Risø–R–1652-update

PALS*fit*: A computer program for analysing positron lifetime spectra

Peter Kirkegaard, Jens V Olsen,
Morten Eldrup, and Niels Jørgen Pedersen.

Final update 30-10-2014

For citation refer to official February 2009 report see
http://palsfit.dk

DTU Risø Campus, Technical University of Denmark
October 2014

Abstract A Windows based computer program PALS*fit* has been developed for analysing positron lifetime spectra. The program is built upon the well tested PATFIT package. The present document describes the mathematical foundation of the PALS*fit* model as well as a number of features of the program. The cornerstones in PALS*fit* are two least-squares fitting modules: *PositronFit* extracts lifetimes and intensities from the measured lifetime spectra. *ResolutionFit* determines the time resolution function from lifetime spectra which have been recorded for this purpose. PALS*fit* also contains a Monte-Carlo spectrum simulator. The PALS*fit* software can be obtained from DTU Risø Campus (www.palsfit.dk).
e-mail address: palsfit@risoe.dtu.dk

ISBN XXX-XX-XXX-XXXX-X (Internet)
ISSN XXXX-XXXX

2014

Contents

1 Introduction and overview *5*

2 About PALS*fit* *6*
2.1 General fitting criterion *6*
2.2 PALS*fit* input *7*
2.3 PALS*fit* output in general *7*
2.4 The *PositronFit* model *8*
2.5 The Main Output from *PositronFit* *10*
2.6 The *ResolutionFit* model *12*
2.7 The Main Output from *ResolutionFit* *12*

3 Experience with PALS*fit* *14*
3.1 *PositronFit* experience *14*
3.2 *ResolutionFit* experience *15*

4 Least-squares fit and statistics *17*
4.1 Unconstrained NLLS data fitting *17*
4.2 Constraints *19*
4.3 Statistical analysis *20*
4.4 Statistical interpretations *22*
4.5 Marquardt minimization *24*
4.6 Separable least-squares technique *26*
4.7 Various mathematics, statistics, and numerics *27*

5 Mathematical model details *35*
5.1 *PositronFit* *35*
5.2 *ResolutionFit* *37*

6 Control files *37*
6.1 *PositronFit* control file *38*
6.2 *ResolutionFit* control file *41*
6.3 Channel ranges in PALS*fit* *42*

7 Simulation of lifetime spectra *43*

References *44*

1 Introduction and overview

An important aspect of doing experiments by positron annihilation lifetime spectroscopy (PALS) is carrying out an analysis of the measured spectra to extract physically meaningful parameters. A number of computer programs have been developed over the last many years by various authors for this purpose. At our laboratory we have concentrated on developing programs for least squares fitting of such spectra.

PALS*fit* is our newest program of this kind. It is based on the well tested PATFIT software [1, 2], which has been used extensively by the positron annihilation community. A brief, preliminary description of the PALS*fit* program was given in [3]. Taking advantage of a new Graphical User Interface (GUI) for Windows, a number of user friendly facilities have been incorporated, in particular graphics displays.

The two cornerstones in PALS*fit* are the following least-squares fitting modules:

- *PositronFit* extracts lifetimes and intensities from lifetime spectra.
- *ResolutionFit* determines the lifetime spectrometer time resolution function to be used in *PositronFit* analyses.

Correspondingly PALS*fit* may run in either of two modes, producing a *PositronFit* analysis or a *ResolutionFit* analysis, respectively.

Common for both modules is that a model function will be fitted to a measured spectrum. The model function consists of a sum of decaying exponentials convoluted with a time resolution function, plus a constant background. The time resolution function is described by a sum of Gaussians which may be displaced with respect to each other. Various types of constraints may be imposed on the fitting parameters.

An example of a PALS*fit* spectrum window is shown in Fig. 1 on p. 6.

In *ResolutionFit*, parameters determining the shape of the resolution function can be fitted, normally by analysing lifetime spectra which contain mainly one component. The extracted resolution curve may then be used in *PositronFit* to analyse more complicated spectra. PALS*fit* provides facilities for accomplishing this process.

Note that in *PositronFit* the shape of the resolution function is fixed. A correction for positrons annihilating outside the sample can be made during the *PositronFit* analysis [1].

In Chapter 2 we give a presentation of *PositronFit* and *ResolutionFit*. This includes a short overview of the mathematical models as well as output examples. In Chapter 3 we convey some experiences we and others have gained with PALS*fit* and its predecessors. Chapters 4 and 5 contain mathematical and statistical details, while Chapter 6 displays examples of so-called control files produced by PALS*fit* (either the *PositronFit* or *ResolutionFit* module).

PALS*fit* is available from the website `www.palsfit.dk`.

A contemporary edition of the PATFIT package, roughly equivalent to PALS*fit* without its GUI, is available too. It contains command-driven versions of *PositronFit* and *ResolutionFit* and might be useful for batch processing or in a Linux environment.

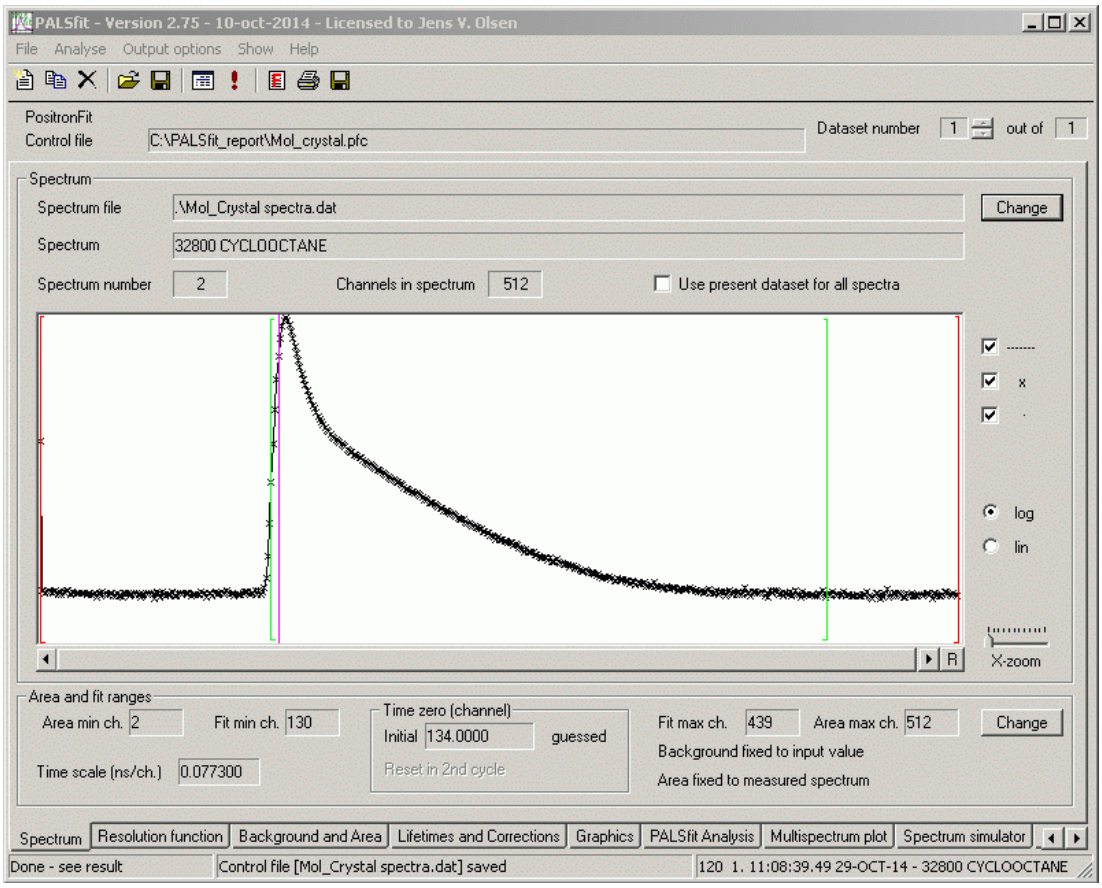

Fig. 1 An example of a window in the PALS*fit* program, which shows a spectrum to be analysed, some of the input parameters for the analysis as well as icons, buttons, menus and tabs that are used to define the analysis and to display the results in numerical or graphical form.

2 About PALS*fit*

2.1 General fitting criterion

Common for *PositronFit* and *ResolutionFit* is that they fit a parameterized model function to a distribution (a "spectrum") of experimental data values y_i. In the actual case these are count numbers which are recorded in "channels". We use the least-squares criterion, i.e. we seek values of the k model parameters $b_1, \ldots, b_k$ that minimizes

$$\phi \equiv \sum_{i=1}^{n} w_i (y_i - f_i(b_1, \ldots, b_k))^2 \tag{1}$$

where n is the number of data values, $f_i(b_1, \ldots, b_k)$ the model prediction corresponding to data value no. i, and w_i a fixed weight attached to i; in this work we use "statistical weighting",

$$w_i = \frac{1}{s_i^2} \tag{2}$$

where s_i^2 is the variance of y_i. As some of the parameters enter our models nonlinearly, we must use an iterative fitting technique. In PALS*fit* we use separable least-square methods to obtain the parameter estimates. Details of the solution methods and the statistical inferences are given in Chapter 4. As a result of the calculations, a number of fitting parameters are estimated that characterize the fitted model function and hence the measured spectrum (e.g. lifetimes and intensities). A number of different constraints may be imposed on the fitting parameters. The two most important types of constraints are that 1) a parameter can be fixed to a certain value, and 2) a linear combination of lifetime intensities is put equal to zero (this latter constraint can be used to fix the ratio of intensities).

