The HEPROW Program System

Manfred Matzke
Physikalisch-Technische Bundesanstalt, D38116 Braunschweig
Present address: Haberweg 5
D3816 Braunschweig

Abstract

The determination of particle energy spectra from integrating measurements is in most cases a so-called ill-posed problem. Computer codes based on Bayes theorem and the maximum entropy method have been developed in the PTB and have been successfully applied in many metrological fields to determine the spectral neutron or photon fluence.

The HEPROW program system is an updated version of the PTB HEPRO system already described in an earlier report. It contains a number of programs for data handling, plotting, folding and unfolding. The codes run in the DOS-Box of a PC and are tested under WINDOWS95 and WINDOWS98.

A CD containing the EXE versions and a demonstration of some examples may be obtained from the author.

The following Fortran computer programs are described:

UMSPHW

------ to transform a file containing a measured pulse-height to a file in HEPRO format. Energy calibration parameters assigning “energy” values to channel numbers are needed.

RSPGW

------ GAUSSIAN broadening of a response matrix, rebinning of a matrix.

OPERAW

------ Program for combination of HEPRO files and multiplication/division addition/subtraction by numbers.

FALTW

------ Program to fold a fluence file with a response matrix and compare measured and calculated pulse-height spectra

GRAVELW

------ Modified SAND-II code[1] for unfolding. (Logarithm of pulse-height spectrum used). Response matrix and a pulse-height spectrum file as input.

UNFANAW

------ Maximum entropy code for unfolding[2;3] The program needs a file called UNFANA.ASC produced before by a GRAVELW run.

MIEKEW

------ Maximum entropy code for unfolding The program needs a file called UNFANA.ASC produced before by a GRAVELW run. The propagation of uncertainties is performed with a so-called ambiguity term [6] .

PLOFW [4]

------ to plot data files or parts of the response function library on the screen, or to create an HPGL file for subsequent plotting[4]. The plot program may be used for a variety of data formats of the files, including the ENDF format.

PLOTAW [4]

------ to plot control data curves from a plot file created while one of the other codes is running.

CONTENTS

1. Introduction

For the evaluation of the spectral particle fluence $\Phi_E(E)$ from integrating measurements the following basic system of linear integral equations has to be solved:[5,6]

$$z_{0i} = \int \mathrm{d}E \; R_i(E)\; \Phi_E(E) \qquad (i = 1,...,M) \qquad (1)$$

Eq. (1) represents the model of the measurement. The vector $\mathbf{z}_0$ is considered as a column vector, transposed it may be written as $\boldsymbol{z}_0^{\mathrm{T}}=(z_{01},\ldots z_{0i},\ldots z_{0M})$, it denotes the (measured) readings of the detector system, more exactly: the expectation values, the real readings being $\mathbf{z'}_0 = \mathbf{z}_0 + \boldsymbol{\varepsilon}$ with the statistically fluctuating quantity $\boldsymbol{\varepsilon}$ ($^{\mathrm{T}}$ meaning transposition). The kernels $R_i(E)$ are the **response functions** for the energy E of the M detector channels of the measuring system.

The solution of eq. (1) has been described by the author before [5,6], some computer programs were developed, which were included into the HEPRO program system of the PTB [7] .

The HEPROW system described here, is an updated version of the HEPRO program system. It contains only the most important codes of HEPRO. Here only a brief description is given, detailed information may be found in reference [7] .

2. NOTATIONS

The description of the notations given in ref. [7] is briefly repeated in the following.

In the general case of determining a neutron or photon fluence by means of pulse height measurements, the measuring device should be capable of giving as a result the spectral fluence at the point of the detector in the detector's absence. This means that all the disturbance of the radiation field produced by the detector must be taken into account in determining the response functions (e.g. inscattering from the housing, flux attenuation etc.).

The reading z_i of channel i of the measured pulse height spectrum in a channel i is represented as a double integral over the energies E_n of the incident particles (neutrons or photons) and the energies E_C deposited in the counter:

$$z_i = \int_{E_{\min}}^{E_{\max}} \mathrm{d}E_n \int_0^{E_{C,\max}} \mathrm{d}E_C \, G(E_C \rightarrow i) \;\; W_{E_C}(E_n \rightarrow E_C) \;\; T_{\Phi}(E_n) \;\; \Phi_E(E_n) \qquad (1)$$

In eq. (1) there is:

$\Phi_E(E_n)$ the spectral particle fluence to be determined, $T_{\Phi}(E_n)$ the fluence response, i.e.

$\Phi_E(E_n) \cdot T_{\Phi}(E_n)$ is the probability density that an incident particle produces the reaction of interest in the counter. The detector is assumed to indicate the incident radiation isotropically.

In the present case of particle reactions, an energy E_C is thought to be created in the counter (energy of the protons and tritons for ^{3}He(n,p) or the energy of the recoil particles for elastic scattering), but E_C might also be any other physical quantity produced in the counter (light, ionization, etc.). In the following the incident particles are often called neutrons, whereas the particles created in the counter and supplying energy to the counter are called protons.

For an ideal counter filled with gas under low pressure, $T_{\Phi}(E_n) = N \, \sigma_{\mathrm{tot}}(E_n)$, where N is the number of possible reacting gas particles and $\sigma_{\mathrm{tot}}(E_n)$ is the total cross section.

$W_{E_C}(E_n \rightarrow E_C)$ is a conditional probability density that by the transfer of energy by the incident particle of energy E_n, the energy produced (the quantity produced) is E_C. There is a normalization condition for $W_{E_C}(E_n \rightarrow E_C)$:

$$\int_0^{E_{C,\max}} \mathrm{d}E_c \;\; W_{E_C}(E_n \rightarrow E_C) = 1 \qquad (2)$$

The function $W_{E_C}(E_n \rightarrow E_C) \cdot T_{\Phi}(E_n)$ is called the **ideal response function:**

$$W_{E_C}(E_n \rightarrow E_C) \cdot T_{\Phi}(E_n) = R_{\mathrm{ideal}}(E_n, E_C) \qquad (3)$$

$G(E_C \rightarrow i)$ is the probability that the quantity E_C produced within the counter is detected in the pulse height spectrum at channel *i*. z_i is the content of channel *i*.

Since $G(E_C \rightarrow i)$ is a probability, it holds that

$$\sum_{i=1}^{M} G(E_C \to i) = 1. \qquad (4)$$

And it follows from eqs. (1) to (4) that:

$$\sum_{i=1}^{M} z_i = \int \mathrm{d}E_n \; T_\Phi(E_n) \; \Phi_E(E_n) \qquad (5)$$

In the following it is assumed that the **ideal response function** of eq. (3), $R_{\mathrm{ideal}}(E_n,E_C)$, is already known, having been previously calculated, e.g. by using the Monte Carlo method [8,9,10,11,12].

For $G(E_C \to i)$ the following integral over a GAUSSIAN function at channel i is assumed :

$$G(E_C \to i) = \int_{E_i}^{E_i+\Delta E_i} \mathrm{d}E \, \frac{1}{\sqrt{2\pi}\sigma(E)} \exp(-(E_C - E)^2 / 2\sigma^2(E) \qquad (6)$$

Here it is required that the pulse height in channel i corresponds to an energy (or another quantity) E_i, for which an energy calibration $E_i = \alpha + \beta \cdot i$ (linear in energy) is assumed to exist.

This requirement is not really a strict one. In principle it is only necessary that the multichannel scale is the same scale as the E_C scale of $W_{E_C}(E_n \to E_C)$.

For the GAUSSIAN resolution function $\sigma(E_C)$ and for the examples considered here, a representation based on physical arguments[13] is used.

Introducing the full width at half maximum by

$F_W(E_C) = 2\sigma(E_C)\sqrt{2\ln(2)}$, it has been shown in reference [10] that a polynomial representation with parameters *a*, *b* and *c* can be used:

$$F_W{}^2(E_C) = a^2 E_C^2 + b^2 E_C + c^2 \qquad (7)$$

With the signal-independent contribution *c*, any electrical noise is taken into account, while contributions proportional to the signal itself (parameter *a*) account for geometry effects such as position-dependent light transmission or amplification. It must carefully be checked whether the latter contribution can be approximated by a GAUSSIAN distribution. Only parameter *b* accounts for the POISSON statistics of the primary ionization in proportional counters or of the photoelectrons produced in the photo tubes [10].

$F_W(E_C)$ can also be obtained by a LAGRANGIAN interpolation formula with three resolution parameters at three energies E_1, E_2, E_3:

$F_W(E_1) = F_{W1}$,

$F_W(E_2) = F_{W2}$,

$F_W(E_3) = F_{W3}$,

$$F_W{}^2(E_c) = F_{W1}^2 \frac{(E_c - E_2)\cdot(E_c - E_3)}{(E_1 - E_2)\cdot(E_1 - E_3)} + F_{W2}^2 \frac{(E_c - E_1)\cdot(E_c - E_3)}{(E_2 - E_1)\cdot(E_2 - E_3)} + F_{W3}^2 \frac{(E_c - E_1)\cdot(E_c - E_2)}{(E_3 - E_1)\cdot(E_3 - E_2)} \tag{8}$$

In practical cases it must be assumed that either the parameter sets *a, b, c* or F_{W1}, F_{W2}, F_{W3} have been determined previously with their uncertainties (standard deviations) before beginning the unfolding.

