Unfolding Neutron Spectra: LOUGHI for Pedestrians

Rindi, A

1977-04-01

Unfolding Neutron Spectra: LOUGHI for Pedestrians

A. Rindi

April 1977

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY

Prepared for the U.S. Department of Energy under Contract W-7405-ENG-48.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-6413

UNFOLDING NEUTRON SPECTRA: LOUHI FOR PEDESTRIANS

A. Rindi

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

1. Introduction

In 1969, Jorma Routti, a Ph.D. student with the Health Physics Department, wrote a very elegant and versatile computer program for the unfolding neutron spectra from measurements performed with threshold activation detectors, multisphere detectors and proton recoil in emulsions as well as for other interesting applications.

In 1970, Routti left LBL, slightly changed his fields of interest and forgot to write a final and complete "manual" on how to understand and use that excellent program. In spite of that, several persons have used and keep using it; they are generally, skilled computer experts, those peculiar people who are able to read (and enjoy) pages of FORTRAN or other ascetic languages like more "down to earth" people read classical poems or Mickey Mouse cartoons.

It is my opinion that every physicist or health physicist involved with spectral measurements has to be able to get advantage of unfolding programs such as LOUHI or other similar excellent programs. With this, I mean to use them as he uses, for instance, a multichannel analyzer where he is very well aware of what is going into the instruments from the input connectors, what the instrument is supposed to do, what its use is, and how and when to push the different buttons on the panels, and what to expect as output on the TV screen or on the teletype without having a very deep knowledge of the electronics circuitry behind the panels.

It was with this purpose in mind that I spent some time, first, by trying to understand the notes (often criptic to me) that Routti left, to interpret his suggestions, to look into the core of LOUHI, and to use LOUHI.

J. Routti has warned me, and I have become aware of it by reading the program, that the program itself is by no means complete in the "programming art" sense: there are several repetitions and redundancies,the input and output can be improved, the computer time it uses can be shortened. However, as it is now, it is already a perfectly operating program. I leave the privilege of perfecting it to someone else. I am not a skilled programmer, not a computer expert, nor a computer fan. I wrote what follows in the hope that it can be of use to other users.

I hope the original author of the program, now Professor Routti, will excuse the many oversimplifications, approximations, and perhaps mistakes of my interpretation.

2. The Mathematics of LOUHI

2.1. Generalities

The general goal in unfolding neutron spectra from threshold measurements, be they p-recoil measurements, activation measurements, moderating multisphere measurements, or others, is the solution of an integral equation of the type

$$A(E') = \int_{E_{min}}^{E_{max}} K(E',E)\ \Phi(E)dE \tag{1}$$

known as the first order Fredholm integral equation, where $\Phi(E)$ is the neutron energy spectrum to be determined, E_{min} and E_{max} are the lower and higher energy limits of the spectrum, $K(E',E)$ is an analytical function expressing the kernel or response function of the detector used, and $A(E')$ is an analytical function expressing the observed (measured) response (e.g. the proton spectrum in the case of a p-recoil measurement). However, with the exception of a proton recoil measurement, where the measured data may be expressed as a continuous recoil spectum, all the other threshold detectors provide only a series of discrete measured values for the $A(E')$. The kernel K also is generally not known analytically but rather as a set of discrete points.

We will focus our interest in the following on threshold detectors. The general equation for one activation detector j is given by

$$A_j = \int_{E_{min}}^{E_{max}} K_j(E)\ \ (E)\ dE \tag{2}$$

where $K_j(E)$ is the analytical response function of the detector j (the cross section as a function of the energy for the particular reaction used), A_j is the measured response of the detector, $\Phi(E)$ is the unknown neutron spectrum and E_{min} and E_{max} are either the limits of sensitivity of the detector or the limits of the neutron spectrum.

As stated above, in practical cases $K_j(E)$ is known only as a series of discrete points. Also, as a consequence, the neutron spectrum will be calculated as a series of points. Equation (2) will then be written

$$A_j = \sum_{i=1}^{N} K_j(E_i)\Phi(E_i) \tag{3}$$

where N is the number of points where the response function is given and consequently the spectrum is calculated. Given M detectors, one shall solve a system of M equations of the type (3). There are several methods for the numerical solution of a system of that type; some have been described by Routti (Ro69a). In the following we will explain the method used in LOUHI.

2.2. The Minimization

The program uses a "generalized least-square method with nonnegative solution". It is summarily described in Ro69a, Ro69b and Ro72.

Suppose we have M different detectors: we will then have M equation of the type (3), one for each detector. It has to be kept in mind that if M<N, as is usually the case, the system of M equations (3) with N unknowns $\Phi(E_i)$ has no unique solution. If no restrictions are placed on the solution, one can find solutions that are not physically acceptable such as negative solutions or solutions that show nonphysical oscillations.

In LOUHI the nonnegativity of the solution is imposed by finding as a solution a series of N values $X(E_i)$ such that $\phi(E_i) = [X(E_i)]^2$. Other constraints in the solutions are used as we will see later.