2.2 PALS*fit* input

PALS*fit* requires — together with the spectrum to be analysed — a set of input data, e.g. some characteristic parameters of the lifetime spectrometer, guesses of the parameters to be fitted, and possible constraints on these parameters. These data are organised in block structured *control data sets* which in turn are collected in *control files*. A control file may thus contain several control data sets.

PALS*fit* can interactively generate and/or edit the control file for either *PositronFit* or *ResolutionFit*. Previously generated control data can be used as default input values. A number of checks on the consistency of the generated control data are built into PALS*fit*. PALS*fit* is largely self-explanatory regarding input editing.

We recommend that the user reads the model descriptions in the following sections and consults Chapter 6, where the structure of the *PositronFit* and *ResolutionFit* control files is discussed. This will give a clear impression of the input possibilities in PALS*fit*.

Note that control files may also be produced and edited by other means than PALS*fit*.

2.3 PALS*fit* output in general

After a successful *PositronFit* or *ResolutionFit* analysis PALS*fit* produces three files: an analysis report (result file), a graphics file and a table file. The analysis report (result file) has the following contents:

a) An edited result section, which is the *Main Output* for the analysis. It contains the final estimates of the fitting parameters and their standard deviations. In addition, all the guessed input parameters as well as information on constraints are quoted. Furthermore, three statistical numbers, "chi-square", "reduced chi-square", and "significance of imperfect model" are shown. They inform about the agreement between the measured spectrum and the model function (Section 4.4). A few key numbers are displayed for quick reference, giving the number of components and the various types of constraints; they are identified by letters or abbreviations.

b) An input echo (optional). This is a raw copy of all the input data contained in the *Control data set*.

c) Fitting parameters after each iteration (optional). The parameters shown are internal; after convergence they may need a transformation prior to presentation in the Main Output.

d) An estimated correlation matrix for the parameters (optional). This matrix and its interpretation is discussed in Section 4.4.

As indicated above, the outputs b)–d) are optional, while the Main Output is always produced.

The graphics file contains data necessary for the generation of plots of measured and fitted spectra and the table file contains a table with the measured and the fitted spectrum values as a function of the channel number.

Apart from showing a result file, PALS*fit* displays data and results in graphical form on the screen. The PALS*fit* plots include measured or fitted spectra or superpositions of both, as well as deviations between them.

2.4 The *PositronFit* model

The *PositronFit* (and *ResolutionFit*) model function consists of a sum of decaying exponentials convoluted with the resolution function of the lifetime spectrometer, plus a constant background. Let t be the time, k_0 the number of lifetime components, a_j the decay function for component j, R the time-resolution function, and B the background. The resulting expression is given in full detail in Section 5.1; here we state the model in an annotated form using the symbol $*$ for convolution:

$$f(t) = \sum_{j=1}^{k_0} (a_j * R)(t) \; + \; B \tag{3}$$

where

$$a_j(\tau) = \begin{cases} A_j \exp(-(\tau - T_0)/\tau_j), & \tau > T_0 \\ 0, & \tau < T_0 \end{cases} \tag{4}$$

In (4) τ_j is the mean lifetime of the jth component, and A_j is a pre-exponential factor. The integral

$$\int_{T_0}^{\infty} A_j \exp(-(\tau - T_0)/\tau_j)\, \mathrm{d}\tau = A_j \tau_j \tag{5}$$

is called the *area* or the *absolute intensity* of the component. If not for the resolution function R, $t = T_0$ would be the onset time for the decaying exponentials, hence T_0 is called "time-zero". We assume, furthermore, that R is given by a sum of k_g Gaussians which may be displaced with respect to each other:

$$R(\tau) = \sum_{p=1}^{k_g} \omega_p G_p(\tau) \tag{6}$$

where

$$G_p(\tau) = \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left(-\frac{(\tau - \Delta_p)^2}{2\sigma_p^2}\right) \tag{7}$$

and

$$\sum_{p=1}^{k_g} \omega_p = 1 \tag{8}$$

The Gaussian (7) is centered around the *shift* Δ_p. Its standard deviation σ_p is related to its Full Width at Half Maximum by

$$\mathrm{FWHM}_p = 2\sqrt{2\ln 2}\,\sigma_p \tag{9}$$

We also see that

$$\int_{-\infty}^{\infty} R(\tau)\mathrm{d}\tau = 1 \tag{10}$$

Regarding the time scale, the choices of $t = 0$ and the time unit are arbitrary. Considering the actual physical experiment, the positron annihilation lifetimes τ_j are measured in ns (or ps). In this *physical time* representation all the other temporal model parameters in (3–10), i.e. t, τ, T_0, Δ_p, and σ_p, will be in ns too, and it is natural to set $T_0 = 0$.

However, in our internal analysis it is more convenient to use a time scale directly related to the spectrum recording system. Let the multichannel analyser have $n_{\rm ch}$ channels, numbered $i_{\rm ch} = 1, 2, \ldots, n_{\rm ch}$. Each channel represents a time slot whose common width will be used as the time unit. We let channel No. $i_{\rm ch}$ begin at $t = i_{\rm ch} - 1$ and end at $t = i_{\rm ch}$. This implies that $t = 0$ corresponds to the beginning of the first channel. In this channel scale the "time-zero" T_0 will usually take some positive fractional value, say $T_0 = 120.36$. The time t defined in this way is called the *channel time*.

Our way of defining the channel time is by no means standard. Others may prefer to let $t = 0$ fall in the middle of a channel, or may choose to number the channels $0, 1, 2, \ldots$ In earlier versions of our software $t = 0$ corresponded to the left end of a fictive channel 0. Of course such differences have no influence on the analysis itself, except for the nominal value of T_0.

The curve given by (3) is continuous, but since the spectra are recorded in channels of a multichannel analyser, this curve shall for proper comparison be transformed into a histogram by integration over intervals each being one channel wide.

If all the $n_{\rm ch}$ channels in the spectrum are used in the least-squares analysis via (1), we have $n = n_{\rm ch}$ and we simply identify the channel number $i_{\rm ch}$ with the data value number i from (1). Thus we substitute for f_i in (1) the channel average of the model count,

$$f_i = \int_{i-1}^{i} f(t)\ {\rm d}t, \qquad i = 1, \ldots, n \tag{11}$$

with $f(t)$ given by (3), so that (11) is fitted to the measured spectrum. However, often only a subset of the channels are used in the analysis. If this subset starts in channel $i_{\rm ch}^{\rm min}$ and ends in $i_{\rm ch}^{\rm max}$ (inclusive), where

$$1 \le i_{\rm ch}^{\rm min} \le i_{\rm ch}^{\rm max} \le n_{\rm ch} \tag{12}$$

we should generalize (11) to

$$f_i = \int_{i_{\rm ch}^{\rm min}+i-2}^{i_{\rm ch}^{\rm min}+i-1} f(t)\ {\rm d}t, \qquad i = 1, \ldots, n \tag{13}$$

where now

$$n = i_{\rm ch}^{\rm max} - i_{\rm ch}^{\rm min} + 1 \tag{14}$$

Anyway, we obtain as the result a model for the least-squares analysis of the form

$$f_i = \sum_{j=1}^{k_0} F_{ij} \ + \ B \tag{15}$$

where F_{ij} is the contribution from lifetime component j in spectrum channel $i_{\rm ch}^{\rm min} + i - 1$. (We relegate the full write-up of F_{ij} to Section 5.1.) We recall that f_i in (11), (13), and (15) corresponds to $f_i(b_1, \ldots, b_k)$ in Section 2.1, formula (1).

The fitting parameters in *PositronFit* are the *lifetimes* (τ_j), the *relative intensities* defined as

$$I_j = \frac{A_j \tau_j}{\sum_{k=1}^{k_0} A_k \tau_k} \tag{16}$$

the *time-zero* (T_0), and the *background* (B). Each of these parameters may be fixed to a chosen value. In another type of constraint you may put one or more linear combinations of intensities equal to zero in the fitting, i.e.

$$\sum_{j=1}^{k_0} h_{lj} I_j = 0 \tag{17}$$

These constraints can be used to fix ratios of intensities. Finally, it is possible to fix the total area of the spectrum in the fitting,

$$\sum_{j=1}^{k_0} A_j \tau_j + \text{background area} = \text{constant} \tag{18}$$

This may be a useful option if, for example, the peak region of the measured spectrum is not included in the analysis.