To discretisize eq. (1) to a form adequate for computer work the following quantities are defined:

$$\Phi_v = \int_{E_v}^{E_v+\Delta E_v} \mathrm{d}E_n\, \Phi_E(E_n) \qquad v = 1,....,N \tag{9}$$

as group fluences in a convenient group representation between $E_1 = E_{\min}$ and $E_N = E_{\max}$ .and $E_{\min} = E_1 \le ...E_v \le ...E_N = E_{\max}$ (10)

The integral over E_n in eq. (1) may be replaced by a sum:

$$\int_{E_{\min}}^{\mathrm{E}_{\max}} \mathrm{d}E_n\, R_{\mathrm{ideal}}(E_n, E_C)\, \Phi_E(E_n) = \sum_{v=1}^{N} \frac{\int_{E_v}^{\mathrm{E}_{v+1}} \mathrm{d}E_n\, R_{\mathrm{ideal}}(E_n, E_C)\, \Phi_E(E_n)}{\int_{E_v}^{\mathrm{E}_{v+1}} \mathrm{d}E_n\, \Phi_E(E_n)} \cdot \Phi_v\,, \tag{11}$$

which is exact per definition. The approximation is performed in the ratio of the right hand side. Assuming that the spectral fluence can be approximated by a constant in the energy interval between E_v and E_{v+1}, eq. (11) may be written as:

$$\int_{E_{\min}}^{\mathrm{E}_{\max}} \mathrm{d}E_n\, R_{\mathrm{ideal}}(E_n, E_C)\, \Phi_E(E_n) = \sum_{v=1}^{N} \frac{\int_{E_v}^{\mathrm{E}_{v+1}} \mathrm{d}E_n\, R_{\mathrm{ideal}}(E_n, E_C)}{E_{v+1} - E_v} \cdot \Phi_v = \sum_{v=1}^{N} R_{\mathrm{ideal},v}(E_C) \cdot \Phi_v \tag{12}$$

where the index *v* at the right hand side of eq. (12) refers to the average of the ideal response function in the interval between E_v and E_{v+1}. In practice there is no difference made between the average value and point values of the response function in the a. m. interval. In the programs used here, $R_{\text{ideal},\nu}(E_C)$ is sometimes also considered as the value of the response function for monoenergetic incident particles at the medium value of energy in the interval between E_ν and $E_{\nu+1}$.

Approximating in eq. (1) the integration over E_C in a similar way:

$$\int_0^{E_{C\max}} \mathrm{d}E_C\, G(E_C \to i)\, R_{\text{ideal},\nu}(E_C) = \sum_{k=1}^{k_{\max}} G(E_{C,k} \to i) \int_{\mathrm{E_{C,k}}}^{\mathrm{E_{C,k+1}}} \mathrm{d}E_C\, R_{\text{ideal},\nu}(E_C)\,, \tag{13}$$

where it is assumed that the function $G(E_C \to i)$ in the interval between $E_{C,k}$ and $E_{C,k+1}$, can be replaced by the value at $E_{C,k}$. Writing

$$\int_{\mathrm{E_{C,k}}}^{\mathrm{E_{C,k+1}}} \mathrm{d}E_C\, R_{\text{ideal},\nu}(E_C) = R_{\text{ideal},k\nu} \qquad , \tag{14}$$

and using eq. (6), the following equation follows from eq. 13):

$$\int_0^{E_{C\max}} \mathrm{d}E_C\, G(E_C \to i)\, R_{\text{ideal},\nu}(E_C) = \sum_{k=1}^{k_{\max}} \mathrm{G}_{\mathrm{ik}}\, R_{\text{ideal},k\nu}$$

where $R_{\text{ideal,k}\nu}$ is now considered as averaged over the energy interval between $E_{C,k}$ and $E_{C,k+1}$ (so-called proton energies) as well as over the energy interval between E_{v} and $E_{\mathrm{v+1}}$ (neutron or photon energies), and where the GAUSSIAN within this interval is approximated by

$$G_{ik} = \frac{E_{i+1} - E_i}{\sqrt{2\pi}\sigma(E_{C,k})} \exp(-(E_{C,k} - E_i)^2 \big/ 2\,\sigma(E_{C,k})^2) \tag{15}$$

$E_{C\max}$ is determined by the kinematics of the nuclear reaction considered, for instance for ^{3}He(n,p) there is $E_{C\max} = E_n + Q$ (where the Q-value is 764 keV) and for the proton recoil there is $E_{C\max} = E_n$ (these energy values may be influenced by the *W*-values of the counting gas).

Eq. (1) can now be written as:

$$z_i = \sum_{k=1}^{k_{\max}} G_{ik} \sum_{\nu=1}^{N} R_{\text{ideal,k}\nu}\, \Phi_\nu \qquad i = 1,...,K \tag{16}$$

or in vector (matrix) notation

$$\mathbf{z} = \mathbf{G} \cdot \mathbf{R}_{\text{ideal}} \cdot \mathbf{\Phi} \tag{17}$$

In eq. (16) the matrix product $\mathbf{G}\cdot\mathbf{R}_{\text{ideal}}$ is called the **response matrix R:**

$$\mathbf{R}=\mathbf{G}\cdot\mathbf{R}_{\text{ideal}}\,. \tag{18}$$

The "model equations" for **z** used for unfolding can therefore be written as:

$$\mathbf{z}=\mathbf{R}\cdot\mathbf{\Phi} \tag{19}$$

A direct solution of eq. (19) would obviously be a very simple way of unfolding. Provided the matrix **R** is quadratic and has maximum rank the solution of eq. (19) is

$$\mathbf{\Phi}=\mathbf{R}^{-1}\cdot\mathbf{z} \tag{20}$$

This method is not investigated in the HEPRO programs. Here, least-squares methods and methods based generally on the maximum entropy principle are preferred.

3. Structure of Data Files and Libraries

3.1 Fluence Files, Single Response Functions [4], General Data Files

All files containing functions (i.e. tabulated x-y-values) within the HEPRO package or in the programs described have the same structure. The fluence files, the multichannel files, the single response functions and the various fluence-to-dose conversion functions consist of a number *KG* of triples $(E(i), F(i), \sigma(i), i=1,...,KG)$, where *E(i)* is the energy, *F(i)* the value of the function considered and $\sigma(i)$ its standard deviation. The energy scale is thought to be in MeV, but some of the programs considered are able to convert this scale (see below).

The structure of a data file is as follows:

Line 1: arbitrary text characterizing the set,
FORMAT 80A1 (80 or less characters)

Line 2: *MODE, MEV* FORMAT *, i.e. format-free, *MODE* = 0:

Point values follow, i.e. *F(i) = F(E(i))*,

the **function** is taken at energy values *E(i)*

MODE = 1:

Average values of *F(i)* and $\sigma(i)$ in the interval $E(i) \le E < E(i+1)$ follow, with

$$F(i) = \frac{\int_{E(i)}^{E(i+1)} F_E(E)dE}{E(i+1) - E(i)}$$, which means that **group-averaged** values follow.

MODE = 2:

Group values of *F(i)* and $\sigma(i)$ follow (e.g. multichannel values or MCNP [14] result),

with $F(i) = \int_{E(i)}^{E(i+1)} F_E(E)dE$, which means that group integrals follow.

MODE = 3:

Group values of *F(i)* and $\sigma(i)$ per “unit lethargy” follow.

For the interpretation of these values it is assumed that $F(i) = \int_{E(i)}^{E(i+1)} F_E(E)dE$ has been calculated or measured, and that $\dfrac{F(i)}{log(E(i+1)/E(i))}$ is contained in the file, which is sometimes called “fluence per unit lethargy”. The interpretation of this terminus is sometimes misleading. Sometimes it is assumed that “lethargy representation” means that the expression $\dfrac{F(i) \cdot \overline{E}}{E(i+1) - E(i)}$ is contained in the file, with either $\overline{E} = \sqrt{E(i+1) \cdot E(i)}$ (logarithmic average of energy) or $\overline{E} = \frac{1}{2}(E(i+1) + E(i))$ (linear average of energy). These expressions are only approximately equal,

$$\frac{1}{log(E(2)/E(1))} \approx \frac{\overline{E}}{E(2) - E(1)}.$$

The difficulties in the interpretation of “lethargy units” can often be seen in graphical representations where in most cases the values $\dfrac{F(i)}{log(E(i+1)/E(i))}$ are plotted in the form of a histogram with horizontal bars in the intervals in a log-x and lin-y representation but the abscissa is denoted as *E* and the ordinate as *F(E)*E* .

MEV = 0: energies *E(i)* in eV

MEV = 1: energies *E(i)* in MeV (recommended for HEPRO and RESTRAW)

MEV = 2: energies *E(i)* in keV.

It should be mentioned that some of the programs of the HEPRO package are able to use an additional parameter *IUN* in this second line with:

IUN=0 --------- no uncertainties in the following columns

IUN=1---------- standard uncertainties follow,

IUN=2---------- standard variances follow

IUN=2---------- relative standard uncertainties (in %) follow.

In the present version not all the codes understand the *IUN* parameter, therefore it is recommended to use only standard deviations.

Line 3:

IL, *IH*,*KG*, *E(KG*+1*)* FORMAT *, i.e. format-free

IL=1, *IH=KG*, are used here as fixed values not yet explicitly needed by the codes. These parameters may be used in later versions of the programs for additional information.

KG is the number of points or groups which follow.

E(KG+1) is the last (right) energy boundary.

The programs need a (dummy) value, even in the case of point values (*MODE*=0).

Line 4 to line *KG*+3: *E*(*i*)*, F*(*i*)*,* $\sigma(i)$ FORMAT*, i.e. format-free

The meaning of the values of *E*(*i*) depend on the *MODE* parameter; for *MODE*=0 the following *E*(*i*), *F*(*i*) values are point values and *KG* values are needed; for *MODE* > 0 the *E(i)* are **left** boundaries of the interval ranging from *E(i*) to *E*(*i*+1) and *KG*+1 values of *E*(*i*) are needed, the last right boundary is found in line 3.

Note: In MCNP [14] data files the right boundaries of energy intervals are used. The values of *F(i)* have been explained above. $\sigma(i)$ is the standard deviation of *F*(*i*)*,* (for Poisson statistics sometimes taken as $\sigma(i) = \sqrt{F(i)}$).

If $\sigma(i)$ is not available, a dummy column should be included since some of the HEPRO programs crash when this column is lacking.

Note: For the plot program PLOF or PLOFW, other data representations are possible (see description of the PLOT program).
Note: For input pulse-height spectra, MODE=2 is required in HEPRO.

3.2 Response Matrices

The format of the response functions for “few channel” unfolding has been described in reference [4], here, the description is given of the format of files containing multichannel response functions, i.e. corresponding to pulse-height spectra.