What LOUHI does is to find appropriate approximate values for $X(E_i)$ that minimize an expression containing different terms that constrain the $X(E_i)$ to the measured responses, to a certain smoothness and to a given shape. This minimization is achieved by successive iterations.

For simplicity and clarity in what follows, $\Phi(E_i)$, $X(E_i)$ or similar discrete functions of the energy E_i or components of an N dimensional vector Φ or X will be indicated with Φ_i, X_i, etc.

Let us consider the quadratic expression

$$Q = Q_0 + \gamma\ (W_1\ Q_1 + W_2\ Q_2) \tag{4}$$

where

$$Q_0 = \sum_{j=1}^{M} r_j^e \left(A_j^m - A_j^c\right) \tag{5}$$

$$Q_1 = \sum_{i=1}^{N} r_i^p \left(X_i^2 - \phi_i^0\right)^2 \tag{6}$$

and

$$Q_2 = \sum_{i=2}^{N-1} r_i^d \left(X_{i-1}^2 - 2\ X_i^2 + X_{i+1}^2\right)^2 \tag{7}$$

In equation (5), A_j^m indicates the measured response of the detector j, r_j^e is a weighting factor (whose possible expressions will be shown in paragraph 2.3). A_j^c is a calculated response for the detector j; it is calculated by an expression similar to equation (3).

$$A_j^c = C_j \sum_{i=1}^{N} K_{j,i}\ X_i^2 \tag{8}$$

where X_i^2 indicates the solution vector that we now call the approximate solution, $K_{j,i}$ is the response function of detector j and C_j is a normalizing factor for detector j which depends on the units chosen for expressing A_j^C, $K_{j,i}$ and X_i.

The smaller the value of Q_0, the closer the measured and calculated responses are, i.e. the closer the approximate solution is to the sought experimental spectrum which generated the measured responses A_j^m. So the first term of (4) constrains the approximate solutions to the measured responses. As we said before, the solutions have to be also constrained to some criteria of shape and smoothness. This is done by the terms Q_1 and Q_2.

The term Q_1 expresses the "distance" of the approximate solutions X_i from a given function ϕ_i: this deviation from a given shape can be weighted, as a function of the energy, with the variable weight r_i^p.

The term Q_2 expresses the numberical second derivation of the approximate solutions X_i. In a simplified view it can be seen as the distance between three consecutive values of the X_i. This criterion too can be weighted as a function of the energy with the variable r_2^d.

By minimizing Q_1 and Q_2 one forces the solution to a given shape and to smoothness, because one imposes to the X_i to be as close as possible to the function ϕ_i^o (given shape) and as close as possible to each other (smoothness). By minimizing Q_0, one constrains the solutions X_i to the measured responses A_j^m. The program LOUHI minimizes the Q by an iterative perturbation method.

In neutron spectroscopy the spectrum often extends over many orders of magnitude and is plotted using semilog or loglog scale. This makes it desirable to have the choice to tie the shape, smoothness, and minimum deviation criteria on a linear, on a relative, or on a logarithmic scale. The term Q_1 will then be expressed by (6) on a linear scale or by

$$Q_1 = \sum_{i=1}^{N} r_i^p \left(\frac{X_i^2-\phi_i^0}{X_i^2}\right)^2 \qquad (9)$$

on a relative scale; or

$$Q_1 = \sum r_i^p \left(\log X_i^2-\log \phi_i^0\right)^2 \qquad (10)$$

on a logarithmic scale.

Similarly, the term Q_2 is expressed by (7) on a linear scale or by

$$Q_2 = \sum_{i=2}^{N-1} r_i^d \left(\frac{X_{i-1}^2 - 2X_i^2 + X_{i+1}^2}{X_i^2}\right)^2 \qquad (11)$$

$$Q_2 = \sum_{i=2}^{N-1} r_i^d \left(\log X_{i-1}^2 - 2 \log X_i^2 + \log X_{i+1}^2\right)^2 \qquad (12)$$

in the relative and logarithmic scales respectively.

With the symbols γ, W_1, W_2, r_i^e, r_i^p and r_i^d we represent possible weighting factors for the different constraints: they are set at the beginning by data cards. In paragraph 2.3 we will show usual expressions for them.

Finding the series of values of X_i that minimize Q, eq. (4) considered as a function of N variables X_i means finding the solution that is closer to the real spectrum and best matches the imposed criteria.

Mathematically, the search for the minima of a function is performed by finding the values of the variables for which the first derivatives of the function are zero and the second derivatives are positive. Analogously in LOUHI one looks for the values of X_i for which the derivative of equation (4).

$$F_i = \frac{\partial Q}{\partial X_i} \qquad (13)$$

are closest to zero and the Hessian matrix of the second derivatives

$$[H] = \left[\frac{\partial^2 Q}{\partial X_i \, \partial X_K}\right] \qquad (14)$$

is definite and positive (the rather lengthy expressions for F_i and [H] are not given here for sake of simplicity.*) The search for the

*They are computed in the subroutines READIN, FCN and RQ of LOUHI.

values of X_i that minimize Q is performed in LOUHI by an iterative perturbation method: it has been described by Davidon (Da 59) and coded by Beals in a computer routine called VARMIT. LOUHI uses a slight modification of this routine. This method is rather sophisticated and cannot be described in detail here. However, we will give a very simple account of what it does.