Normally in an experiment a fraction α of the positrons will not annihilate in the sample, but for example in the source or at surfaces. In *PositronFit* it is possible to make a correction for this ("source correction"). First, the raw data are fitted in a first iteration cycle. Then, the spectrum of the source correction is subtracted and the corrected spectrum fitted again in a second iteration cycle. In this second cycle it is optional to choose another number of lifetime components as well as type and number of constraints than were used in the first iteration cycle. The source correction spectrum f_i^{s} itself is composed of k_{s} lifetime components and expressed in analogy with (15) (with $B = 0$) as follows:

$$f_i^{\mathrm{s}} = \sum_{j=1}^{k_{\mathrm{s}}} F_{ij}^{\mathrm{s}} \tag{19}$$

If τ_j^{s} and A_j^{s} are the lifetime and pre-exponential factor, respectively, of source-correction component j, then

$$\sum_{j=1}^{k_{\mathrm{s}}} A_j^{\mathrm{s}} \tau_j^{\mathrm{s}} = \alpha \sum_{j=1}^{k_0} A_j \tau_j \tag{20}$$

The necessary mathematical processing of the *PositronFit* model for the least-squares analysis is outlined in Section 5.1.

2.5 The Main Output from *PositronFit*

In the following we give an example of the Main Output part of a *PositronFit* analysis report produced by PALS*fit*, with a brief explanation of its contents (for more details about the input possibilities consult Section 6.1):

```
PALSfit - Version 2.75 10-oct-2014 - Licensed to Jens V. Olsen
Input file: C:\PALSfit_report\Mol_crystal.pfc

P O S I T R O N F I T . Version 2.75      Job time 11:16:32.87 30-OCT-14
*************************************************************************
32800 CYCLOOCTANE
Intrinsic spectrum
*************************************************************************
Data set 1                                          L  T  I  B  Z  A  G
                                                    3  1  1  2  0  1  3
Time scale   ns/channel       :  0.077300
Area range   starts in ch.    35 and ends in ch.   512
Fit range    starts in ch.   136 and ends in ch.   501

Resolution   FWHM (ns)        :    0.5395    0.3539    0.4036
Function     Intensities (%)  :   20.0000   20.0000   60.0000
             Shifts (ns)      :    0.0412   -0.0749    0.0000

----------------- I n i t i a l    P a r a m e t e r s -----------------
             Time-zero (ch.no):  136.3000G
             Lifetimes (ns)   :    0.2300F   0.4000G   2.6500G
             Intensities (%)  :                        28.0000F

Background   Fixed input value:  680.0000
Area         Fixed to measured spectrum ch. 35 -> 512      : 9.05171E+06

----- R e s u l t s  b e f o r e  S o u r c e  C o r r e c t i o n -----
```

```
Convergence obtained after   6 iterations
Chi-square =      557.48  with  362 degrees of freedom
             Lifetimes (ns)   :    0.2300F   0.4401    2.6699
             Intensities (%)  :   30.6011   41.3989   28.0000F
Time-zero    Channel number   :  135.2961
Total-area   From fit         : 9.05171E+06     From table : 9.05171E+06

------------------- S o u r c e  C o r r e c t i o n -------------------
             Lifetimes (ns)   :    0.3840    0.9056
             Intensities (%)  :    8.0000    0.4527
             Total (%)        :  100.0000

--------------- 2 n d  c y c l e  P a r a m e t e r s -----------------
             Lifetimes (ns)   :    0.1200G   0.3600G   1.2000G   2.8000G
             Lin.comb.coeff.  :   -3.0000    0.0000    1.0000    1.0000

Area         Fixed to measured spectrum ch. 35 -> 512      : 8.31407E+06

######################## F i n a l  R e s u l t s ########################
Data set 1                                          L  T  I  B  Z  A  G
                                                    4  0 -1  2  0  1  3
Convergence obtained after   6 additional iterations
Chi-square =      437.05  with  359 degrees of freedom
Reduced chi-square = Chi-square/dof =    1.217  with std deviation 0.075
Significance of imperfect model =  99.70 %

             Lifetimes (ns)   :    0.1374    0.3564    1.2085    2.7592
             Std deviations   :    0.0048    0.0012    0.0975    0.0154

             Intensities(%) LC:   11.1032   55.5874    5.0632   28.2463
             Std deviations   :    0.0963    0.3850    0.2996    0.4631

Background   Counts/channel   :  680.0000
             Std deviations   :     fixed

Time-zero    Channel number   :  135.3537
             Std deviations   :    0.0113
Total-area   From fit         : 8.31407E+06     From table : 8.31407E+06

######################### P o s i t r o n F i t ########################
```

This output was obtained by running PALS*fit* with the control data set listed in Section 6.1. It does not represent a typical analysis of a spectrum, but is rather meant to illustrate a number of features of the program.

After a heading which contains the spectrum headline the key numbers are displayed in the upper right hand corner. `L` indicates the number of lifetime components (k_0), `T` the number of fixed lifetimes, `I` the number and type of intensity constraints (a positive number for fixed intensities, a negative number for linear combinations of intensities, i.e. the number m, Section 6.1, Block 5), `B` the type of background constraint (KB, Section 6.1, Block 6), `Z` whether time-zero is free or fixed (0 = free, 1 = fixed), `A` the type of area constraint (KAR, Section 6.1, Block 7), and `G` the number of Gaussians used to describe the time resolution function (k_g). The rest of the upper part of the output reproduces various input parameters, in particular those for the resolution function, the shape of which is fixed, and the initial values (`G` for guessed and `F` for fixed) of the fitting parameters.

The next part (results before source correction) contains the outcome of the first iteration cycle. If convergence could not be obtained, a message will be given and the iteration procedure discontinued, but still the obtained results are presented. Then follows information about the goodness of the fit (Section 4.4).

The next part (source correction) shows the parameters of the chosen source correction, which accounts for those positrons that annihilate outside the sample, as well as optional initial values of the fitting parameters for the second iteration cycle.

The "Final Results" part contains the number of iterations in the final cycle, followed by three lines with information about the goodness of the fit (Section 4.4). Then follows a survey of the final estimates of the fitted (and fixed) parameters and their standard deviations. The"LC" in the intensity line indicates that we have intensity constraints of the linear-combination type (cf. the negative I in the upper right hand corner). The "total area from fit" is calculated as $\sum_j A_j \tau_j$ plus the background inside the "area range" specified in the beginning of the Main Output. The "total area from table" is the total number of counts in the (source corrected) measured spectrum inside the "area range".

2.6 The *ResolutionFit* model

The *ResolutionFit* model function is the same as for *PositronFit*, Eqs. (3–15). A few additional formulas relevant to *ResolutionFit* are given in Section 5.2. The purpose of *ResolutionFit* is to extract the shape of the resolution function. The widths and displacements (Eqs. (9) and (7)) of the Gaussians in the resolution function are therefore included as fitting parameters. However, in order not to have too many fitting parameters (which may lead to ill-defined estimates of the parameters) the intensities of the Gaussians are fixed parameters. For the same reason it is normally advisable to determine resolution functions by fitting only simple lifetime spectra, i.e. spectra containing only one major lifetime component. The extracted resolution function may then be used in *PositronFit* to analyse more complicated spectra. Along the same line, *ResolutionFit* does not include as many features as does *PositronFit*, e.g. there is no source correction and there are no constraints possible on time-zero or on the area. However, the background can be free or fixed, just like in *PositronFit*.

Hence, the fitting parameters in *ResolutionFit* are the lifetimes (τ_j), their relative intensities (I_j), the background (B), the time-zero (T_0), and the widths and displacements of the Gaussians in the resolution function. Each of these parameters, except T_0, may be constrained to a fixed value and, as in *PositronFit*, linear combinations of lifetime intensities may be constrained to zero in the fitting.

The various input options will be illustrated in the sample output in the next section.