Considering an incident particle of energy E_n, its response function in the general case can be considered as a multichannel spectrum with K channels. (see section 1). In general the channel scaling (not the bin width) of a response matrix should agree with that of a measured pulse height spectrum. This means that to be able to perform the unfolding of a pulse height spectrum a calibration of the measured pulse height spectrum is needed which assigns to each channel a certain physical quantity (e.g. a proton energy or an electric current or a quantity proportional to a light output). The same physical quantity is used for scaling the response function for an incident particle of a certain energy E_n (see eq. (1): The E_C scale of the response matrix must equal the E_i scale of the pulse-height spectrum). In the program codes and in the following description the incident particles are termed "neutrons" and the E_C scale is termed "proton energy". Constant bin-width (equidistant energy intervals) is required for all the channels in the response matrix file and the bin-width is termed *EKA*.
This means that channel number *k* has the boundaries *EKA**(*k*-1) at the left and *EKA*k* at the right. In the same units, e.g. MeV, the GAUSSIAN parameters (see section 1) are expected by the programs and the measured pulse height spectrum must be calibrated in the same units but may have a bin width different from the response matrix. The structure of a response matrix file for $R_{\mathrm{ideal},k\nu}$ (ideal response matrix , see eq. (14)) or $R_{i\nu}$ (real response matrix, see eq. (18)) is as follows:

The energy scale E_n of the incident particles may be arbitrarily binned. the energies E_n are considered as parameters of single response functions.

The response matrix file has formatted (ASCII) structure:

1st record: *EKA* FORMAT *, i.e. format-free

2nd record: E_n, *IH*, *E*(1), *E*(IH+1) FORMAT *, i.e. format-free

where the incident particle energy E_n is a parameter, *IH* is the number of channels of the response function for this E_n , i.e. the number of following response function values, *E*(1) and *E*(*IH*+1) are the first (left) boundary and the last (right) boundary, where *E*(1)=0 (in all cases) and *E*(*IH*+1)=*EKA***IH*.

3rd record:

Response function for E_n , i.e. *IH* values of R_{kv} for *k*=1,…,*IH* (format-free)

R_{kv} is thought to represent the average value of the response function in the interval between E_ν and $E_{\nu+1}$, or one monoenergetic response function value assigned to the medium value of E_n in the interval between E_ν and $E_{\nu+1}$.

next record: new values for the next neutron energy E_{n+1} , i.e. new values for E_{n+1} , *IH*, *E*(1), *E*(*IH*+1)

next record: R_{kv+1} (*k*=1, ..., *IH*)

This 2-record structure is repeated for all neutron energies.
The neutron energies should be arranged in ascending order, and they may have an arbitrary energy structure.

4. The Programs

All the programs of the HEPROW package are running in the DOS box and can be used with an input file as command-line argument, e.g. typing

```
GRAVELW Gravelw.inp
```

means that the GRAVELW program is started and imports the input data from the file gravelw.inp. When the name of the input file is not given in the command-line, the program asks for terminal input (interactive input).

In cases where the structure of the input file is not quite clear to the user, it is recommended to use the programs without an input file as command-line argument. A lot of additional information is then displayed at the screen.

All the programs write a protocol file, the name of which is read from the first line of the corresponding input file. There is no strict rule for the names of the files, the author prefers to use the extension *.LOG for the protocol files and the extension *.FLU for the fluence files.

Nearly all codes produce a file with graphic data, the name of which equals the protocol file but has the extension *.PLO. The graphics can be displayed after a program run by using the PLOTAW program (just typing PLOTAW xxxxx.PLO).

The extension *.MTX is used for the covariance matrix of fluence in the MIEKEW program, the extension *.PCR is used for the covariance matrix of a pulse-height spectrum created by the FALTW program.

4.1 The UMSPHW Program

The UMSPHW program transforms a measured pulse height spectrum in an arbitrary format into the structure of the HEPRO package.

The structure of the pulse height spectrum is expected as follows:

A certain number of lines to be skipped during reading followed by a set of format-free given pulse-height values (separator between the values: blank, or comma, or semicolon). It is assumed that poisson statistics hold, i.e. the standard uncertainty assigned to the contents in one channel is defined as the square root of the number of counts. It is assumed that no correlations between different channels exist.

For the output file an energy has to be assigned to each channel, this energy (or some other quantity, which is the same quantity (scaling) as that used for the response function) is thought to be related to the left side boundary of a channel.

The assignment is performed by *EL*(*I*)=(*I*-1-*XI0*)**EKA*-*Q*, where

EL(*I*) is the left hand energy boundary of channel No. *I*.

I is the channel number,

EKA is the channel width, i.e. channel No. *I* is ranging from *EL*(*I*) to *EL*(*I*+1)=*EL*(*I*)+*EKA*

XI0 is the channel "number" (real value) which is related to an energy of 0.0

Q=0.0 is normally used. For ^{3}He in the PTB files there is a value *Q*=0.764 used for better comparison between neutron and proton scale.

EXAMPLES: (with *Q*=0.0)

1.) If the zero value of energy is in the first channel at 0.66, *XI0*=0.66 has to be used, and the first channel reaches from *EL*(1)= - 0.66**EKA* to *EL*(1)+*EKA*=-0.66**EKA*+*EKA* = 0.34**EKA* .

2.) If the zero value of energy is in the 3rd channel at 2.44, *XI0*=2.44 has to be used, and the 3rd channel reaches from (2-2.44)**EKA* = -0.44**EKA* to -0.44**EKA*+*EKA*=0.56**EKA*

When using no input file as command line argument, the “file” CON is used as default, that means the user is prompted by the program to type the input data.

Example an input file
(comments in the input file should be written right of column 40, they are ignored during reading)

```
UMSPHDF.log                      name of the protocol file
BACKGROUND                       comment to charaterize the task
BACK.PH                          name of input pulse height file
backout.phs                      name of output file (HEPRO-format)
5.33                             XI0, channel No. related to energy zero
1.2e-3                           EKA, channel width in MeV
0.0                              Q-value which is Q=0.764 for 3-He in PTB
0.0                              Tcorr, dead time correction factor
1.0                              TIME, counting time (1.0 for fluence)
1.0                              RMON, monitor reading for division
12                               UNS, number of pulses in each channel
1                                ISKIP, number of lines to be skipped
1024                             NUMB, number of channels to be read
```

Comments:

UNS is the number of pulses to be added to the uncertainty in each channel, e.g. a global background uncertainty.

NOTE: the standard uncertainty $\sigma(i)$ in channel *i* is the square-root of the sum of a value of *NUMBER*(*i*)+*UNS*, where *NUMBER*(*i*) is the number of registered counts in channel *i*).

A number of ISKIP header lines in the pulse-height spectrum file can be over-read before reading the channel contents.
TIME is the counting time, *TIME*=1.0 should be used for later calculating fluence.

The dead time correction factor *Tcorr* is $\tau / TIME$ where τ is the dead time. For *Tcorr*=0.0 there is no deadtime correction. Dead time correction is performed in channel *i* by $NUMBER(i)_{\text{true}} = \dfrac{NUMBER(i)_{\text{measured}}}{1.0 - Tcorr * NUMBER(i)_{\text{measured}}}$, where

$NUMBER(i)_{\text{measured}}$ is the measured number of counts in a channel *i*.
The readings $NUMBER(i)_{\text{measured}}$ are divided by *RMON***TIME*.

The *NUMBER*(*i*) values are read FORMAT-FREE:
READ(15,*) (*NUMBER*(*i*),*i*=1,*NUMB*).

As mentioned above, uncertainties are calculated with the assumption of poisson statistics (Square roots).
NOTE: for *NUMBER*(*i*)=0 the uncertainty is set to 1

The output file has the structure explained in section 2 with *MODE* = 2 (group values in each channel).

NOTE: It is not required that the pulse height spectrum and the response matrix have the same bin-width *EKA*, as the programs perform a rebinning, if necessary. Only the energy scaling mentioned above must agree.

4.2 The OPERAW Program

Program for combination of two HEPRO data files and multiplication/division by numbers.
For the combination of two files, both files must have the same energy structure, but may differ in the *MODE* parameter (see section 2). The files are converted to the *MODE* parameter of the first file.
The program may be used too for multiplication or division of the contents of one data file .

Example of input files:

(comments in the input file should be written right of column 40, they are ignored during reading)

<u>Example 1</u> : 1st file is multiplied by a number, second file divided by another number, The y-values of both files are multiplied:

```
operaw.LOG                              protocol file
ambe.122                                first hepro file
*5.5                                    first to be multiplied by 5.5
icrp-60                                 second hepro file
/2.0                                    to be divided by 2.0
operadf.flu                             resulting file
*                                       both single results are multiplied
```

<u>Example 2</u> : First file is multiplied by a number, no second file used.

```
operaw.LOG                              protocol file
ambe.122                                first hepro file
*5.5                                    first to be multiplied by 5.5
                                        blank line (no second file)
operadf.flu                             resulting file
```

Comment: The following symbols for operation are possible: + or - or * or /

For doing nothing with the single files the value 1.0 should be used together with the corresponding operation symbol.

If $\otimes$ denotes the single operation symbols, and VALUE characterizes a factor or a sum etc., the result of a OPERAW run is always:

$(\text{FILE1} \otimes \text{VALUE1}) \otimes (\text{FILE2} \otimes \text{VALUE2})$

4.3 The RSPGW Program

The RSPGW program has the following purposes:

--- to perform a rebinning of a response matrix to a new bin structure for "protons" or "neutrons", where the incident particles are termed "neutrons", and it is assumed that the secondary particles termed "protons" correspond to the measured pulse-height spectrum.