At every step of the iteration, the values of the first derivatives F_i and of the Hessian [H] are compared with those of the previous step. If certain conditions that we will see in the following are matched, the process is stopped and the approximate solutions X_i^2 found at that step are written in the output as final solution. If not, a perturbation is added to the X_i^2 and these X_i^2 perturbed become the new X_i^2 for which the values of F_i and [H] are computed and so on. The values of the perturbations are suitably computed in VARMIT and depend on how close the F_i are to zero, on the value of [H] and on other conditions.

The process starts and proceeds in the following way. First, one inputs into the program a starting function P_i which has a shape similar to the spectrum solution one is seeking and to which one wants to tie the solutions (e.g. a function of the type 1/E for cosmic neutron or accelerator neutron measurements). This is done through the data card P. If P_i is not input as a series of discrete values but as an analytical function, the program, for each energy value E_i (specificed with data cards KFIX or KERNEL), calculates the corresponding values P_i. These are the first values for the X_i^2. The program then computes the first values for Q, F_i and [H). Note that in this first step Q_1 will be equal to zero because there are not yet values for the ϕ_i^0. After that, the P_i are perturbed, i.e. a suitable positive or negative quantity is added to each P_i. The quantity to add is determined by suitable mathematical rules in the routine VARMIT. The perturbed P_i become the new X_i^2. Then the new values of Q, F_i and [H] are computed. The values for the ϕ_i^0 to be used for computing Q_1 will be at this step the initial values of P_i.

New and old values of F_i and [H] are compared: the new F_i may be closer or farther to zero than the old F_i and the new [H] may be more positive (larger) or smaller than the old [H]. As a consequence of this, a new perturbation is added to the X_i^2 that becomes the new X_i^2. New values for Q, F_i, and [H] are computed and compared with the previous ones. The values for the ϕ_i^0 to be used for computing Q_1 may be either the previous X_i^2 or the value of the starting function P_i (this is decided with the card MAKEPF). From the comparison of the new and previous F_i and [H] a new suitable value for the perturbation is found and the new iteration starts. The iteration process is terminated when

a. The values of Q remain the same for 4 successive iterations, or

b. Every value X_i^2 differs from the correspondent of the previous iteration by less than 10^{-8}.

In such a case one says that the iterations converged. The process is also stopped after 100 iterations, but in such a case there is no convergence.

2.3. The Weighting Functions

As shown in equations (4) to (12), equation (4), that is minimized in the program, contains six parameters that allow to weight the constraints applied to the solution. In other words, one can force the solution to be more or less smooth, to show more of less oscillations, to be more or less close to the measured values, to be more or less close to a given shape, etc. It has to be kept in mind that forcing the solution to one criterion may loose up the constraint to another criterion, e.g. a solution too smooth or too close to a 1/E shape may be a little too far from the measured values, i.e may give some calculated responses rather different from the measured ones. It is not easy to find a good balance between all the weighting functions.

These weighting functions, as indicated in equations (4) to (12) are:

r_j^e that force the solution to the measured values by weighting the difference between the measured and the calculated responses. We remind that the calculated responses are the values that would be measured by each detector if the experimental spectrum would be that indicated by the approximate solutions. There are as many r_j^e as detectors. They are set by the data card ESCALE.

r_i^p that force the solution to be close to a given solution. The number of r_i^p's is equal to the number of energy points. They are set by the car PSCALE.

r_i^d that force the solution to smoothness or minimal oscillations by weighting the difference between three consecutive components of the soltuion. The number of r_i^d's is equal to the number of energy points. They are set by the card DSCALE.

W_1 and W_2 that weight the overall relative importance of the two constraints expressed by Q_1 and Q_2 in equation (4). W_1 and W_2 have just one value each and are set through the card WEIGHT.

γ that weights the overall importance of the two constraints expressed by Q_1 and Q_2 in equation (4) relative to the constraint expressed by Q_0. γ has only one value that is set by the card GAMMAS.

Because r_j^e may be given a value for every different detector and r_i^d and r_i^p for every energy point, they can be expressed as a function of the measured responses of the detector or of the given function ϕ_i^o respectively.

The general expression that indicates the values which can be given to the functions r_j^e, r_i^p and r_i^d is

$$r_k = \frac{w_k \, 10^h}{c \, (f_i)^\ell} \qquad (15)$$

where

w_k can be put = 1 or to a value read in. It can have a value for each detector.

h can be put= 0 or to one value read in (the value of WHAT (2) in the cards DSCALE, ESCALE, OR PSCALE).

c can be put= 1 or to the number M of detectors for r_j^e or the number N of energy points for r_i^d and r_i^p.

f_i can be put equal to A_j^m, the values of the measured responses, for r_j^e or to ϕ_i^o for r_i^d and r_i^p.

