
61. Alfred S. Carasso, Admin. Bldg., Rm. A302, National Bureau of Standards, Washington, D.C. 20234
62. Allan D. Carlson, Radiation Physics Bldg., Rm. B-113, National Bureau of Standards, Washington, D.C. 20234
63. Peter E. Castro, Management Services Div., Eastman Kodak Company, Rochester, NY 14650
64. Burton H. Colvin, Admin. Bldg., Rm. A-437, National Bureau of Standards, Washington, D.C. 20234
65. Jane E. Cope, EG&G ORTEC, 100 Midland Road, Oak Ridge, TN 37830
66. Lloyd A. Currie, Chemistry Bldg., Rm. A-209, National Bureau of Standards, Washington, D.C. 20234
67. Vance Faber, Los Alamos National Laboratory, MS B-265, P. O. Box 1663, Los Alamos, NM 87545
68. Henry Fleming, National Earth Satellite Service, NOAA, Code S/RE 21-B, Federal Bldg. #4, Rm. 0211, Suitland, MD 20233
69. Robert W. Gerlach, Chemistry Bldg., Rm. B-226, National Bureau of Standards, Washington, D.C. 20234
70. Gene H. Golub, Dept. of Computer Science, Stanford University, Stanford, CA 94305

71. Richard J. Hanson, Numerical Mathematics Div., Sandia National Laboratories, Albuquerque, NM 87185
72. Dr. N. Hertel, 167 Taylor Hall, University of Texas-Austin, Austin, TX 78712
73. Glenn R. Ingram, Bldg. 225, Rm. A-151, National Bureau of Standards, Washington, D.C. 20234
74. Ronald G. Johnson, Radiation Physics Bldg., Rm. B-112, National Bureau of Standards, Washington, D.C. 20234
75. Jack Lagnese, National Science Foundation, Applied Mathematics Program, 1800 G Street, N.W., Washington, D.C. 20550
76. NBS Library, Administration Bldg., National Bureau of Standards, Washington, D.C. 20234
77. W. H. Miller, 1033A Engineering, University of Missouri-Columbia, Columbia, MO 65201
78. G. L. Morgan, MS D406, Los Alamos National Laboratory, Los Alamos, NM 87545
79. Dianne P. O'Leary, Computer Science Department, University of Maryland, College Park, MD 20742

80-84. Bert W. Rust, Bldg. 225, Rm. A-151, National Bureau of Standards, Washington, D.C. 20234

85. Francis Sullivan, Bldg. 225, Rm. A-151, National Bureau of Standards, Washington, D.C. 20234
86. Dominic Vecchia, National Bureau of Standards, 325 Broadway, Boulder, CO 80303
87. Grace Wahba, Dept. of Statistics, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706
88. B. W. Wehring, 214 Nuclear Engineering Lab, University of Illinois at Urbana-Champaign, Urbana, IL 61801

89-115. Technical Information Center (TIC)

116. Dr. Joze Rant, Univerza v Ljubljani, Institut Jozef Stefan, Ljubljana, Yugoslavia.