--- to create a monoenergetic response matrix, sometimes needed for checking results, the response matrix created has the form:

$$R_{i\nu} = \delta_{i\nu} \tag{21}$$

--- to create a GAUSSIAN response matrix

$$R_{i\nu} = G_{i\nu} = \frac{\Delta E_i}{\sqrt{2\pi}\sigma(E_\nu)} exp(-(E_\nu - E_i)^2 / 2\sigma(E_\nu)^2) \tag{22}$$

--- to perform GAUSSIAN broadening of a response matrix $R_{\mathrm{ideal},i\nu}$ (see section 2):

$$R_{i\nu} = \sum_k \frac{\Delta E_i}{\sqrt{2\pi}\sigma(E_k)} exp(-(E_k - E_i)^2 / 2\sigma(E_k)^2) \quad \cdot \; R_{\mathrm{ideal,k\nu}} \tag{23}$$

The GAUSSIAN width σ is a polynomial of the second degree according to section 2. The structures of the input files are different for the different tasks. There is a more detailed description in reference [7]. The user may construct an input file interactively by running the program first without using an input file i.e. without a file name as command-line argument.

Examples for input files:

Example 1: The ideal response function contained in file ORIGINAL.RSG shall be transformed to a Gaussian broadened response function in file RSG_1.RSP, the bin structure shall be kept, Gaussian parameters *a, b, c* shall be used.

(comments in the input file should be written right of column 40, they are ignored during reading)

```
rsg_1.log                             printer (protocol) file
0.0                                   q-value (see HEPROW report)
rsg_1.rsp                             name of output resp. functions
1                                     gauss-folding wanted
original.rsp                          input (ideal resp. functions)
1                                     neutron energies from resp. file
1                                     proton energies from resp. file
0.1,10.0                              wanted neutron energy range (MeV)
-1.0,4.0e-2                           parameter a  (see HEPRO report)
```

```
1.2e-1                                parameter b  (see HEPRO report)
5.0e-3                                parameter c  (see HEPRO report)
0                                     no asymmetry correction
norm_1                                name of file for output norm
0                                     no multiplication of resp with factor
```

Note: The negative sign in line 9 is for input of parameters *a, b, c*, the file NORM_1 contajns the integral response (efficiency).

Example 2: The ideal response function contained in file ORIGINAL.RSG shall be transformed to a Gaussian broadened response function in file RSG_2.RSP, the bin structure shall be kept, not the Gaussian parameters a, b, c but the FWHM at 3 energies shall be used.

(comments in the input file should be written right of column 40, they are ignored during reading)

```
rsg_2.log                             printer (protocol) file
0.0                                   q-value (see HEPRO report)
rsg_2.rsp                             name of output resp. functions
1                                     gauss-folding wanted
original.rsp                          input (ideal resp. functions)
1                                     neutron energies from resp. file
1                                     proton energies from resp. file
0.1,10.0                              wanted neutron energy range (MeV)
1.0,126.6                             energy E1, FWHM(E1)
4.0,288.5                             energy E2, FWHM(E2)
8.0,469.1                             energy E3, FWHM(E3)
0                                     no asymmetry correction
norm_2                                name of file for output norm
0                                     no multiplication of resp with factor
```

Example 3: A rebinning of the response matrix RSG_1:RSG shall be performed (details are described in reference [7]).
Note: only the first 40 characters in each line are read from the input file

```
rsg_3.log                        printer (protocol)file
0.0                              q-value (see HEPRO report)
rsg_3.rsp                        name of output resp. functions
0                                no folding, only transformation of binning
rsg_1.rsp                        input resp. functions
0                                neutron energies not from response file
0                                proton energies not from response file
0.12,9.9                         wanted neutron energy range (MeV)
0                                ISC (binning not proportional to FWHM)
2                                group structure from input file follows:
490,1                            number of energy points, linear binned
5.0e-3                           EKA (new bin width multichannel part)
0                                no asymmetry correction of response
norm_3                           name of file for output norm
0                                no multiplication of resp with factor
```

4.4 The FALTW Program

This program performs a folding of a fluence spectrum file with a response matrix according to

$$z_i = \sum_{v} R_{iv} \cdot \Phi_v \quad \text{with } i = 1,\ldots,M \qquad (24)$$

where z_i represents the number of the counts in the calculated pulse-height spectrum, which may be compared with the measured pulse-height spectrum z_{i0} .If there is a measured pulse-height spectrum available together with the fluence spectrum, the FALTW program also calculates a difference pulse-height spectrum.

The structures of the input files are different for the different tasks. There is a more detailed description in reference [7]. The user may construct an input file interactively by running the program first without using an input file i.e. without a file name as command-line argument.

Examples for input files:

Example 1: A pulse-height spectrum OPR_1.PHS has been measured, .a fluence file MIK_1b.FLU has been obtained from the MIEKEW unfolding code together with a covariance matrix MIEKE.MTX. A new pulse-height spectrum FALTW_1.PHS shall be evaluated using fluence, with covariance matrix and the reponse matrix RSG_1.RSP. The difference spectrum between OPR_1.PHS and FALTW_1.PHS shall be calculated.

(comments in the input file should be written right of column 40, they are ignored during reading)

```
faltw_1.log                         name of protocol file
0.0                                 q-value
1                                   measured multichannel file is available
0.12,7.0                            considered energy range multich. file
opr_1.phs                           name of measured multichannel file
4.00,7.8,0                          considered neutron energy range, mede
..\resp\rsg_1.rsp                   name of response matrix file
1                                   one range for energy calib. correction
1.0,1.0                             energy, calibration factor
mik_1b.flu                          name of fluence file
2                                   uncertainties from covariance matrix
mieke.mtx                           covariance matrix of the fluence
faltw_1.flu                         flux output file
faltw_1.phs                         pulse height output
faltw_1.phd                         difference file
```

Comment:

There is a small correction of the energy calibration in certain energy ranges possible. This correction is performed by a factor called VER according to:

E(new) = (E(old)+Q)*VER-Q.

Here, no energy correction is performed, since the correction factor 1.0 is chosen.

Example 2: From a known fluence spectrum without known covariance matrix, a theoretical pulse-height spectrum shall be calculated. The channel contents in the new pulse-height spectrum shall be chosen in a way that an overall number of 10^6 counts is registered.

(comments in the input file should be written right of column 40, they are ignored during reading)

```
faltw.log                       protocol file
0.0                             q-value
0                               no measured multichannel file available
0.0,5.0                         'energies' in multichannel scale
1.6056519e-2                    channel width in new file in 'energy' units
0.3,4.5,0                       photon (neutron) energy range
..\resp\rsg_1.rsp               response matrix
1                               1 range for calibration correction
1.0,1.0                         energy, correction factor for calibr.
Flux                            name of fluence file
0                               no uncertainties from fluence file
faltw.flu                       output fluence
faltw.phs                       output pulse height spectrum
1.0e6                           overall number of registered pulses
```

Comment:

For the uncertainty propagation the following relations are used within the program:

From the fluence vector $\boldsymbol{\Phi}$ with covariance matrix $\mathbf{S}_\Phi$ the standard uncertainty matrix $\mathbf{S}_z$ of the calculated pulse-height spectrum may be calculated from:

$\mathbf{S}_z = \mathbf{R}^\mathrm{T} \cdot \mathbf{S}_\Phi \cdot \mathbf{R}$. When for the unfolding (evaluation of $\boldsymbol{\Phi}$) only the upper part of the measured pulse-height spectrum was used, the lower channels of the measured pulse-height spectrum are not correlated with the evaluated fluence values. Then, for the difference pulse-height spectrum

$$\mathbf{z}_d = \mathbf{z}_0 - \mathbf{R} \cdot \boldsymbol{\Phi} \qquad (25)$$

for the lower channels the following uncertainty matrix is obtained:

$$\mathbf{S}_{zd} = \mathbf{S}_{z0} + \mathbf{R}^{\mathrm{T}} \cdot \mathbf{S}_{\Phi} \cdot \mathbf{R} \tag{26}$$

where the uncertainty matrix of the measured values, $\mathbf{S}_{z0}$ is diagonal. For the case of calculating a difference pulse-height spectrum with exact uncertainty propagation including the fluence uncertainties, the resulting covariance matrix file of the difference spectrum with the extension *.PCR is automatically created by the FALTW program. This file is later used by the GRAVELW program (see section 3.5)

4.5 The GRAVELW Program

4.5.1 Formulas

To introduce the condition of positive fluence into the unfolding procedure, an algorithm similar to that introduced in SAND-II [1] can be applied. In this case not the group fluence values Φ_v but its logarithms $\ln(\Phi_v)$ are determined by a special iteration procedure. The original SAND-II code was designed for half life determination and later used for multiple foil reactor dosimetry. Uncertainties of the readings could not be taken into account considered, a weighting proportional to the relative variances of the readings was introduced by the author into the algorithm [7] . A new code called GRAVEL (now GRAVELW) was written and successfully applied in “few channel” as well as in “multichannel” unfolding [5,6] .

In the χ^2 expression considered, the logarithms of the pulse height spectrum values $\mathbf{z}_0$ (see eq. (1)) are used with a diagonal (relative) covariance matrix:

$$\chi^2 = \sum_i (\ln z_{0i} - \ln z_i)^2 \cdot \frac{1}{\rho_i^2} \tag{27}$$

with (see eq. (1))

$$z_i = \sum_v R_{iv} \cdot \Phi_v = \sum_v R_{iv} \cdot \exp(\ln \Phi_v) \tag{28}$$

and the ρ_i are the relative standard deviations of the z_{0i}, namely $\rho_i = \frac{\sigma_{0i}}{z_{0i}}$. In the original SAND-II code the "weights" ρ_i were not included.