ℓ can have values 0, 1, or 2.

W_1, W_2, and γ will be given a value in the cards WEIGHT and GAMMAS respectively. It can be any value.

It is my personal opinion that there is a certain redundance in all these weighting factors that can confuse the normal LOUHI user. They were introduced in the program to study the effects of the different weights on the accuracy of the match and the shape of the solution. However, a normal user may not be interested in making too many checks once he found a solution which matches well enough the experimental data and the other criteria. It is, however, instructive, as one gets acquainted with using LOUHI, to make some checks.

In the paragraph devoted to the explanation of the data cards use, some typical values for these weights are indicated. It is useful to remark that, in most cases the solution obtained is not very dependent on the smoothness requirement. That is to say that even a weak smoothness condition is enough to suppress the fluctuations in the solution, in particular, when such fluctuations do not improve the match between the measured and calculated responses. The Q_1 condition is not usually needed in neutron spectroscopy problems but can be used to specify cutoff energies, etc.

3. How LOUHI Works

3.1. Generalities

The program LOUHI is controlled by a series of data cards (see Chapt. 4) by which the response functions of the detectors used, the energy values where the solution has to be calculated, the measured responses of the detectors, the different weights and other options are inputted. A data card GO starts the iteration process.

The LOUHI code includes also provisions to internally test any solutions. A solution is sought using the calculated responses as input and by perturbing them with radom errors. The new solution obtained is then compared with the one which gave the calculated responses. However, we do not describe here this check possibility--details about it can be found in Ro69.

The program can be run on line with the computer o off line. In the on line run, one can show on the cathode ray display a continuous graph of the $\phi_i = X_i^2$ as function of energy, that is the solutions calculated after one or more series of iterations. The values of the measured responses and of the calculated responses for each detector are also displayed. The operator then judges whether or not the solutions are smooth enough, the shape of the spectrum is physically acceptable and the match between the calculated and measured responses is satisfactory. He can decide to tie some values of the solution to certain values for smoothing further the spectrum or for slightly modifying its shape or for trying to obtain a better match for some detectors. Or, else, he can decide to vary the values of some weights, eliminate some detectors, increase the number of iterations, vary the energy limits, etc. This is done by typing in new data cards through the teletype. After a new GO data card has been typed in, the program goes through the new series of iterations and the new solutions are displayed.

To interact on line with the computer requires a certain skill and familiarity from the operator. The use of the various data cards will be explained in the following chapters.

In the off line mode the program normally outputs a table with the final values of the ϕ_i, the integrated flux, absorbed dose and dose equivalent, the values of the measured and calculated responses for the detectors used, and, optionally, a Cal Comp plot of some of these functions.

It also prints some quantities calculated during the iterations that give an idea of the grade of convergence and of matching reached.

3.2. The Core of LOUHI

The program is written in FORTRAN IV language. It articulates into 30 subroutines and 3 functions. Their names are, in alphabetical order. ADOSE (function), AFIX, CALCF, CCLOGTC, CCLOUHI, CLOCKI, CSTRAN, DAFILE, DIFCK, ERROR, FCN, FDOSE (function), FETHELP, GRAPH, INTER, KERNEL, LINEOUT, MATMPY, MINMAX, MPRINT, MTMPY, OLGA, OVERFL, READER, READIN, RESULT, RITEOT, RNDF (in ascent), ROUND (function), RQ, SCALE, VARMIT, WEIGHT.

The program also uses several system subroutines, i.e. subroutines provided by the Berkeley computer library (the user from another center has to check if these routines exist in his computer library and if they have the same name).

In the following, the function of some of the LOUHI subroutines are briefly explained. They are listed in the order they appear in LOUHI.

LOUHI (Main Program): Initializes some variables, writes some initial output and gives the control first to the subroutine READER.

READER: Reads in the data cards and switches to various parts or other subroutines according to the title of the card.

KERNEL: It is reached by the cards KREAD and KFIX. Reads in the response function for the different detectors at the specified energy points and reads in the energy points at which the solutions have to be calculated.

INTER: Interpolates the values of the response function of the detectors at the energy values where the solutions are calculated.

SCALE: Calculates the different weights.

RQ: Calculates some expressions used for evaluating the calculated responses of the detectors. (See Ro69a, p. 104)

AFIX: It is reached by the card KERNEL and reads in the response function and the energies when that card is used.

WEIGHT: Was assigning quadrature weighting for integrating schemes. However, part of these calculations are presently done in the subroutine RESULT.

FCN: Calculates Q_0, Q_1 and Q_2, and Q following the expressions (4) to (12) and the F_i of (13).