2.7 The Main Output from *ResolutionFit*

In the following we give an example of the Main Output part of a *ResolutionFit* analysis report produced by PALS*fit*, with a brief explanation of its contents (for more details about the input possibilities consult Section 6.2):

```
PALSfit - Version 2.75 10-oct-2014 - Licensed to Jens V. Olsen
Input file: C:\PALSfit_report\Cu-ref.rfc

R E S O L U T I O N F I T Version 2.75    Job time 11:22:45.46 29-OCT-14
*********************************************************************
39699 CU-ANNEALED
in file: .\Metal defects spectra.DAT
*********************************************************************
Data set 1
Time scale   ns/channel       :  0.013400
Area range   starts in ch.     5 and ends in ch.  1000
Fit range    starts in ch.   140 and ends in ch.   500

Initial      FWHM (ns)        :    0.2600G   0.3000G   0.4000G
Resolution   Intensities (%)  :   77.0000   19.0000    4.0000
Function     Shifts (ns)      :    0.0000F   0.0223G  -0.0462G

Other init.  Time-zero (ch.no):  173.0000
Parameters   Lifetimes (ns)   :    0.1100F   0.1800F   0.4000G
```

```
Background   fixed to mean from ch.   650 to ch.  1000   =     12.0342

######################## F i n a l  R e s u l t s ########################

Convergence obtained after  17 iterations
Chi-square =  315.32  with  351 degrees of freedom
Reduced chi-square = Chi-square/dof =    0.898  with std deviation 0.075
Significance of imperfect model =   8.54 %
-------------------------------------------------------------------------
Resolution function:                                              G  W  S
                                                                  3  0  1
             FWHM (ns)        :    0.2396    0.3100    0.2844
             Std deviations   :    0.0007    0.0143    0.0104

             Intensities (%)  :   77.0000   19.0000    4.0000

             Shifts (ns)      :    0.0000    0.0575   -0.1900
             Std deviations   :     fixed    0.0097    0.0056
-------------------------------------------------------------------------
Lifetime components:                                              L  T  I
                                                                  3  2  0
             Lifetimes (ns)   :    0.1100    0.1800    0.4023
             Std deviations   :     fixed     fixed    0.0143


             Intensities (%)  :   86.5086   10.6963    2.7951
             Std deviations   :    1.0833    1.3525    0.2938
-------------------------------------------------------------------------
Background:                                                          B
                                                                     1
             Counts/channel   :   12.0342
             Std deviation    :      mean
-------------------------------------------------------------------------
Time-zero    Channel number   :  170.3136
             Std deviations   :    0.1380

Total-area   From fit         : 1.93631E+06     From table : 1.93653E+06

Shape parameters for resolution curve (nsec):
              N       2       5      10      30     100     300    1000
 FW   at     1/N  0.2552  0.3977  0.4868  0.6217  0.7661  0.8826  0.9928
 MIDP at     1/N  0.0006  0.0010  0.0004 -0.0042 -0.0134 -0.0193 -0.0226

 Peak position of resolution curve: Channel #  170.6092
######################## R e s o l u t i o n F i t ########################
```

This output was obtained by running PALS*fit* with the control data set listed in Section 6.2.

After a heading which includes the spectrum headline, the upper part of the output reproduces various input parameters in a way that is very similar to the *PositronFit* output. The important difference is that in *ResolutionFit* all the FWHMs and all of the displacements (called "shifts" for brevity) except one, may be fitting parameters. In addition, the background is displayed as well as the channels between which it is calculated, if the background is fixed to the mean value between these channel limits.

In the "Final Results" part the number of iterations used to obtain convergence is given first. The next three lines contain information about how good the fit is (for definition of the terms see Section 4.4).

The main part of the output, i.e. the estimated values of the fitted (and fixed) parameters and their standard deviations, follows next (for fixed parameters `FIXED` is written instead of the standard deviation). This part is divided into three, one giving the parameters for the resolution curve, one with the lifetimes and their intensities, and one showing the background. Each part has one or three key numbers displayed in the upper right hand corner. For the resolution function the `G` indicates the number of Gaussians (k_g), `W` the number of fixed widths, and `S` the number of fixed displacements (shifts). For the lifetime components the `L` indicates the number of these (k_0), `T` the number of fixed lifetimes, and `I`

the number and type of intensity constraints. As in *PositronFit*, a positive value means fixed intensities, while a negative value indicates constraints on linear combinations of intensities, the absolute value giving the number of constraints. Next follows the background output, where `B` indicates the type of background constraint (KB, Section 6.2, Block 6), and after the estimated time-zero the "total area from fit" and "total area from table" are given, both calculated as in *PositronFit*. Finally, for easy comparison of the extracted resolution curve with other such curves, a table of the full width of this curve at different fractions of its peak value is displayed, as well as of the midpoints of the curve compared to the peak position. The latter number clearly shows possible asymmetries in the resolution curve. Also the channel number of the position of the peak (maximum value) of the resolution curve is given.

3 Experience with PALS*fit*

In this section we shall give a short account of some of the experiences we (and others) have had with PALS*fit* and its predecessor versions, specifically the program components *PositronFit* and *ResolutionFit*. In general, these fitting programs have proved to be very reliable and easy to use. Further discussion can be found in [1].

The aim of fitting measured spectra will normally be to extract as much information as possible from the spectra. This often entails that one tries to resolve as many lifetime components as possible. However, this has to be done with great care. Because of the correlations between the fitting parameters, and between the fitting parameters and other input parameters, the final estimates of the parameters may be very sensitive to small uncertainties in the input parameters. Therefore, in general, extreme caution should be exercised in the interpretation of the fitted parameters. This is further discussed in e.g. [4, 5, 6, 7, 8, 9, 10, 11]. In this connection, an advantage of the software is the possibility of various types of constraints which makes it possible to select meaningful numbers and types of fitting parameters.

3.1 *PositronFit* experience

The experience gained with *PositronFit* over a number of years shows that in metallic systems with lifetimes in the range $0.1 - 0.5$ ns it is possible to obtain information about at most three lifetime components in unconstrained analyses [12, 13, 14] while in some insulators where positronium is formed, up to four components (unconstrained analysis) may be extracted e.g. [15, 16, 17, 18]. (This does not mean of course that the spectra cannot be composed of more components than these numbers. This problem is briefly discussed in, e.g. [4, 10]. Various other aspects of the analysis of positron lifetime spectra are discussed in for example [19, 20, 21, 22]). In this connection it is very useful to be able to change the number of components from the first to the second iteration cycle. In this way, the spectrum can be fitted with two different numbers of components within the same analysis (it is also advantageous to use this feature when a source correction removes, e.g., a long-lived lifetime component from the raw spectrum).

In our experience *PositronFit* always produces the same estimates of the fitted parameters after convergence, irrespective of the initial guesses (except in some extreme cases). However, others have informed us that for spectra containing very many counts (of the order of 10^7) one may obtain different results, depending upon the initial guesses on the fitting parameters, i.e. local minima exist in the χ^2 as function of the fitting parameters; these minima are often quite shallow. When this happens, *PositronFit* as well as most other

least-squares fitting codes are in trouble, because they just find some local minimum. From a single fitting you cannot know whether the absolute minimum in the parameter space has been found. The problem of "global minimization" is much harder to solve, but even if we could locate the deepest minimum we would have no guarantee that this would give the "best" parameter values from a physical point of view. In such cases it may be necessary to make several analyses of each spectrum with different initial parameter guesses or measure more than one spectrum under the same conditions, until enough experience has been gained about the analysis behaviour for a certain type of spectra.

3.2 *ResolutionFit* experience

When using a software component as *ResolutionFit* an important question of course is whether it is possible in practice to separate the resolution function reliably from the lifetime components. Our experience and those of others [10, 18, 23, 24, 25] suggest that this separation is possible, although in general great care is necessary to obtain well-defined results [1, 10]. The reason for this is the same as mentioned above, viz. that more than one minimum for χ^2 may exist.

From a practical point of view the question arises as to whether there is too strong correlation between some of the parameters defining the resolution function and the lifetime parameters, in particular when three Gaussians (or more) are used to describe the resolution function. As in the example used in this report (Section 2.7), we have often measured annealed copper in order to deduce the resolution function. Even with different settings of the lifetime spectrometer, the copper lifetime normally comes out from a *ResolutionFit* analysis within a few ps (statistical scatter) of 110 ps (in agreement with results of others, e.g. [26]). Thus, the lifetime is well-defined and separable from the resolution function, even though many parameters are free to vary in the fitting procedure. However, because of the many parameters used to describe the resolution function, one frequently experiences that two (or more) different sets of resolution function parameters may be obtained from the same spectrum in different analyses, if different initial guesses are applied. The lifetimes and intensities come out essentially the same in the different analyses, the fits are almost equally good, and a comparison of the widths at the various heights of the resolution curves obtained in the analyses show that they are essentially identical. Thus, in spite of the many fitting parameters (i.e. so many that the same resolution curve may be described by more than one set of parameters), it still seems possible to separate the lifetimes and resolution function reliably, at least when the lifetime spectrum contains a short-lived component of about 90 % intensity or more.

On the other hand, one cannot be sure that the lifetimes can always be separated from the resolution function easily. If, for example, the initial guesses for the fitting parameters are far from the correct parameters, the result of the fitting may be that, for instance, the fitted resolution function is strongly asymmetrical thereby describing in part the slope of the spectrum which arises from the shorter lifetime component. This latter component will then have a shorter lifetime than the correct one. Such cases — where the resolution function parameters will be strongly correlated to the main lifetime — will be more likely the shorter the lifetime is and the broader the resolution function is.

In principle, it is impossible from the analysis alone to decide whether lifetimes and resolution function are properly separated. However, in practice it will normally be feasible. If the main lifetime and the resolution curve parameters are strongly correlated, it is an indication that they are not properly separated. This correlation may be seen by looking for the changes in the lifetimes or resolution function when a small change is made in one of the resolution function parameters (intensity or one of the fitting parameters using a constraint). Other indications that the lifetimes and resolution function are not properly

separated will be that the resulting lifetime deviates appreciably from established values for the particular material or that the half width of the resolution function deviates clearly from the width measured directly with, e.g., a Co-60 source. If the lifetime and the resolution function cannot be separated without large uncertainties on both, one may have to constrain the lifetime to an average or otherwise determined value. Thus, it will always be possible to extract a resolution function from a suitably chosen lifetime spectrum.