Assuming that there is a solution, $\ln \Phi_v^{(1)}$, already known in the vicinity of the exact solution, $\ln z_i$ in eq. (28) can be expanded into a TAYLOR series truncating after the second term:

$$\ln z_i = \ln z_i^{(1)} + \sum_v w_{iv}^{(1)} \cdot (\ln \Phi_v - \ln \Phi_v^{(1)}) \tag{29}$$

with

$$z_i^{(1)} = \sum_v R_{iv} \cdot \exp(\ln \Phi_v^{(1)}) \quad \text{and} \tag{30}$$

$$w_{iv}^{(1)} = \frac{R_{iv} \cdot \exp(\ln \Phi_v^{(1)})}{z_i^{(1)}} \tag{31}$$

To find the minimum of the χ^2 expression eq. (27) an iteration procedure can be constructed [5]. In each iteration step the current solution (k+1) is obtained from the previous solution (k) by

$$\ln \Phi_\mu^{(k+1)} - \ln \Phi_\mu^{(k)} = \lambda_\mu^{(k)} \cdot \sum_i (\ln z_{0i} - \ln z_i^{(k)}) \cdot \frac{w_{i\mu}^{(k)}}{\rho_i^2}, \quad \text{where} \qquad \lambda_\mu^{(k)} = \left(\sum_i \frac{w_{i\mu}^{(k)}}{\rho_i^2} \right)^{-1} \tag{32}$$

It should be mentioned that the BUNKI (SPUNIT) algorithm[15] is similar to the SAND-II (Gravel) algorithm. Starting from the SAND-II iteration formula eq. (32) and developing the right hand side around $z_i^k = z_{i0} + \delta_i^k$ one obtains in first order:

$$\Phi_\mu^{(k+1)} = \Phi_\mu^{(k)} \frac{\sum_i \frac{z_{0i}}{z_i^k} \cdot \frac{R_{i\mu}}{z_{0i}} \cdot \frac{1}{\rho_i^2}}{\sum_i \frac{R_{i\mu}}{z_{0i}} \cdot \frac{1}{\rho_i^2}}$$

which equals the algorithm used in the SPUNIT code [16]. It has been found that the results obtained by the two codes are nearly identical.

For the iteration procedure in the SAND-II (GRAVEL) and SPUNIT code, a first input spectrum is needed when the iteration is started. A solution always exists , but the solution spectrum depends on this input spectrum in a way which is not quite transparent so that an uncertainty propagation cannot be easily performed [6] .

4.5.2 Running the GRAVELW Program

The GRAVELW program needs an input file, where all relevant data are listed.

The structures of the input files are different for the different tasks. There is a more detailed description in reference [7]. The user may construct an input file interactively by running the program first without using an input file i.e. without a file name as command-line argument. In this case some additional information is displayed.

Example of an input file:

(comments in the input file should be written right of column 40, they are ignored during reading)

```
grv_1.log                         printer (protocol) file
0.0,0                             q-value (see HEPRO report), ISPUNIT
opr_1.phs                         multichannel input file (HEPRO format)
3.50,7.0                         “energy” range considered in opr_1.phs
1.0                               energy correction for multi-ch.
-0.05,7.8,0                       neutron energy range,linear medium v.
..\resp\rsg_1.rsp                 response matrix input file
0                                 no fluence file available,use const
0.5,10000,200,1                   eps,msand,idr,modpl
grv_1.flu                         output fluence file
grv_1.phs                         output calculated multi-ch. file
grv_1.pds                         output calculated-difference file
n                                 no scaling of stand. deviation wanted
j                                 output in file UNFANA.asc wanted
```

In the new version the SPUNIT mode[15] is possible, indicated by the second number in line 2. ISPUNIT=1 means calculation in SPUNIT mode (see above), ISPUNIT=0 means calculation in SAND-II mode. The program reads the data of the pulse-height spectrum, line 3, within a range defined by line 4. It opens the response matrix file, line 7, and reads the relevant data defined in line 4 and line 6.

In line 6 the energy range for the neutrons is required. If these numbers are negative, the energy range is calculated automatically. The numbers then are taken as fractions of the response function. For the value of 0.05 (see above), the program selects the left neutron energy that 5% of the response function range is left from the lowest pulse-height channel. A value of 0.95 for the upper energy value means that 95% are left of the upper boundary of the pulse-height spectrum. Negative values make only sense for nearly rectangular response matrices.

The program takes the “neutron” energies from the response matrix in the range considered and assigns a neutron group to every point value E_n read from the matrix. If the third number in line 6 is zero, the response matrix energy E_n in the interval with index v is interpreted as the linear middle value of the group, $E_n = 0.5 * (E_v + E_{v+1})$ for a value of 1 the logarithmic representation $E_n = \sqrt{E_v * E_{v+1})}$ is used. In every run the energy structure at the “neutron” side is determined by the “point” energies in the response matrix.

In line 5 an energy correction factor VER may be given, for a correction of the energies of the pulse-height spectrum according to :
E(new) = (E(old)+Q)*VER-Q.
In the example above, no energy correction is performed, since the correction factor 1.0 is chosen.
The values EPS, MSAND cut the iteration procedure at either a limit $\chi^2 \leq EPS$ or after MSAND operations. Each IDRth iteration results are written to the protocol file.

In each GRAVELW run a file called UNFANA.ASC is produced, which contains the relevant data needed by the programs MIEKEW and UNFANAW

4.6 The MIEKEW Program

4.6.1 Formulas

The following text is nearly identical with parts of references [6, 7]
The existing models for solving the inverse problem can be treated from two points of view.
In the first method the minimum least-squares solution according to

$$\chi^2 = (\mathbf{z}_0 - \mathbf{R} \cdot \mathbf{\Phi})^{\mathrm{T}} \cdot \mathbf{S}_{z0}^{-1} \cdot (\mathbf{z}_0 - \mathbf{R} \cdot \mathbf{\Phi}) \;\rightarrow\; \text{Min} \qquad (33)$$

is determined, where $\mathbf{S}_{z0}$ is the uncertainty matrix or covariance matrix of the measured values $\mathbf{z}_0$. A solution of eq. (33), in principle, should always exists for realistic physical problems (e. g. when the condition of non-negative fluence values $\mathbf{\Phi}$ is used in the minimisation process) but in many cases the solution may not be unique.
In the second method the ambiguity is taken into account and a probability distribution of the possible fluence components is determined. Both ways can be traced back to the Bayesian theory together with the principle of Maximum Entropy [2], they describe two ways of inference.
The Bayesian method is used to provide a formal foundation for the use of *a priori* information. The *a priori* information is not limited to data values and uncertainties but can include correlations and various physical constraints. Sometimes the only *a priori* information might be that all the fluence values should not be negative. Starting from

an *a priori* probabiltiy distribution (called Prior) the posterior probability (called posterior) of Bayes theorem can be obtained from the well-known relation [17] [18] " posterior proportional to likelihood times prior".

Generally, to establish the joint posterior probability distribution $P(\mathbf{x})$ of a set of variables $\mathbf{x}$, the distribution $P(\mathbf{x})$ is factorized by

$$P(\mathbf{x}) = P_L(\mathbf{x}) \cdot w(\mathbf{x}) \tag{34}$$

into the unknown likelihood $P_L(\mathbf{x})$ and the prior $w(\mathbf{x})$. The distribution $P(\mathbf{x})$ is then obtained by the Lagrange method using the principle of maximum entropy

$$S \;=\; -\int P(\mathbf{x}) \log(\frac{P(\mathbf{x})}{w(\mathbf{x})})\, \mathrm{d}\mathbf{x} \;\Rightarrow\; \max \tag{35}$$

and observing the normalization condition and certain constraints on $\mathbf{x}$ [17]. The likelihood too, may be the result of maximizing the entropy where as a prior the so-called data prior is used. Most of the unfolding models applied use as the likelihood the probability distribution of the fluence vector:

$$P_L(\mathbf{\Phi}) = \quad C_1 \cdot \exp(-\frac{\beta}{2} \cdot \chi^2(\mathbf{\Phi})) \tag{36}$$

with a normalization constant C_1, with $\chi^2(\mathbf{\Phi})$ defined by eq. (33), and where the parameter β is determined (see below) from the condition that the expectation value of $\chi^2(\mathbf{\Phi})$ in the posterior probability distribution (with the notation $<\chi^2(\mathbf{\Phi})>$) equals the number of degrees of freedom M. Obviously, the minimum least-squares method is equivalent to maximizing the likelihood but obviously the two ways mentioned are based on different models concerning the value of $\chi^2(\mathbf{\Phi})$ ($\chi^2 \rightarrow \min.$ or $<\chi^2> = M$). However, it can be shown that both ways come nearly to the same result in the cases where the minimum least-square solution exists and the uncertainties of the readings are small.

Minimum least-squares method is often used to fit data, where it is assumed that a model applied is true. This is the first level of inference [19]. At the second level of inference it is intended to infer which model is most plausible given the data, which means discussing probabilitity distributions and expectation values.

The properties of the minimum least-squares codes are less satisfactory, when there is a lack of pre-information on the particle spectrum. In this case, the singularity or quasi-singularity of the matrix $\mathbf{B}$ [5, 6, 7] which results from the minimisation process of eq. (33), namely

$$\mathbf{B} = \mathbf{R}^{\mathrm{T}} \cdot \mathbf{S}_{z0}^{-1} \cdot \mathbf{R}\,, \tag{37}$$

may lead to an ambiguity of the solutions, even if the constraint of non-negative fluence is included into the least-squares algorithm. For the equivalent Maximum Likelihood method it follows that a unique, most probable particle spectrum might not exist. However, similar to the maximum likelihood method, it is possible to construct a *posteriori* probability density of the fluence using Bayes' theorem with eq. (34) [2,3]. The solution spectrum can then be defined as the expectation value of $\mathbf{\Phi}$, denoted by $< \mathbf{\Phi} >$, which can be calculated from the probability density. This method was used in reference [20]; later, a more modern interpretation was given in references [5, 6, 2] using the principle of maximum entropy.

In fluence spectrum unfolding, together with eq. (35) certain constraints usually are formulated and used with the maximum entropy method [17, 18]. When the constraining conditions to be imposed are included into eq. (35), it is possible to formulate the extremum principle for the determination of the fluence probability density $P(\mathbf{\Phi})$ using as the prior the constant probability distribution ("equal distribution of ignorance" [18]).

For the case realized in the MIEKE code, two constraints are assumed to exist:

(a) all group fluence values should be non-negative, and

(b) the expectation value $< \chi^2 >$ of χ^2 (see eq. (33)), should be equal to the number of degrees of freedom involved, which is equal to the number M of elements in $\mathbf{z}_0$.