RESULT: Prints out the vast majority of the results. In particular, it prints the solution function $\phi_i = X_i^2$ (called differential flux in the output print) calculates and prints the integral flux, integral absorbed dose, integral dose equivalent and several other quantities.

FDOSE Function: Calculates the flux-to-dose equivalent conversion factors following the expression proposed in Ri74.

MPRINT Prints the values of some variables, according to the card PRINT.

ADOSE Function: Calculates the flux-to-absorbed dose conversion factors according to the expression proposed in Ri77.

MTMPY and MATMPY: Perform some minor computations.

GRAPH: Plots graphs.

VARMIT
ERROR
READIN : Performs the iterations for the minimization.

RITEOUT: Writes the values of several intermediate variables in the output.

OLGA: Set the kernels for unfolding spectra from proton recoil measurements in photographic emulsions.

The other subroutines direct the plotting of the results in the Cal Comp or on the TV screen.

3.3. DATA INPUT

Program LOUHI is self-contained. The user needs only supply the data and indicate which options and calculations are desired. All parameters used are initialized, they can be reset at any time and they will remain at the new values until further altered. The usage of the program is controlled by LOUHI data cards, the order of which should follow the desired operation. For uniformity, all these cards have the same format, this also implies that the numbers which would ordinarily be integers are read in as decimals; they will be converted into integers by the program.

The LOUHI data cards are written in FORMAT(A10, 6E10.0, A10). The first word is a left-adjusted codeword, CODEW, which identifies the function of the input card. Up to six parameters, (WHAT(I), I=1, 6), follow the codeword, the numbers have different meanings for different codewords. The last alphanumeric word, NAME, may be used to identify the data card, it is not used by the program, except when specified below. The data cards, in alphabetical order according to the codewords that identify them, are listed in the following: their function and the function of the associated parameters are explained.

Note that the order of the data cards in the program will follow some priority rules according to the desired operation.

CCPLOT Makes a Calcamp plot of the results,

WHAT(1) = 1. Plots the solution vector,
WHAT(2) = 1. plots the confidence limits of the solution vector,
WHAT(3) = 1. plots the test solution,
WHAT(4) = 1. plots the approximate solution,
WHAT(5) = 1. plots the proton spectrum,
WHAT(6) = 1. plots the uncertainties of the proton spectrum,

If one does not want some of these options, put the corresponding WHAT equal to 0., i.e. do not punch anything in that part of the card.

$$\text{NAME} \begin{cases} = \text{LINLIN} & \text{specifies linear energy and intensity scales,} \\ = \text{LINLOG} & \text{specifies linear energy and logarithmic intensity scale,} \\ = \text{LOGLIN} & \text{specifies logarithmic energy and linear intensity scale,} \\ = \text{LOGLOG} & \text{specifies logarithmic energy and intensity scales.} \end{cases}$$

Note: NAME of the card is used by the program.

CPTIME Prints the central processor time elapsed and specifies automatic display of the results during on-line runs,

WHAT(1) = 1. for no display of the results, this is the initial value,
= 2. for automatic VISTA display of the results.

Note: This card is not required during the run off line. During the run on line it is used to have an idea of the computer time that will be charged to the user or of the computer time that remains if a fixed amount of time has been allocated.

DATA Reads in a measured response,

WHAT(1) If one uses the card KERNEL for inputting the response of the detectors (the K_j of eq. 3), it specifies the number of the response function read in with the KERNEL card, i.e. the detector's number (in the order in which it appears in the cards following the KERNEL card) that has measured response WHAT(2).
If instead, one uses the KREAD and KFIX cards, it is a progressive number from 1 to MC (see KFIX card).

WHAT(2) is the measured response, that is A_j in equation (3)

WHAT(3) is the normalizing factor, that is C_j in equation (8),

WHAT(4) gives an option of setting the weighting factor that is r_j^e of equation (5).

It puts
$$r_j^e = \frac{\text{WHAT}(4)}{\frac{\text{WHAT}(2)}{\text{WHAT}(3)}}^{2} = \frac{\text{WHAT}(4)}{(A_j^m)^2}$$

This option is redundant with the options given with the card ESCALE.

It is initialized = 1. We suggest to neglect this option.

Note: A_j is the reading of the detctor j. It can be expressed in any unit provided one gives the normalizing factor C_j a correspondent value. If we want the results ϕ_i as flux density in, i.e., neutrons cm^{-2} s^{-1}, C_j has to be expressed in units such as to satisfy the expression

$$A_j = C_j, \sum_{1}^{N} \phi_i \ \sigma_{i,j}$$

where $\sigma_{i,j}$ is the response function of the detector j as it has been introduced with the cards KERNEL or KREAD.

For activation detectors, generally, the response function is the cross section of the detector (expressed in b or mb) and A_j is expressed as activity (in cpm) at saturation. A calibration for that particular detector at a given energy provides the activity at saturation per unit flux at a given energy (calibration factor in cpm per unit flux); our normalizing factor C_j will, then, be this calibration factor divided by the cross section at that energy. C_j will then have the units of cpm per unit flux per unit cross section. Analogously for other detectors, the normalizing factor C_j is generally the calibraion factor for that detector divided by the value of the response function of that detector at the energy point where the calibration has been performed.