A separate question is whether a sum of Gaussians can give a proper representation of the "true" lifetime spectrometer resolution curve, or if some other functional form, e.g., a Gaussian convoluted with two exponentials [23, 25], is better. Of course, it will depend on the detailed shape of the spectrometer resolution curve, but practical experience seems to show that the two descriptions give only small differences in the extracted shape of the curve [10, 25], and the better the resolution is, the less does a small difference influence the extracted lifetime parameters [10]. The sum-of-Gaussians used in PALS*fit* was chosen because such a sum in principle can represent any shape.

Once a resolution function has been determined from one lifetime measurement, another problem arises: Can this function be used directly for another set of measurements? This problem is not directly related to the software, but we shall discuss it briefly here. The accuracy of the determined resolution function will of course depend on the validity of the basic assumption about the measured lifetime spectrum from which it is extracted. This assumption is that the spectrum consists of a known number of lifetime components (e.g. essentially only one as discussed above) in the form of decaying exponentials convoluted with the resolution function. However, this "ideal" spectrum may be distorted in various ways in a real measurement. For example, instead of one lifetime, the sample may give rise to two almost equal lifetimes which cannot be separated. This will, of course, influence the resulting resolution function. So will source or surface components which cannot be clearly separated from the main component. Another disturbance of the spectrum may be caused by gamma-quanta which are scattered from one detector to the other in the lifetime spectrometer. Such scattered photons may give rise to quite large distortions of a lifetime spectrum. How large they are will depend on energy window settings and source-sample-detector arrangement of the lifetime spectrometer [10, 27, 28]. (Apart from the distortions, these spectrometer characteristics will, of course, also influence the width and shape of the correct resolution function.) In digital lifetime spectrometers that have been developed in recent years it seems possible to discriminate more effectively against some of these undesired distortions of measured spectra [29, 30, 31].

Finally, by means of an example let us briefly outline the way we try to obtain the most accurate resolution function for a set of measurements. Let us say that we do a series of measurements under similar conditions (e.g. an annealing sequence for a defect-containing metal sample). In between we measure an annealed reference sample of the same metal, with — as far as possible — the same source and in the same physical arrangement, and thereby determine the resolution curve. This is done for example on January 2, 7, 12, etc. to keep track of possible small changes due to electronic drift. We then make reasonable interpolations between these resolution curves and use the interpolated values in the analysis of the lifetime spectra for the defect containing samples. Sometimes it is not feasible to always measure the annealed sample in exactly the same physical arrangement as the defect containing sample (for example if the annealing sequence takes place in a cryostat). Then we determine resolution curves from measurements on the annealed sample inside and outside the cryostat (the results may be slightly different) before and after the annealing sequence. The possible time variation (due to electronic drift) of the resolution function is then determined from measurements with the annealed sample outside the cryostat. The same variation is finally applied to the resolution curve valid for measurements inside the cryostat.

As we often use many parameters to describe a resolution function these parameters may

appear with rather large scatter. To obtain well-defined variations with time it is often useful in a second analysis of the annealed metal spectra to constrain one or two of the parameters to some average values. With this procedure we believe that we come as close as possible to a reliable resolution function. We are reluctant to determine the resolution function directly from the spectra for the defected metal sample, as we feel that the lack of knowledge of the exact number of lifetime components makes the determination too uncertain.

Let us finally point to one more useful result of an ordinary *ResolutionFit* analysis apart from the extraction of the resolution curve, viz. the determination of the "source correction" (Section 2.4). If the sample gives rise to only one lifetime component, any remaining components must be due to positrons annihilating outside the sample and is therefore normally considered as a source correction. In the *ResolutionFit* Main Output (Section 2.7) the 0.110 ns is the annealed-Cu lifetime, while the 0.18 ns, 10.6962% component is the estimated lifetime and intensity component for the positrons annihilating in the 0.5 mg/cm^2 nickel foil surrounding the source material. The 0.4023 ns, 2.7951% component, that is determined by the analysis, is believed to arise from positrons annihilating in the NaCl source material and on surfaces. This component may be different for different sources and different samples (due to different backscattering). We consider the latter two components as corrections to the measured spectrum in any subsequent *PositronFit* analysis (when the same source and similar sample material have been used).

4 Least-squares fit and statistics

The first four sections of the present chapter contain general information about nonlinear least-squares (NLLS) methods and their statistical interpretations with relevance for PALS*fit*, but without going into details with the specific models involved; these are discussed in Chapters 2 and 5.

The last three sections are of a more technical nature. Section 4.5 presents essential principles of modern NLLS solution methods. Section 4.6 documents the separable least-squares technique which is of utmost importance for the efficiency and robustness of PALS*fit*, and Section 4.7 contains various mathematical and numerical details.

4.1 Unconstrained NLLS data fitting

We shall first present an overview of the classical unconstrained nonlinear least-squares (NLLS) method for data fitting.

In the classical setup it is assumed that some general model is given,

$$y = f(x; b_1, b_2, \ldots, b_k) = f(x; \boldsymbol{b}) \tag{21}$$

where x and y are the independent and dependent variable, respectively, and $\boldsymbol{b} = (b_j)$ is a parameter vector with k components. (All vectors in this work are considered as column vectors.) The components b_j may enter linearly or nonlinearly in (21), and so we may talk about linear and nonlinear parameters b_j. Further, a set of n data points (x_i, y_i) $(i = 1, \ldots, n)$ is given, x_i being the independent and y_i the dependent variable; we shall here introduce the data vector $\boldsymbol{y} = (y_i)$, also called the *spectrum*. Such a spectrum is usually the result of an experiment. We assume $n \geq k$. According to the least squares principle we

should determine $\boldsymbol{b} \in \mathbb{R}^k$ such that

$$\phi(\boldsymbol{b}) = \sum_{i=1}^{n} w_i (y_i - f(x_i; \boldsymbol{b}))^2 \tag{22}$$

is minimized. The w_i are the *weights* of the data; until further notice they are just arbitrary fixed positive numbers accompanying our sample. (In many applications weights are omitted which corresponds to equal weighting, $w_i = 1$.)

When setting up equation (22) it was assumed that the x_i were sample *points* corresponding to the independent variable x in (21). In practice, however, we do not always have this situation. For example, if x represents time, and the equipment records certain events in fixed time intervals (t_{i-1}, t_i) called *channels*, it would be natural to compare y_i with an average of the model function in (21) over (t_{i-1}, t_i). Hence it is appropriate to replace (22) by

$$\phi(\boldsymbol{b}) = \sum_{i=1}^{n} w_i (y_i - f_i(\boldsymbol{b}))^2 \tag{23}$$

In general we shall need a "recipe" $f_i(\boldsymbol{b})$ to compute the model values to be compared with the data values y_i. No sample points x_i of the independent variable enter (23) directly but may possibly be needed for calculating $f_i(\boldsymbol{b})$. The reformulation (23) is just a generalisation of the pointwise formulation (22) who has $f_i(\boldsymbol{b}) = f(x_i, \boldsymbol{b})$. This has no influence on the least squares analysis to be described presently. In the following we shall assume that the functions f_i are sufficiently smooth in the argument $\boldsymbol{b}$.

By introducing the matrix $\mathbf{W} = \mathrm{diag}(w_i)$ and the n-vector $\boldsymbol{f}(\boldsymbol{b}) = (f_i(\boldsymbol{b}))$ we can express (23) in vector notation as follows:

$$\phi(\boldsymbol{b}) = \|\mathbf{W}^{1/2}(\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{b}))\|^2 \tag{24}$$

Here $\|\cdot\|$ denotes the usual Euclidean norm. The corresponding minimization problem reads

$$\phi_{\min} = \min_{\boldsymbol{b} \in \mathbb{R}^k} \{\|\mathbf{W}^{1/2}(\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{b}))\|^2\} \tag{25}$$

A solution $\boldsymbol{b}$ to (25) satisfies the gradient equation

$$\nabla \phi(\boldsymbol{b}) = \mathbf{0} \tag{26}$$

which is equivalent to the k equations

$$\frac{\partial \phi(\boldsymbol{b})}{\partial b_j} = 0, \qquad j = 1, \ldots, k \tag{27}$$

By (23) and (27) we obtain

$$\sum_{i=1}^{n} w_i (y_i - f_i(\boldsymbol{b})) p_{ij} = 0, \qquad j = 1, \ldots, k \tag{28}$$

where

$$p_{ij} = \frac{\partial f_i(\boldsymbol{b})}{\partial b_j} \tag{29}$$

It is practical to collect the derivatives (29) in the $n \times k$ matrix

$$\mathbf{P} = (p_{ij}) \tag{30}$$

The equations (28) are called the *normal equations* for the problem. They are in general nonlinear and must be solved iteratively. Solution methods will be discussed in Sections 4.5 and 4.6. Only for linear or linearized models the normal equations are linear.