By means of the extremalisation procedure, the probability density

$$P(\mathbf{\Phi}) = \quad C_1 \cdot \exp(-\frac{\beta}{2} \cdot \chi^2(\mathbf{\Phi})) \quad \text{for all} \quad \Phi_\nu \geq 0 \qquad (38)$$

is obtained, the prior of eq. (34) is taken as a constant value, defined for all $\Phi_\nu \geq 0$ and vanishing for negative values due to condition (a). C_1 is the normalization constant. β is a "temperature" parameter to be determined from condition (b). The probability density $P(\mathbf{\Phi})$ represents a multivariate normal distribution with a χ^2-exponent, truncated because of condition (a). Although the exponent is degenerate in the case $N > M$ (or for ill-conditioned matrix $\mathbf{B}$), the distribution can be normalized and the expectation value $< \mathbf{\Phi} >$ with its uncertainty matrix

$$\mathbf{S}_\Phi = < \mathbf{\Phi}\ \mathbf{\Phi} > - < \mathbf{\Phi} > < \mathbf{\Phi} > \qquad (39)$$

can be calculated. For β = 0.5 and excluding the constraint of non-negative fluence values, $P(\mathbf{\Phi})$ is equivalent to the likelihood expression used in BAYES' theorem if a normal distribution is assumed for the measured data.

The probability density of eq. (38) is introduced into the MIEKE code of the HEPRO package [7] and is (of course) also implemented in the MIEKEW program. A Monte Carlo code with importance sampling is used to calculate expectation values.

In thermodynamic particle statistics a well-known algorithm exists for the solution of N-dimensional integrals: the importance sampling random walk.

The idea is to approximate the expectation value by an average over a set of $\mathbf{\Phi}$ vectors taken from near the point of maximum probability. A complete description of this method is given by Wood (see also reference [7]) therefore only the essential features will be indicated here. The method consists in performing a random walk in the N-dimensional space spanned by the N axes of $\mathbf{\Phi}$. Each point in this space is equivalent to a fluence vector $\mathbf{\Phi}$. The random walk is started at an arbitrary point in this space and then certain MARKOV-chain transition probabilities are used to produce a successive set of fluence vectors. The links in this chain are denoted by $\mathbf{\Phi}^{(n)}$. For the construction of such transition probabilities the special form of the distribution $P(\mathbf{\Phi})$ of eq. (38) is taken into account, so that fluence vectors $\mathbf{\Phi}^{(n)}$ are avoided where $P(\mathbf{\Phi}^{(n)})$ is rather small.

In practice one starts with a vector $\mathbf{\Phi}^{(0)}$, which for reasons of computing time should originate from a region of high probability. At each step (n) the value of $\chi^2(\mathbf{\Phi}^{(n)})$ must be calculated. Then a trial state (n+1) (new fluence vector $\mathbf{\Phi}^{(n+1)}$ is chosen randomly and uniformly (i.e. with equal probability for any state (n+1)) from the possible states in the neighbourhood of $\mathbf{\Phi}^{(n)}$. In the procedure most preferred for the selection of the neighbouring state, one randomly chosen component (channel) Φ_μ^n of $\mathbf{\Phi}^{(n)}$ is changed randomly with a maximum permissable displacement δ:

$$\Phi_\nu^{(n+1)} = \Phi_\nu^{(n)} \quad \text{for} \quad \nu \neq \mu \tag{40}$$

$$\Phi_\nu^{(n+1)} = \Phi_\nu^{(n)} + \delta \quad \text{for} \quad \nu = \mu$$

The probability densities $P(\mathbf{\Phi}^{(n+1)})$ and $P(\mathbf{\Phi}^{(n)})$ are then compared.

If $P(\mathbf{\Phi}^{(n+1)}) > P(\mathbf{\Phi}^{(n)})$ the trial state is accepted as a realization of the Markov chain.

In the other case, i.e. if $r = \dfrac{P(\mathbf{\Phi}^{(n+1)})}{P(\mathbf{\Phi}^{(n)})} \leq 1$, a new random number ξ between 0 and 1 is compared with *r*. For $\xi \leq r$ the trial state is also accepted, while for $\xi > r$ it is rejected and the old state (*n*) is taken as a new realization (*n*+1).

In thermodynamic particle physics there are several additional methods used for accelerating the Markov process [21] In the present case of unfolding pulse height spectra the method described above seems to be the fastest.

It can be shown in general [21] that by means of this random walk, expectation values of a function $g(\mathbf{\Phi})$ can be calculated as averages in the Markov chain.

$$\left\langle g(\mathbf{\Phi})\right\rangle = \frac{1}{T}\sum_{n=1}^{T} g(\mathbf{\Phi}^{(n)}) \tag{41}$$

where (*n*) indicates the Markov chain realization, and *T* is the number of trials. In addition the variance of *g* can be evaluated by means of

$$\sigma_g^2 = \left\langle g^2\right\rangle - \left\langle g\right\rangle^2 \tag{42}$$

It can be shown that in the limit $T \to \infty$ these mean values converge asymptotically against the corresponding expectation values.

In practice the Monte Carlo bias of the mean values may be obtained from the results of independently repeated Markov chains. The parameter *ß* must be chosen so that

$< \chi^2(\mathbf{\Phi}) > = M$ (number of degrees of freedom). (43)

For practical cases, in a first run *ß* is chosen as a value valid for a GAUSSIAN distribution *ß*=1, and is adjusted in later runs [7].

Some important statements have to be made concerning uncertainty propagation:

In spectra unfolding work, where in most cases the matrix **B** of eq. (37) is of deficient rank [5, 6, 2], in principle, two fluctuation components exist, one resulting from the uncertainties of the input quantities $\mathbf{x}_0$ (readings, response functions, *a priori* information on the fluence), one resulting from the ambiguity of the solution[6] (compare, for example with the microscopic and macroscopic fluctuations in particle statistics). The fluctuation of the resulting fluence can be considered as being composed of both contributions:

$$\delta\mathbf{\Phi} = < \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}_0} > \delta \mathbf{x}_0 + \delta \mathbf{\Phi}_1 \tag{44}$$

perpendicular to each other, i. e. $< \delta\mathbf{x}_0\, \delta\mathbf{\Phi}_1 > = \mathbf{0}$. The resulting covariance matrix is thus the sum of two components: $\frac{\partial < \mathbf{\Phi} >}{\partial \mathbf{x}_0}\mathbf{S}_{x0}\frac{\partial < \mathbf{\Phi} >}{\partial \mathbf{x}_0} + \mathbf{S}_{\Phi 1}$. For the MIEKE algorithm, the corresponding formulas are quite simple to obtain from the probability distribution. Considering only the uncertainties $\mathbf{S}_{z0}$ of $\mathbf{z}_0$, it can be shown that $\frac{\partial < \mathbf{\Phi} >}{\partial \mathbf{z}_0} = -\beta\, \mathbf{S}_\Phi \cdot \mathbf{R} \cdot \mathbf{S}_{z0}^{-1}$, and with the definition of **B** (eq. (37)) one finds

$$\mathbf{S}_{\Phi 1} = \mathbf{S}_\Phi - \beta^2 \mathbf{S}_\Phi \cdot \mathbf{B} \cdot \mathbf{S}_\Phi \,. \qquad (45)$$

In linear least-squares methods $\beta^2 \mathbf{S}_\Phi = \mathbf{B}^{-1}$, which holds for non-singular **B** and for a gaussian distribution. In this case, the solution is unique and $\mathbf{S}_{\Phi 1}$ vanishes. The ambiguity part $\mathbf{S}_{\Phi 1}$ cannot be obtained so easily for the other maximum entropy models. For UNFANA [3] and MAXED [22] , only the conventional uncertainty propagation corresponding to $\mathbf{z}_0$ and (in principle to $\mathbf{R}_0$) is included. It has been found that the part from the uncertainty propagation of $\mathbf{z}_0$ for the MIEKE code ($\beta^2 \mathbf{S}_\Phi \cdot \mathbf{B} \cdot \mathbf{S}_\Phi$) agrees well with the corresponding result of UNFANA.

It must be mentioned that sometimes, when a rather large temperature parameter has to be used, the fluence uncertainty due to uncertainty propagation $\beta^2 \mathbf{S}_\Phi \cdot \mathbf{B} \cdot \mathbf{S}_\Phi$ turns out to be greater than the part $\mathbf{S}_\Phi$ where both contributions (including ambiguity) are combined. This makes no sense and possibly shows the limits in practical unfolding work: a high temperature parameter restricts the possible ambiguity.

A short comment should be made on the uncertainty propagation for the uncertainty of response functions:

For the MIEKE algorithm one obtains in obvious matrix notation (indices have to be chosen carefully):

$$\frac{\partial < \mathbf{\Phi} >}{\partial \mathbf{R}} = \beta \left\{ \mathbf{S}_\Phi \cdot \mathbf{S}_{z0}^{-1} \cdot \mathbf{z}_0 - \left(< \mathbf{\Phi\Phi\Phi} > - < \mathbf{\Phi} >< \mathbf{\Phi\Phi} > \right) \cdot \mathbf{R} \cdot \mathbf{S}_{z0}^{-1} \right\}. \qquad (46)$$

As is usual in uncertainty propagation, the triple correlation function can be approximated in gaussian approximation:

$$< (\mathbf{\Phi}_1 - < \mathbf{\Phi}_1 >)(\mathbf{\Phi}_2 - < \mathbf{\Phi}_2 >)\,(\mathbf{\Phi}_3 - < \mathbf{\Phi}_3 >) > = \mathbf{0}\,. \qquad (47)$$

This means that the usual propagation of uncertainties even for the response matrix, can be compiled after an unfolding run, provided the resulting covariance matrix $\mathbf{S}_\Phi$ given by eq. (39) is compiled by the code.