Remember that, inside the program,the measured response of the detector A1(J) is put

$$A1(J) = \frac{A_j}{C_j}$$

DSCALE Specifies the energy-dependent weighting terms for the second difference criteria, r_i^d, in equations (7), (11), or (12). The scaling is performed as with PSCALE.

ESCALE Specifies the weights for the responses r_j^e in equation (5)

WHAT(1) = 1. sets $r_j^e = 1.$,
= 2. sets $r_j^e = 1./M$,
= 3. sets $r_j^e = 1./A_j$,
= 4. sets $r_j^e = 1./(M\,A_j)$,
= 5. sets $r_j^e = 1./A_j^2$,
= 6. sets $r_j^e = 1./(M\,A_j^2)$, → normal choice
= 7 to 12 reads in the following cards, in FORMAT (8E 10.0), M values for the W_K of eq. (15). Then it sets the r_j^e as if they were given by a WHAT(1) = WHAT(1)-6 and multiplies them by the corresponding W_K.

WHAT(2) ≠ 0. multiplies the weights obtained above by $10.^{\text{WHAT(2)}}$

FORMAT Reads in on the next card the format for reading in the kernel, for instance (6E12.3), initialized (8E10.0).

FREEP Specifies an iterative mode for the computation, in which the approximate solution is set equal to the solution after each minimization, and computation is repeated until the convergence criteria specified has been met,

WHAT(1) sets the convergence limit equal to WHAT(1), iterations are repeated until the approximate and the solution vector differ by less than WHAT(1) at any point,

WHAT(2) specifies the maximum number of iterations.

Note: Do not use this card unless one is very familiar with the program.

GAMMAS Specifies the value of the overall weight, γ, in equation (4),

WHAT(1) is the value of γ.

GO Performs teh minimization and prints and displays the results. It is the last card of a series of instructions. It starts the iteration process.

GUESS Specifies the initial guesses required by the minimization algorithm for the solution vector. These guesses are the starting values that the approximate solution X_i has to have, such that the minimization iterations can start, i.e. such that the first values for equations (13) and (14) are computed.

WHAT(1) = 1. reads in the following cards the intial guesses for the number of steps specified by KERNEL or KFIX and STEPS, in FORMAT (8E10.0),

= 2. obtains the guesses from the starting function P_i set by the card P

= 3. sets the guesses equal to the solutions obtained in the previous calculations,

= 4. obtains the guesses from the test solution.

IOFILE Specifies the input file for reading in LOUHI data control cards, this can be changed at any time during on-line runs,

WHAT(1) = 2. specifies the card reader, this is the initialized value,

= 6. specifies the teletype input,

WHAT(2) = 1. displays the dayfile on the cathode-ray-tube during on-line runs.

KERNEL Reads in the kernel composed of the response functions,

WHAT(1) is the number of response functions, that is M in equation (5),

WHAT(2) is the number of energy bins, that is N in equation (3). The response functions follow the KERNEL card, each starting with a label on the first card in FORMAT(A10) followed by N numbers written in the format specified by FORMAT card. The last response function card is followed by a card specifying the abscissa, or energy scale. Two numbers, DE and EMIN, are read in FORMAT (2E10.0). If DE > 0. then the energy scale is set up at EMIN and increasing in steps of DE. DE = - 1. reads in the energy scale on the following cards in format specified by the FORMAT card.

Note: This option KERNEL was the first one introduced in the program. To provide further flexibility, the options KREAD and KFIX were introduced later, but KERNEL was left in the program. So, one has the choice between using KERNEL or KREAD plus KFIX -- do not use both!

<u>KFIX</u> Sets up the kernel of energy to be used in LOUHI computations. An energy scale independent of those read in with the cross sections with KREAD card can be used, and the kernel used in computations can be set up by selecting only some of the cross sections read in with the KREAD card. The kernel can be reset at any time without having to read in new cross section curves. WHAT(1-4) specify the energy scale, WHAT(5) the selection of the cross sections, and WHAT(6) the print-in options, as explained below.

WHAT(1) = 0. maintains the energy scale as set by the previous KFIX card,

= N sets the number of energy points to the number N up to 40, as defined in more detail by the parameter KE given by WHAT(2).

WHAT(2) = KE specifies how the N energy points are given, in particular,

KE = 0. maintains the previous energy bins but only changes the index of the last bin to N,

KE = 1. reads N new energy points on the following cards in FORMAT(8E10.0).

KE = 2. sets up the energy scale by starting at WHAT(3) and incrementing linearly by multiplying by WHAT(4).

KE = 3. sets up a logarithmic energy scale by starting at WHAT(3) and incrementing by multiplying by WHAT(4).

WHAT(3) specifies the starting point of the energy scale used with KE = 2 or 3.