It is instructive to consider the linear case in some detail. Here (21) takes the form

$$y = \sum_{j=1}^{k} g_j(x) b_j \tag{31}$$

The x-dependence in $g_j(x)$ is arbitrary and may very well be nonlinear; what matters is that the fitting parameters b_j should enter linearly in the model. The derivatives $p_{ij} = g_j(x_i)$ are independent of b_j, and (23) can be written

$$\phi(\boldsymbol{b}) = \sum_{i=1}^{n} w_i \Big(y_i - \sum_{j=1}^{k} p_{ij} b_j \Big)^2 \tag{32}$$

The normal equations take the classical form

$$\sum_{j'=1}^{k} \sum_{i=1}^{n} w_i p_{ij} p_{ij'} b_{j'} = \sum_{i=1}^{n} w_i y_i p_{ij}, \qquad j = 1, \ldots, k \tag{33}$$

The equations (24–25) can be written

$$\phi(\boldsymbol{b}) = \|\mathbf{W}^{1/2}(\boldsymbol{y} - \mathbf{P}\boldsymbol{b})\|^2 \tag{34}$$

$$\phi_{\min} = \min_{\boldsymbol{b} \in \mathbb{R}^k} \{\|\mathbf{W}^{1/2}(\boldsymbol{y} - \mathbf{P}\boldsymbol{b})\|^2\} \tag{35}$$

The problem (35) is solved by (33) which can be written

$$\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}\,\boldsymbol{b} = \mathbf{P}^{\mathrm{T}}\mathbf{W}\boldsymbol{y} \tag{36}$$

where T stands for transpose. For unweighted data we have $\mathbf{W} = \mathbf{I}_n$ ($\mathbf{I}_n$ is the unit matrix of order n), and so

$$\mathbf{P}^{\mathrm{T}}\mathbf{P}\,\boldsymbol{b} = \mathbf{P}^{\mathrm{T}}\boldsymbol{y} \tag{37}$$

Assuming that the coefficient matrix $\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}$ in (36) is nonsingular, it must be positive definite too. The same applies to $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ in (37). The case described in (31–37) represents a general *linear regression* model. It is a fundamental building block in NLLS procedures and their statistical analysis.

Returning to the nonlinear case we shall ignore the complications from possible non-uniqueness when solving the normal equations (28). Here we just assume that a usable solution $\boldsymbol{b}$ can be found.

4.2 Constraints

It is important to be able to impose constraints on the free variation of the model parameters. In principle a constraint could be an equality, $h(\boldsymbol{b}) = 0$, as well as an inequality $h(\boldsymbol{b}) \geq 0$, where $h(\boldsymbol{b})$ is an arbitrary function of the parameter vector.

Although inequality contraints could sometimes be useful, we abandon them in this work because they would lead to quadratic programming problems, and thereby complicate our models considerably. In our algorithm there is, however, a built-in sign check on some of the nonlinear parameters (e.g. annihilation rates). Should an iteration step make such a parameter negative, a new iterate is determined by halving the correction vector from the old one. As a rule, many such "sign excursions" means an inadequate model parameterizing. On the other hand, no sign checks are made on the linear parameters.

Incorporation of general equality constraints would be possible in the framework of our least-squares method. However, apart from trivial single-parameter constraints, $b_j = c$, linear constraints on the linear parameters are sufficient for our purpose, and as we shall see, involve straightforward generalizations of the unconstrained setup discussed previously.

In Section 4.6 we shall describe the separable least-squares technique used in PALS*fit*. The effect of this method is to define subproblems in which the minimization takes place in the space of the linear parameters only. Hence the incorporation of constraints can just as well be discussed in terms of the linear model (31) where $\phi(\boldsymbol{b})$ is given by (32). In other words, in the constraints analysis we replace k by the number p of linear parameters in

the model and consider an all-linear model where $\boldsymbol{b}$ is replaced by the "linear" parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^p$.

Thus we assume that m independent and consistent linear constraints on the p components of $\boldsymbol{\alpha}$ are given ($m \leq p$):

$$h_{l1}\alpha_1 + \cdots + h_{lp}\alpha_p = \gamma_l, \qquad l = 1, \ldots, m \tag{38}$$

In vector form (38) reads

$$\mathbf{H}\boldsymbol{\alpha} = \boldsymbol{\gamma} \tag{39}$$

where $\mathbf{H} = (h_{lj})$ is an $m \times p$ matrix and $\boldsymbol{\gamma} = (\gamma_l)$ is an m-vector. Both $\mathbf{H}$ and the augmented matrix $(\mathbf{H}, \boldsymbol{\gamma})$ are of rank m.

A number of technical questions about how the constraints (38) or (39) influence the NLLS procedure will be discussed in Section 4.7.

4.3 Statistical analysis

In this and the following section we address the question of the statistical scatter in the parameters and $\phi_{\min}$ that can be expected in NLLS parameter estimation.

Suppose the spectrum (y_i) contains experimental values subject to statistical fluctuations, while the weights (w_i) are fixed. Ideally we should imagine an infinite ensemble of similar spectra $\boldsymbol{y} = (y_i)$ be given. Let us first consider the unconstrained case. Through solution of the normal equations (28) each spectrum $\boldsymbol{y}$ gives rise to a parameter estimate $\boldsymbol{b} = \boldsymbol{b}(\boldsymbol{y})$. Hence also $\boldsymbol{b}$ becomes a random (vector) variable with a certain joint distribution.

We shall use the symbol E[·] for expected value (ensemble mean) and Var[·] for variance. We introduce the "ensemble-mean spectrum"

$$\boldsymbol{\eta} = (\eta_i) = \mathrm{E}[\boldsymbol{y}] \tag{40}$$

and the corresponding hypothetic estimate

$$\boldsymbol{b}_0 = (b_{j0}) = \boldsymbol{b}(\boldsymbol{\eta}) \tag{41}$$

Thus $\boldsymbol{b}_0$ is the solution of (28) corresponding to the particular spectrum (η_i). Now, given an arbitrary spectrum (y_i), let the corresponding parameter vector be $\boldsymbol{b} = (b_j)$. If we assume that $\boldsymbol{b} - \boldsymbol{b}_0 = \Delta\boldsymbol{b} = (\Delta b_j)$ is so small that our model is locally linear in $\boldsymbol{b}$ around $\boldsymbol{b}_0$, we have to a first-order approximation

$$f_i(\boldsymbol{b}) = f_i(\boldsymbol{b}_0) + \sum_{j=1}^{k} p_{ij}\Delta b_j \tag{42}$$

where p_{ij} are the derivatives (29) evaluated at $\boldsymbol{b}_0$. We insert (42) into the normal equations (28) and obtain a linear equation system of order k with Δb_j as unknowns. In vector notation this system reads

$$\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}\,\Delta\boldsymbol{b} = \mathbf{P}^{\mathrm{T}}\mathbf{W}\,\Delta\boldsymbol{y} \tag{43}$$

where $\Delta\boldsymbol{y}$ is a vector with components

$$\Delta y_i = y_i - f_i(\boldsymbol{b}_0), \qquad i = 1, \ldots, n \tag{44}$$

We note the similarity with the linear case (36). The system (43) has the solution

$$\Delta\boldsymbol{b} = \mathbf{K}\,\Delta\boldsymbol{y} \tag{45}$$

where

$$\mathbf{K} = (\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{W} \tag{46}$$

The *covariance matrix* of a vector variable $\boldsymbol{v}$ will here be denoted $\boldsymbol{\Sigma}(\boldsymbol{v})$. (Other names for this matrix are dispersion matrix and variance-covariance matrix, since the diagonal row

contains the component variances.) It is well-known that if two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ are related by a linear transformation

$$\boldsymbol{w} = \mathbf{A}\boldsymbol{v} \tag{47}$$

then

$$\boldsymbol{\Sigma}(\boldsymbol{w}) = \mathbf{A}\boldsymbol{\Sigma}(\boldsymbol{v})\mathbf{A}^{\mathrm{T}} \tag{48}$$

Our primary goal is to estimate the covariance matrix

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = (\sigma_{jj'}) \tag{49}$$

Equation (45) shows that $\Delta\boldsymbol{b}$ is related to $\Delta\boldsymbol{y}$ by a (locally) linear transformation, and so we obtain from (48) the approximate result

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = \mathbf{K}\boldsymbol{\Sigma}(\boldsymbol{y})\mathbf{K}^{\mathrm{T}} \tag{50}$$

We now assume that the measurements y_i are independent. Let

$$\mathrm{Var}[y_i] = s_i^2, \qquad i = 1, \ldots, n \tag{51}$$

such that s_i is the standard deviation of y_i. Then $\boldsymbol{\Sigma}(\boldsymbol{y}) = \mathrm{diag}(s_i^2)$. We also assume that the variances s_i^2 $(i = 1, \ldots, n)$ are known, or at least that estimates are available. With this knowledge it is appropriate to use the *statistical weighting* introduced in (2) in Section 2.1. We can show that this leads to a simple form of $\boldsymbol{\Sigma}(\boldsymbol{b})$. By using (50) and observing that (2) implies

$$\mathbf{W}\boldsymbol{\Sigma}(\boldsymbol{y}) = \mathbf{I}_n \tag{52}$$

we obtain after reduction the formula

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = (\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P})^{-1} \tag{53}$$

which holds at least approximately. It is exact in the linear regression case (36).