It should be mentioned that the maximum entropy algorithms allow other probability distributions to be constructed [23]. In the MIEKE code the evaluation of *P*(*H**) for the dose-equivalent $H^* = \sum_{v} h_{\Phi v} \Phi_v = \mathbf{h}_\Phi \cdot \mathbf{\Phi}$ can be performed ($\mathbf{h}_\Phi$: fluence-to dose conversion factor):

$$P(H^*) = <\delta(H^* - \mathbf{h}_\Phi \cdot \mathbf{\Phi})> \qquad (48)$$

The variance $<H^* H^*> - <H^*><H^*> = \mathbf{h}_\Phi \; \mathbf{S}_\Phi \; \mathbf{h}_\Phi$ of the distribution again includes contributions from the uncertainty propagation and the ambiguity.

The ambiguity part the dose equivalent uncertainty may thus be represented by :

$$\sigma_{\text{ambiguity},H} = \sqrt{\mathbf{h}_\Phi (\mathbf{S}_\Phi - \beta^2 \mathbf{S}_\Phi \cdot \mathbf{B} \cdot \mathbf{S}_\Phi) \mathbf{h}_\Phi} \; , \qquad (49)$$

Instead of the dose equivalent probability distribution the corresponding probability distribution of other integrals like a certain detector reading *z* can be calculated. From $z = \sum_{v} R_v \; \Phi_v = \mathbf{R} \cdot \mathbf{\Phi}$ the probability distribution

$$P(z) = <\delta(z - \mathbf{R} \cdot \mathbf{\Phi})> \qquad (50)$$

may be obtained, and the uncertainty may be calculated in the same manner as for the dose equivalent.

4.6.2 Running the MIEKEW Program

The MIEKEW program needs an input file, where all the relevant data are listed.
The structures of the input files are different for the different tasks. There is a more detailed description in reference [7]. The user may construct an input file interactively by running the program first without using an input file i.e. without a file name as command-line argument. In this case some additional information is displayed.

Note: For every MIEKEW run the file UNFANA.ASC is needed. This file must be created before, e.g. by a GRAVELW run, i.e. the MIEKEW program is thought to succeed a GRAVELW run.

Example of an input file:

(comments in the input file should be written right of column 40, they are ignored during reading)

```
mik.log                          name of Protocol (printer) file
mik.flu                          name of the new unfolded fluence file
5000000,10,4,0.50                trials, loops, thermalizations, delt
5.0,0                            temperature parameter,no covariance matrix
6                                number of additional group integrals
1.0                              right boundary of 1. energy interval
2.0                              right boundary of 2. energy interval
5.0                              right boundary of 4. energy interval
6.4                              right boundary of 5. energy interval
7.05                             right boundary of 6. energy interval
7.7                              right boundary of 7. energy interval
mik.gru                          name of the file for group integrals
1                                1 additional file for integrating
..\HEPROW\ed-ohg\hicru.sv        name including path
HICRU                            short name used here
2,0.01,10.0,4                    interpolation law, boundaries, plot mode
```

Comments and hints:

In the third line of the input file the number of trials and the number of the loops is required. In the example given above, 10 loops each with 5000000 trials are performed. As usual in Monte Carlo statistics a Monte Carlo bias may be estimated by assuming that the results within a single loop are independent from the results of the other loops. The number of loops is followed by the number of thermalizations at the beginning of the calculation. A number of 4 is found to be adequate. The delta-parameter (see eq. (40)) should be chosen that (see protocol file) 40% to 60% of the trials are accepted. It is recommended to check the expectation value of $<\chi^2>$ after each run, and eventually to increase or to decrease the temperature parameter in the fourth line to obtain an expectation values which is nearly 1.

If instead of 0 (second number in line 4) the number 1 is given, the program calculates the entire covariance matrix, which needs sometimes hours of computing time.

In line 5 a number of energy intervals can be given (followed by the energy boundaries of the intervals). The program calculates fluence integrals und their uncertainties for these intervals. Results are stored in a file with HEPRO format, the name of which is in the line which follows.

The following number is used for additional folding integrals. In the example given above, the folding for 1 additional file (hicru.sv) is performed (see also protocol file).

The last line contains (for this file) the interpolation law, the energy boundaries to be considered and the plotting mode[4].

For the interpolation code the following numbers may be used:

1 : *Y* is CONSTANT IN *X*	$Y = \mathrm{const}$
2 : *Y* is LINEAR IN *X*	$Y = a\,X+b$
3 : *Y* is LINEAR IN LOG(*X*)	$Y = a \log(X)+b$
4 : LOG(*Y*) is LINEAR IN *X*	$\log(Y) = a\,X+b$
5 : LOG(*Y*) LINEAR IN LOG(*X*)	$\log(Y) = a \log(X)+b$

Other interpolation codes are not possible. The interpolation mode is needed for function (point) values of the data, for group averaged data the parameter is ignored.

Extraploation mode 0 means cutting the function at the energy boundaries, a value of 1 means linear extrapolation.

For the plotting mode see reference[4] and compare with the LAY parameter of the PLOFW program, a value 888 means: no plot.

4.7 The UNFANAW Program

4.7.1 Formulas

For the MIEKE code, the computing time turned out to be very long, in particular for the case that the number of fluence groups is large compared with the number of measurements ($M<<N$). K. Weise[3] therefore proposed an analytical approach to the Monte Carlo results to reduce the computing time. Weise replaced the distribution of eq. (38) by the simpler exponential **ansatz**[3]:

$$P(\mathbf{\Phi}) = C_1 \cdot \exp(-\mathbf{b}^{\mathrm{T}} \cdot \mathbf{R} \cdot \mathbf{\Phi}) \qquad (51)$$

and used the same constraints for the extremum principle of maximum entropy as for the MIEKEW calculations. The parameters $\mathbf{b}$ for maximum entropy can be obtained by a simple solution of a non-linear matrix equation. Here, the expectation values can easily be calculated. The covariance matrix $\mathbf{S}_{\Phi}$ of the distribution turns out to be diagonal and must be properly interpreted. To obtain the uncertainty associated with the expectation value $<\mathbf{\Phi}>$ due to the uncertainty matrix $\mathbf{S}_{z0}$, the gaussian law of uncertainty propagation is used. K. Weise is the author of the subroutine UNFANA[3] which is part of the UNFANAW program. The subroutine UNFANA uses eq. (51) and performs the uncertainty analysis.

Weise has shown that for $N \rightarrow \infty$ the expectation values $< \mathbf{\Phi} >$ obtained are identical for both distributions of eqs. (38) and (51). It has been found from a variety of examples that there is good agreement for $M \approx N$, too.

4.7.2 Running the UNFANAW Program

The UNFANAW program is similar in structure as the MIEKEW program. The input files are nearly identical with the exception of the lines 3 and 4 for the MIEKEW input, where Monte Carlo parameters are given.

The user may construct an input file interactively by running the program first without using an input file i.e. without a file name as command-line argument. In this case some additional information is displayed.

Note: For every UNFANAW run the file UNFANA.ASC is needed (The same holds for a MIEKEW run, as described above). This file must be created before, e.g. by a GRAVELW run, i.e. the UNFANAW program is thought to succeed a GRAVELW run.

Example of an input file:

(comments in the input file should be written right of column 40, they are ignored during reading)

```
unf_1.log                         name of Protocol (printer) file
unf_1.flu                         name of the new unfolded fluence file
4                                 number of additional group integrals
5.0                               right boundary of 1. energy interval
6.4                               right boundary of 2. energy interval
7.05                              right boundary of 3. energy interval
7.7                               right boundary of 4. energy interval
unf_1.gru                         name of the file for group integrals
1                                 1 additional file for integrating
..\HEPROW\ed-ohg\hicru.sv         name including path
HICRU                             short name used here
2,0.01,10.0,4                     interpolation law, boundaries, plot mode
1.0
```

Comments and hints:

In line 5 a number of energy intervals can be given (followed by the energy boundaries of the intervals). The program calculates fluence integrals und their uncertainties for these intervals. Results are stored in a file with HEPRO format, the name of which is in the line which follows.

The following number is used for additional folding integrals. In the example given above, the folding for 1 additional file (hicru.sv) is performed (see also protocol file). The last but one line contains (for this file) the interpolation law, the energy boundaries to be considered and the plotting mode [4]. In the last line the expectation value for $<\chi^2>$ is quoted, which for the resulting fluence values should hold. Sometimes, due to obviously wrong uncertainties estimates of the pulse-height spectrum, a value of $<\chi^2>=1$ cannot be reached. In such cases a greater value may be used.

For the interpolation code the following numbers may be used:

1 : *Y* is CONSTANT IN *X*	*Y* = const
2 : *Y* is LINEAR IN *X*	*Y = a X+b*
3 : *Y* is LINEAR IN LOG(*X*)	*Y = a* log*(X)+b*
4 : LOG(*Y*) is LINEAR IN *X*	log*(Y) = a X+b*
5 : LOG(*Y*) LINEAR IN LOG(*X*)	log*(Y) = a* log*(X)+b*

Other interpolation codes are not possible. The interpolation mode is needed for function (point) values of the data, for group averaged data the parameter is ignored.

By the program the uncertainty matrix of the fluence (uncertainty propagation containing the uncertainties of the readings) is evaluated and stored in a file with the extension *.FCV.

5. The PLOFW Program for plotting Files

In the HEPRO package [7] two plot codes are included: PLOF and PLOTA. These programs were originally compiled for running in the DOS box. In the present version two Quick-WIN compilations named PLOFW and PLOTAW are available. In every run of the programs a plot file is created, which can be displayed at the screen by using the PLOTAW program. A description and an example are given further below.

The PLOFW program at the beginning just asks for the name (including the complete path) of the file to be plotted:

```
PROGRAM PLOF BY MANFRED MATZKE
 PLOTTING SPECTRUM FILES OR CROSS SECTION FILES OF SPECIAL
 FORMAT. >>>>>>>>VERSION JULY 25, 2000<<<<<<<<
 YOU ARE PROMPTED BY THE PROGRAM TO TYPE THE PARAMETERS
 *********************************************************
 typing # leads to a DOS window
 leave this screen by closing it (typing EXIT)
 *********************************************************
 give now either #
           or the name of the file containing the data
```

The next question is for the plotting mode, requiring *IFEE* and *LAY*:

```
 ifee=0 : F(E)   is plotted
 ifee=1 : F(E)*E is plotted
          (F(E)*E means function per unit lethargy)
 lay-parameter for axis scaling:
 lay=1: x-linear,      y-linear
 lay=2: x-logarithm,   y-linear
 lay=3: x-linear,      y-logarithm
 lay=4: x-logarithm,   y-logarithm
 give ifee, lay
----------------------------------------------
```

The values of the parameters have to be chosen according to the list above.