WHAT(4) specifies the energy increments used with KE = 2 or 3.

WHAT(5) = 0. maintains the present cross sections in the kernel as set by the previous KFIX card; this option should be used only when changing the number of energy bins used in the computation, that is with WHAT(2) = 0.,

= M specifies the number of cross sections to be included in the kernel, in particular,

M = MC uses all the cross sections as read in with KREAD card

M < MC selects M cross sections as specified by M indeces read on the following cards in FORMAT(8I10); these indeces refer to the order of cross sections read with KREAD card.

WHAT(6) = Prints out values of some internal variables (see subroutine KERNEL). Unless you have to debug some mistakes, leave WHAT(6) = 0. If WHAT(6) =

1. Prints after *** CONTROL CARD KFIX, the values of M, N, and NC.
2. After the above printing, prints the values of N, YVAL(J) that are the energy values set by KFIX.
3. After the above printing, prints the values for the response functions extrapolated for the energy values of KFIX.
4. After the above printing, prints the values of the response function and corresponding energies as set by KREAD.

Note: a. When one changes the energy values from those specified by KREAD, the program calculates the values of the response functions at the new energy values by linear extrapolation from the values read by KREAD. If the energy points set by KFIX go beyond the values set by KREAD, the program assumes a constant value for the response functions (the one at the lowest energy value of KREAD for the energies lower than that and the one at the highest energy value of KREAD for the energies higher than that.

b. Remember to always have a STEPS card when changing the energy values or limits with KFIX.

KREAD Read in cross section data to be used in setting up the kernel by subsequent use of KFIX card.

WHAT(1) = MC is the number of cross sections read in on the following cards.

The first card of each cross section curve gives the parameters KC(J), NC(J), KF, and KR in FORMAT (A10, 3E10.0), where

KC(J) is a alphanumeric title for the Jth cross section,

NC(J) is the number of points in the Jth curve; if NC(J) = 0 is read, then the same number of points as for the previous (J-1)th curve is assumed; up to 40 points,

KF specifies the format for reading the cross sections in particular, KF = 0. maintains the current input format, initialized (8E10.0), while

KF = 1. read a new format on the following card, for instance (2E15.3),

KR specifies the arrangement of the cross section data on the following data cards, in particular,

KR = 0. uses the same energy points as for the previous cross section curve without reading any new energy points,

KR = 1. reads in first the energy point array and then the cross section point array, both in the format as specified by KF, and

KR = 2. reads in alternatively an energy point and a cross section point by using the format as specified by KF.

MAKEPF Sets the approximate solution ϕ_i^0 equal to the solution obtained in the previous calculations (eqs. (6), (9), and (10)).

MINMAX Specifies the scales for Calcomp and cathode-ray-tube plotting, unless specified, these scales are set automatically,

WHAT(1) ≠ 0. is the lower limit of the abscissa scale,

WHAT(2) ≠ 0. is the upper limit of the abscissa scale,

WHAT(3) ≠ 0. is the lower limit of the ordinate scale,

WHAT(4) ≠ 0. is the upper limit of the ordinate scale,

WHAT(5) ≠ 0. is the number of cycles for the logarithmic ordinate scale.

MODE Specifies the mode for the a priori conditions

WHAT(1) = 1. for the linear criteria given in equations (6) and (7).

= 2. for the relative criteria of equations (9) and (11).

= 3. for the logarithmic criteria of equations (10) and (12). This is the initialized value.

NOTIE Releases the ties to the approximate solution ϕ_i^0 by setting W_1 and r_i^p equal to zero. (See equations (6), (9), and (10).)

OLGA Reads in a proton spectrum and sets up the (n,p) scattering kernel automatically, thus no RECOIL and PROTONS cards are needed, WHAT(1) = 0. reads in the cards as prepared for RECOILP program, that is first the correction factors F1 and F2 for the shrinkage, and then the coordinate points, up to a non-zero value in columns 71-72,

= 1. reads in the cards punched by RECOILP program, that is the recoil proton track length distribution divided into 82 bins, in FORMAT(16I5),

WHAT(2) = 0. takes 40 points starting in bin number 11,

= 1. selects the bins optimally.

P Specifies the approximate solution P_i

WHAT(1) = 0. reads in the values P_i, i = 1,2,...,N, in FORMAT (8E10.0),

≠ 0. sets P_i = WHAT(1) $E_i^{\mathrm{WHAT(2)}}$.

PCORRECT Normalizes the approximate solution so that the responses are best matched by a solution of the shape of the P vector,

WHAT(1) } specify the weights for A_j as with ESCALE for the
WHAT(2) } normalization.

PERTUR Perturbs the responses read or those generated by the test solution by normally distributed random errors,

WHAT(1) is the standard deviation of the perturbations, note that WHAT(1) = 0. returns the unperturbed values,

WHAT(2) = 1. specifies the first perturbation,

= 0. for additional perturbations of the same responses, starting with the values stored with WHAT(2) = 1.