Still under the assumption of a locally linear model and of statistical weighting as described, we shall next study the distribution of $\phi_{\min}$ in (25). Here we make the additional assumptions that we have an *ideal model*, i.e.

$$f_i(\boldsymbol{b}_0) = \eta_i \tag{54}$$

and that each measurement y_i has a Gaussian distribution,

$$y_i \in \mathrm{N}(\eta_i, s_i^2) \tag{55}$$

Then by (44) and (54)

$$\Delta y_i \in \mathrm{N}(0, s_i^2) \tag{56}$$

In Section 4.7 it is shown that $\phi_{\min}$ under these assumptions has a χ^2-distribution with f degrees of freedom,

$$\phi_{\min} \in \chi^2(f) \tag{57}$$

where f is the number of data values minus the number of fitted parameters,

$$f = n - k \tag{58}$$

For this reason $\phi_{\min}$ is often called χ^2. Thus

$$\chi^2 \equiv \phi_{\min} = \min_{\boldsymbol{b}\in\mathbb{R}^k} \sum_{i=1}^{n} \Big(\frac{y_i - f_i(\boldsymbol{b})}{s_i}\Big)^2 \tag{59}$$

The results derived for $\boldsymbol{\Sigma}(\boldsymbol{b})$ and $\phi_{\min}$ are independent of the applied fitting technique. But we have assumed an unconstrained variation of all components of the k-vector $\boldsymbol{b}$. When linear constraints on the linear parameters are included, the analysis still holds for a "basic subset" of k_{free} independent parameter components, as will be shown in Section 4.7. Thus in the distribution (57–58) for $\phi_{\min}$ we should replace k with k_{free}. To obtain $\boldsymbol{\Sigma}(\boldsymbol{b})$ we incorporate the linear constraints (38) or (39) and express the remaining components

(deterministically) in terms of the free ones. These operations as well as the resulting formula for $\mathbf{\Sigma}(\boldsymbol{b})$ are given in Section 4.7.

If the parameter vector $\boldsymbol{b}$ is transformed to another vector $\boldsymbol{b}_1$ before the output is presented, the covariance matrix of $\boldsymbol{b}_1$ is computed as

$$\mathbf{\Sigma}(\boldsymbol{b}_1) = \mathbf{J}\mathbf{\Sigma}(\boldsymbol{b})\mathbf{J}^{\mathrm{T}} \tag{60}$$

where

$$\mathbf{J} = \mathrm{d}\boldsymbol{b}_1/\mathrm{d}\boldsymbol{b} \tag{61}$$

is the Jacobian of the transformation, cf. (48). In PALS*fit* we use only simple transformations when passing from $\boldsymbol{b}$ to $\boldsymbol{b}_1$, or no transformation at all. Examples are lifetimes τ_j in ns instead of annihilation rates λ_j in channels^{-1}, and widths in FWHM instead of in standard deviations. These give rise to trivial diagonal elements in $\mathbf{J}$. On the other hand, the presentation of relative intensities I_j instead of absolute intensities J_j induces a diagonal block in the upper-left corner of $\mathbf{J}$ with the (j, j')-entry $I_j(\delta_{jj'} - I_j)/J_j$.

4.4 Statistical interpretations

In Section 4.3 we discussed the estimation of the covariance matrix

$$\mathbf{\Sigma}(\boldsymbol{b}) = \begin{pmatrix} \sigma_{11} & \dots & \sigma_{1k} \\ \dots & \dots & \dots \\ \sigma_{k1} & \dots & \sigma_{kk} \end{pmatrix} \tag{62}$$

The standard deviations of the estimated parameters are extracted from its diagonal as $\sigma_j = \sqrt{\sigma_{jj}}$, while the off-diagonal entries contain the covariances. In the usual way we construct the correlation matrix

$$\mathbf{R} = \begin{pmatrix} 1 & \dots & \rho_{1k} \\ \dots & \dots & \dots \\ \rho_{k1} & \dots & 1 \end{pmatrix} \tag{63}$$

by the formula

$$\rho_{jj'} = \sigma_{jj'}/(\sigma_j\sigma_{j'}) \tag{64}$$

A consequence of the assumed normal distribution of y_i is that the parameter estimates too will be (approximately) normally distributed and their joint distribution is completely determined by the covariance matrix $(\sigma_{jj'})$. The natural statistical interpretation of $(\sigma_{jj'})$ or $(\sigma_j, \rho_{jj'})$ is an estimate of the covariance structure of the computed parameters in a series of repetitions of the spectrum recording under identical physical conditions. We would expect that $(\sigma_{jj'})$ might show a good deal of scatter in such a series.

The standard deviations and correlation coefficients may be used to compute estimated standard deviations of new parameters that are functions of the primary parameters presented in the output from PALS*fit*, e.g. a mean lifetime or a trapping rate. The standard deviation of such a parameter, z, is given (to a first-order approximation) by:

$$\sigma_z = \Big\{ \sum_{j=1}^{k}\sum_{j'=1}^{k} \frac{\partial z}{\partial b_j}\frac{\partial z}{\partial b_{j'}}\rho_{jj'}\sigma_j\sigma_{j'} \Big\}^{\frac{1}{2}} =$$

$$\Big\{ \sum_{j=1}^{k} \Big(\frac{\partial z}{\partial b_j}\Big)^2\sigma_j^2 + 2\sum_{j=1}^{k-1}\sum_{j'=j+1}^{k} \frac{\partial z}{\partial b_j}\frac{\partial z}{\partial b_{j'}}\rho_{jj'}\sigma_j\sigma_{j'} \Big\}^{\frac{1}{2}} \tag{65}$$

This result follows by the transformation rule (60).

There is another property of the correlation matrix which might be useful in practice. Suppose an analysis of a given spectrum results in an estimated parameter vector $\mathbf{b} = (b_j)$, $j = 1, \dots, k$. One may ask: What happens to the remaining components if one of them,

say b_1, is forced to be shifted a small amount Δb_1, and the analysis is repeated with the same spectrum? It can be shown that the other components will be shifted according to the formula

$$\Delta b_j = (\sigma_j/\sigma_1)\rho_{1j}\,\Delta b_1, \qquad j = 2, \ldots, k \tag{66}$$

A proof can be found in Section 4.7.4. The formula (66) refers to a single spectrum and is therefore deterministic. In principle its validity is restricted to small shifts due to the nonlinearity of our models. In our experience the formula is applicable up to at least $\Delta b_1 \approx 2\text{–}3 \times \sigma_1$ for well-defined fitting problems with small σ_j. For fits with large σ_j it seems to be valid only up to $\Delta b_1 \approx 0.1\text{–}0.2 \times \sigma_1$, and in certain pathological cases it fails completely; such failures may be ascribed to imperfect models or strong nonlinearities.

We saw in Section 4.3 that $\chi^2 = \phi_{\min}$ in (59) under certain assumptions has a χ^2-distribution with f degrees of freedom, i.e. (57) holds good with

$$f = n - k + m = n - k_{\text{free}} \tag{67}$$

Here m is the number of constraints, so that $k_{\text{free}} = k - m$ is the effective number of free parameters in the estimation. The mean and variance of $\phi_{\min} = \chi^2$ are

$$\mathrm{E}[\chi^2] = f \tag{68}$$

and

$$\mathrm{Var}[\chi^2] = 2f \tag{69}$$

From the χ^2 statistics one can derive a "goodness-of-fit" significance test for the validity of the asserted ideal model, cf. (54). In such a χ^2-test we compute the probability $P\{\chi^2 < \chi^2_{\text{obs}}\}$ that a χ^2-distributed variable with f degrees of freedom will not exceed the observed value χ^2_{obs}. A value close to 100% indicates systematic deviation from the assumed model, and we use the phrase "significance of imperfect model" for this probability. We also compute the quantity

$$V = \chi^2/f, \tag{70}$$

with mean

$$\mathrm{E}[V] = 1 \tag{71}$$

and variance

$$\mathrm{Var}[V] = 2/f \tag{72}$$

V is sometimes called the "reduced chi-square" or the "variance of the fit"; with a good fit this quantity should be close to unity.

We conclude this section with some comments on the underlying assumptions in the statistical NLLS analysis which were:

1. Small fluctuations of each data value y_i, i.e. $\mathrm{Var}[y_i]$ small.
2. Our model is only weakly nonlinear in the parameter vector $\boldsymbol{b}$.
3. An ideal model which means that (54) holds.
4. The data values y_i are independent.
5. Each y_i has a Gaussian distribution.
6. "Statistical weighting" (2).
7. The population variances $\mathrm{Var}[y_i]$ are known in advance.

Assumptions 1, 2, 3 should be considered together; for example, violation of 1 and 3 may both invalidate the linear approximation (42).