The next parameters *iun, fakt* relate to uncertainties or are used for scaling:

```
---------------------------------------------------
 iun=0       : no uncertainties to plot
 if iun = 1  : standard deviations are expected in the file
 if iun = 2  : variances are expected in the file
 if iun = 3  : relative stand. deviations are expected in the file
 if iun = 300: relative stand. deviations are expected in the file
               given in %
 if iun =-1  : the square roots of the y-values are taken
 fakt     : all y-values are multiplied by fakt
 fakt=0.0 : normalization of this curve to max. of the first
            curve. (for the first curve use fakt=1.0 !!)
 fakt should normally be chosen at fakt=1.0
 give iun,fakt
```

As mentioned above, for *IUN*, a value of 1 should normally be used, however, PLOFW also understands the other parameter values.

A lot of different formats including the ENDF format can be handled by the PLOFW program. It is recommended to try the ENDF format; the user is prompted to MAT, MT values of an ENDF file. A parameter *IFMT* is used to distinguish the formats.

```
ifmt=1: special file format of the MIEKE and HEPRO packages
     =2: multichannel file, i. e. a few comment lines,
         (you are prompted to the number to skip)
         followed by lines of x,y, or x,y,sig values,
         sig-values are read in if iun > 0
    =22: multichannel file, i. e. a few comment lines,
         (you are prompted to the number to skip)
```

```
        followed by lines of y, or y, sig values,
        sig-values are read in if iun > 0
  =222: standard multichannel file (lines with ! in 1st column ignored)
   =3 : endf format, you are prompted to mat, mt values
   =4 : special format (KNAUF,PTB), you are prompted
   =5 : DOSCROSS (SAND-II) FORMATTED OR UNFORMATTED CROSS
        SECTION LIBRARY of the IAEA NMF package
        you are prompted to the name of a reaction
   =55: SAND-II new format (e.g.RESTRANS-LIBRARY)
   =6 : format of the GRAF-FILES of the PTB
        (can also be used for plotting up to 15 columns)
   =7 : special format of response functions (HEPRO package)
   =8 : Techplot-files x,y values between BEGIN and END
   =9 : MCNP, 3 comment lines, group fluence values, unit MeV
**************** give ifmt*************
```

After this input a screen is shown containing the desired plot. There are a number of possibilities of changing the outfit. The user should type either **CH** to change the outfit, (this input is identical with pressing ENTER), or **HP** to output the plot as a file in the HPGL language. The user is asked for the name of the HPGL file. **END** is to be used to finish the program, **OK** to continue with another file.

After **CH** has been typed or ENTER has been pressed the following screen appears:

```
PLOF-PLOTA VERSION JULY 25, 2000: change switches:
  1 : lay   (lin-log scaling of the axes)
  3 : modpl (plotting mode: POINTS or LINE or HISTOGRAM
      or special symbol, 38 symbols are available)
  4 : fakt  (multiply x-y-values by factor+offset)
      use fakty=0.0 to scale to maximum values
  5 : text  (new text for curve characterization)
  6 : com   (new comment line for all plots)
  7 : xmin,xmax (x-axis: new values for next plot)
  8 : ymin,ymax (y-axis: new values for next plot)
  9 : iun   (change plotting uncertainties)
 10 : ifee  (change plotting mode: y(X) or y(X)*X)
      ifee=1 means Y(X)*X  i.e. fluence per unit lethargy
 11 : xtext (new text on x-axis)
 12 : ytext (new text on y-axis)
 13 : integral over a certain x-range, or normalize curves
 14 : change LINETYP, PENWIDTH, PATTERN length for LASERJET
 15 : change colour or tick-length of the graph for plotting
 16 : DOS SCREEN (goto DOS level) leave this screen by EXIT
 17 : change background colour from white to black or from black to white
 18 : save as bitmap file (e.g. for later use under Windows)
 ---> give one number between 1 and 18
 ++ or press ENTER to see the graph again
```

After typing of one of the numbers, a second window is opened explaining what to do. As in the present version the LASERJET output works only with HP LASERJET 4, the user should choose the HP-Graphic language output and may edit this file, for instance, before transferring the graphic to a text program

such as WINWORD. In the QUICK-WIN application PLOFW, the user may mark graphic or text areas on the screen for copying.
The user should try the 38 special symbols available by changing MODPL (No. 3).

The PLOF or PLOFW program can be used with the first 4 parameters as command-line arguments. For instance, the call
PLOFW XXXXX 1 2 0 1.0 1
means a run of PLOFW with
file name=XXXXX
IFEE=1
LAY=2
IUN=0
FAKT=1.0
IFMT=1

The plot programs require a file called FARBDAT and search this file in the current directory or in the directory c:\fonts\ or d:\fonts\. The user may look at the contents of this file for editing and changing default color and pen width parameters.
It is recommended to place the .EXE versions of the PLOTAW and PLOFW plot programs in one of the automatically investigated (global) path directories.

References

[1] W.N. McElroy, S. Berg, T. Crockett and R.G. Hawkins, "A Computer-Automated Iterative Method for Neutron Flux Spectra Determination by Foil Activation", Report AFWL-TR-67-41, U.S. Air Force Weapons Laboratory, 1967.

[2] K. Weise and M. Matzke, "A priori distributions from the principle of maximum entropy for the Monte Carlo unfolding of particle energy spectra", Nucl. Instr. and Methods A280 (1989) 103-112.

[3] K. Weise, "Mathematical Foundation of an Analytical Approach to Bayesian-Statistical Monte Carlo Spectrum Unfolding", Report PTB-N-24, Physikalisch-Technische Bundesanstalt, Braunschweig, 1995.

[4] M. Matzke, “Computational Work with Response Functions: The RESTRAW Program System“, to be published as a PTB laboratory report.

[5] M. Matzke, "Unfolding of Particle Spectra", International Conference: Neutrons in Research and Industry, Crete, Greece, 1996, George Vourvopoulos, Editor, Proc. SPIE 2867 (1997), 598-607, ISBN 0-8194-2263-0

[6] M. Matzke, "Propagation of Uncertainties in Unfolding Procedures", Nucl. Instr. and Methods A476 (2002) 230-241

[7] M. Matzke, "Unfolding of Pulse Height Spectra: The HEPRO Program System", Report PTB-N-19, Physikalisch-Technische Bundesanstalt, Braunschweig, 1994.

[8] Weyrauch, M. , "The Reaction ^{3}He(n,p)T: Kinematics and Cross Sections", Braunschweig: Physikalisch-Technische Bundesanstalt, Report PTB-7.51-91-1, (1991)

[9] Weise, K., Weyrauch, M., Knauf, K., "Neutron Response of a Spherical Proton Recoil Proportional Counter", Nucl. Instrum. Methods A309 (1991) 287-293

[10] Dietze, G., Klein, H., " NRESP4 and NEFF4: Monte Carlo Codes for the Calculation of Neutron Response Functions and Detector Efficiencies for NE213 Scintillation Detectors", Report PTB-ND-22, PTB Braunschweig, 1982

[11] Dietz, E., Matzke, M., Sosaat, W., Urbach, G., and Weyrauch, M., "Neutron Response of an ^{3}He proportional counter," Nucl. Instr. Methods A232 (1993) 521-528

[12] Bürmann, L., Ding, S., Guldbakke, S., Klein, H., Novotny, T., Tichy, M., "Response of NE213 Liquid Scintillation Detectors to High-energy Photons ($E_\gamma > 3$ MeV)", Nucl. Instr. Methods A332(1993) 483-492

[13] Dietze, G., Klein, "Gamma Calibration of NE213 Scintillation Counters", Nucl. Instrum. Methods, 193(1982) 549-556

[14] J. F. Briesmeister (Ed.), “MCNP- A General Monte Carlo N-Particle Transport Code,” Version 4A, LA-12625-M, Los Alamos, 1993.

[15] K. A. Lowry and T. L. Johnson, "Modification to Iterative Recursion Unfolding Algorithms and Computer codes to Find More Appropriate Neutron Spectra", NRL Report 5340 Naval Research Laboratory, Washington D. C. 1984

[16] J. J. Doroshenko, S. N. Kraitor, T. V. Kuznetsova, K. K. Kushnereva, E. S. Leonov, “ New Methods for Measuring Neutron Spectra with Energy from 0.4 eV to 10 MeV by Track and Activation Detectors”, Nucl. Technol. 33 (1977) 296-304

[17] K. Weise and W. Wöger, " A Bayesian theory of measurement uncertainty", Meas. Sci. Technol. **4**, (1993) 1-11, A more detailed version is available in German: " Eine Bayessche Theorie der Messunsicherheit", PTB-Report N-11, Braunschweig, Physikalisch-Technische Bundesanstalt 1992.

[18] H. Jeffreys, "Theory of Probability", Clarendon Press, Oxford, 1967

[19] D. J. C. MacKay, "Bayesian neural networks and density networks", Nucl. Instr. and Meth. A354, (1995) 73-80

[20] M. Matzke and K. Weise, "Neutron Spectrum Unfolding by the Monte Carlo Method", Nucl. Instr. and Methods A234 (1985) 324-330

[21] Wood, W. W. in: "Physics of Simple Liquids", ed. H. N. V. Temperley, North Holland, Amsterdam, New York, (1968) p.115

[22] M. Reginatto, P. Goldhagen, "A Computer Code for the Deconvolution of Multisphere Neutron Spectrometer Data using the Maximum Entropy Method", Report EML-595, Environmental Measurements Laboratory, New York 1998

[23] M. Matzke, "Estimation of Dose Equivalent from Reaction Rates of Bonner Spheres without Using a priori Fluence Information", Rad. Prot. Dosim. 23 (1988) 297-300.