Note: This option is used only when performing the internal tests.

PLOT Plots the results on the cathode-ray-tube screen during on-line VISTA runs, the parameters WHAT(I), I=1,2,...,6, and NAME are the same as with CCPLOT.

PROTONS Reads in a measured proton recoil spectrum and adjusts it for the energy scale specified after RECOIL card. This card is followed by cards specifying points of the proton spectrum, as many as set by WHAT(1) on RECOIL card, in FORMAT(8E10.0).

WHAT(1) = 0. sets the uncertainties of the proton spectrum automatically,

$\neq$ 0. reads the uncertainities on the following cards in FORMAT(8E10.0) for all the points in the proton spectrum,

WHAT(2) = 0. uses the energy scale specified after RECOIL card,

$\neq$ 0. reads in the energy scale in FORMAT(8E10.0).

PSCALE Specifies the energy-dependent weighting terms for the approximate solution, r_i^p, in equations (6), (9), and (10).

The scaling is done as with ESCALE, except that the number of points specified by KERNEL or KFIX and STEPS are used and as a weighting term P_i instead of A_j are used. WHAT(1) and WHAT(2) have the same meaning as with ESCALE.

PUNCH Punches the solution vector obtained.

RECOIL Sets up the (n,p) scattering kernel for proton recoil tests and calculations,

WHAT(1) is the number of energy bins in the proton recoil spectrum,

WHAT(2) is the number of points in the energy scale for the neutron spectrum,

WHAT(3) - 0. is the density of hydrogen atoms in the emulsion, initialized 0.06.

The RECOIL card is followed by cards specifying WHAT(1) energy bins in the proton recoil spectrum in FORMAT (8E10.0) and a card specifying DE and EMIN as with KERNEL card. DE>0 and DE = -1. have the same meaning as with KERNEL card. DE = -2. specifies the same energy scale for the proton and neutron spectra, this is the preferred mode. then it is also required that WHAT(1) = WHAT(2).

STEPS Specifies the number of free parameters in the solution,

WHAT(1) specifies that at ever WHAT(1) energy step the amplitude, $\emptyset_i = X_i^2$, is a free parameter, the other $\emptyset_i$'s are obtained through interpolation. In other words, it calculates the solution for every WHAT(1) energy points.

STOP Terminates the computations and end files the disc files used. This has to be the last card in the data deck.

TEST Specifies a test solution and computes the corresponding responses,

WHAT(1) = 0. reads in the test solution on the following cards in format specified by the FORMAT card, number of points fixed by KFIX,

= -1. specifies a test solution of the 1./E form as $\emptyset_i$ = WHAT(2) $E_i^{\mathrm{WHAT(3)}}$,

>0. specifies a test function of the double Gaussian form

$$\emptyset_i = \mathrm{WHAT(1)}\ \exp\left(-\frac{(\log E_i - \log(\mathrm{WHAT(2)}))^2}{2 \cdot \mathrm{WHAT(3)}^2}\right)$$

$$+ \mathrm{WHAT(4)}\ \exp\left(-\frac{(\log E_i - \log(\mathrm{WHAT(5)}))^2}{2\ \mathrm{WHAT(6)}^2}\right).$$

Note: This option is used only when performing the internal tests.

TIE Ties the solution to a point by setting W_1 = 1. and by applying a special weight r_i^p,

WHAT(1)	specifies the number of the energy bin, i,
WHAT(2)	is the corresponding P_i,
WHAT(3)	is the corresponding r_i^p.

TITLE Reads on the following card a title for the run, in FORMAT(8A10)

WEIGHT Specifies the weighting terms W, and W_2 in equation (4),

WHAT(1)	is the term W_1. start = 0
WHAT(2)	is the term W_2. start = 1.

REFERENCES

Ro 69a Routti, J.T., "High-energy neutron spectroscopy with activation detectors incoporating new methods for the analysis of Ge(Li) gamma-ray spectra and the solution of Fredholm integral equations, Ph.D. Thesis, University of California, Lawrence Berkeley Laboratory report UCRL-18514 (1969).

Ro 69b Routti, J.T., "Mathematical considerations of determinig neutron spectra from activition measurements, University of California Lawrence Berkeley Laboratory report UCRL-19375 (1969).

Ro 72 Routti, J.T., "Interactive graphical computing for simulating and unfolding measured distributions", Computer Physics Communications 4, 33 (1972).

Da 59 Davidson, W.C., "Variable metric method for minimization", Argonne National Laboratory report ANL-5990 (1959).

Ri 74 Rindi, A., "An analytical expression for the neutron flux-to-dose conversion factors", Health Physics 27, 3, 322 (1974).

Ri 77 Rindi, A., "An analytical expression for the neutron flux-to-absorbed dose conversion factor. To be published in Health Physics (1977).

LAWRENCE BERKELEY LABORATORY
TECHNICAL INFORMATION DEPARTMENT
1 CYCLOTRON ROAD
BERKELEY, CALIFORNIA 94720