For NLLS problems with strong non-linearities it is well-known that our covariance matrix formula, which is based on linear expansion of the model, may in general produce over-optimistic standard deviation estimates. Nevertheless, such estimates could be useful for qualitative purposes.

Assumption 1 is a fair approximation in PALS*fit* applications, where it should be understood in the relative meaning; it holds provided the counts y_i are not too small.

Assumption 3 expresses that our model "explains" the observed data perfectly, apart from the inevitable statistical noise. This hypothesis was subject to a chi-square goodness-of-fit test as explained.

Assumption 4 is natural in many applications; however in practice some measurements might show a certain correlation between neighbouring data values.

Assumption 5 is needed only for the analysis of the goodness-of-fit. Many distributions encountered in practice do not deviate much from the normal distribution and thus admits an approximately correct analysis. In particular, this is true for Poisson counting statistics, again provided the counts are large enough.

Regarding Assumption 6, statistical weighting is a convenient means to equalize the impact from the individual observations y_i on the fit. To accomplish it we shall need (estimates of) the variances $\mathrm{Var}[y_i]$ (see also Assumption 7).

Regarding Assumption 7, the theoretical values of the population variances $\mathrm{Var}[y_i]$ are sometimes unavailable and need to be estimated. In some applications the variances are only known up to a constant of proportionality. By using statistical weighting nevertheless, this would not affect the outcome of the NLLS parameter estimation itself. However, the chi-square analysis would not be possible in the usual way due to the lack of normalization.

4.5 Marquardt minimization

As mentioned in Section 4.1 the normal equations (28) are in general nonlinear and must be solved iteratively. We now describe such an iterative method called Marquardt's principle, which is an efficient combination of two classical unconstrained minimization procedures; contraints will be taken care of as described in Sections 4.2 and 4.7.

Basically, we use Newton's iterative method (other names are the Gauss-Newton or the Taylor series method), which we shall presently explain. However, first we shall prove the following expansion formula which is approximately correct provided $\boldsymbol{d}$ is small, the fit is good, the model locally linear, and $\boldsymbol{b}$ is close to the solution of the NLLS problem:

$$\phi(\boldsymbol{b}+\boldsymbol{d}) = \phi(\boldsymbol{b}) + \nabla\phi(\boldsymbol{b})\cdot\boldsymbol{d} + \boldsymbol{d}^T\mathbf{P}^T\mathbf{W}\mathbf{P}\boldsymbol{d} \tag{73}$$

with the usual meaning of $\mathbf{W}$ and $\mathbf{P}$. Using a quadratic Taylor expansion we obtain

$$\phi(\boldsymbol{b}+\boldsymbol{d}) = \phi(\boldsymbol{b}) + \nabla\phi(\boldsymbol{b})\cdot\boldsymbol{d} + \tfrac{1}{2}\boldsymbol{d}^T\mathbf{S}\boldsymbol{d} + \mathcal{O}(\|\boldsymbol{d}\|^3) \tag{74}$$

Here $\mathbf{S} = \{s_{jj'}\}$ is the Hessian of $\phi(\boldsymbol{b})$. From the expression (23) we find

$$s_{jj'} = \frac{\partial^2\phi(\boldsymbol{b})}{\partial b_j \partial b_{j'}} = 2\sum_{i=1}^{n} w_i\Big(\frac{\partial f_i}{\partial b_j}\frac{\partial f_i}{\partial b_{j'}} - (y_i - f_i)\frac{\partial^2 f_i}{\partial b_j \partial b_{j'}}\Big) \tag{75}$$

with $f_i = f_i(\boldsymbol{b})$. We shall neglect the term $\sum_{i=1}^{n} w_i(y_i - f_i)\partial^2 f_i/\partial b_j \partial b_{j'}$ in (75). The reason for doing so is that we expect some cancellation to take place in the summation process, because the residuals $y_i - f_i$ are supposed to fluctuate around zero when the fit is good. We have also assumed that the second derivatives $\partial^2 f_i/\partial b_j \partial b_{j'}$, which express the nonlinearity of the model, are not too large. Hence, approximately

$$\mathbf{S} = 2\mathbf{P}^T\mathbf{W}\mathbf{P} \tag{76}$$

Inserting this in (74) we establish (73). Returning to Newton's method, let $\boldsymbol{b}$ be a guessed or previously iterated parameter vector. Newton's correction step $\boldsymbol{d}$ now solves the local minimization problem

$$\min_{\boldsymbol{d}\in\mathbb{R}^k}\{\phi(\boldsymbol{b}+\boldsymbol{d})\} \tag{77}$$

where $\phi(\boldsymbol{b}+\boldsymbol{d})$ is approximated by (73). For brevity we shall write

$$\mathbf{A} = \mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P} \tag{78}$$

Assuming that $\mathbf{P}$ has full rank, $\mathbf{A}$ will be positive definite. By taking gradients we obtain

$$\nabla\phi(\boldsymbol{b}+\boldsymbol{d}) = \nabla\phi(\boldsymbol{b}) + 2\mathbf{A}\boldsymbol{d} + \mathcal{O}(\|\boldsymbol{d}\|^2) \tag{79}$$

We equate this to zero and then compute the Newton step from the normal equation system (cf. (36) and (43))

$$\mathbf{A}\,\boldsymbol{d} = \boldsymbol{g} \tag{80}$$

Here the vector $\boldsymbol{g}$ is given by

$$\boldsymbol{g} = -\frac{1}{2}\nabla\phi(\boldsymbol{b}) \tag{81}$$

According to (23) its components are

$$g_j = \sum_{i=1}^{n} w_i(y_i - f_i(\boldsymbol{b}))p_{ij}, \qquad j = 1,\ldots,k \tag{82}$$

Subsequently $\boldsymbol{b}+\boldsymbol{d}$ replaces $\boldsymbol{b}$ as the new iterate, and the iterations continue. With the pure Newton method we cannot guarantee that the new $\phi = \phi(\boldsymbol{b}+\boldsymbol{d})$ is smaller than the old one. Indeed the procedure often tends to diverge due to strong nonlinearities, in particular when the initial guess is bad. To ensure a decrease in ϕ we introduce the Marquardt [32] modification of (80),

$$(\mathbf{A} + \lambda\mathbf{D})\boldsymbol{d} = \boldsymbol{g} \tag{83}$$

where $\mathbf{D}$ is a diagonal matrix with the same diagonal row as the positive definite matrix $\mathbf{A}$. λ is a parameter that is at our disposal. It provides interpolation between Newton's method and a gradient-like method. The former is obtained by setting $\lambda = 0$, cf. (80). On the other hand, when $\lambda \to \infty$ we obtain a solution vector proportional to $\mathbf{D}^{-1}\boldsymbol{g}$. According to (81) $\boldsymbol{g}$ is proportional to the negative gradient vector $-\nabla\phi$, so $\mathbf{D}^{-1}\boldsymbol{g}$ becomes a scaled version of $-\nabla\phi$ and shares with this the property that ϕ (assumed $> \phi_{\min}$) certainly decreases initially along the correction vector, although it need not have the steepest descent direction. We can now roughly sketch Marquardt's procedure. The equation to be solved at iteration number r reads

$$(\mathbf{A}^{(r)} + \lambda^{(r)}\mathbf{D}^{(r)})\boldsymbol{d}^{(r)} = \boldsymbol{g}^{(r)} \tag{84}$$

From its solution $\boldsymbol{d}^{(r)}$ we calculate

$$\boldsymbol{b}^{(r+1)} = \boldsymbol{b}^{(r)} + \boldsymbol{d}^{(r)} \tag{85}$$

and a new ϕ-value, $\phi^{(r+1)}$. Now it is essential that $\lambda^{(r)}$ is so chosen that

$$\phi^{(r+1)} \le \phi^{(r)} \tag{86}$$

If we are not already at the minimum, it is always possible to satisfy (86) by selecting a sufficiently large $\lambda^{(r)}$, and so we avoid the divergence problems encountered in Newton's method. However, $\lambda^{(r)}$ should not be chosen unnecessary large, because we then get a small correction vector of gradient-like type which would give slow convergence. In the later iterations, when convergence is approached, λ should be small. Then we approach Newton's method which has a fast (quadratic) rate of convergence near the minimum. The procedure has converged when $\phi^{(r)}$ and $\boldsymbol{b}^{(r)}$ are stationary with increasing r. For the detailed strategy we refer to Marquardt [32]. The algorithm is sometimes called the Levenberg-Marquardt method (LM) since Levenberg [33] already in 1944 put forward essential parts of the ideas taken up by Marquardt in 1963 [32].

Over the years LM has undergone a number of refinements, adding more robustness to it. In earlier versions of PALS*fit* we used LM as in [32]. But pure LM puts no bounds on the step vector $\boldsymbol{d}(\lambda)$. Modern LM implementations use a "trust-region" enhancement and replace the unrestricted minimization of ϕ by the quadratic programming problem

$$\min_{\boldsymbol{d}(\lambda)} \left\{ \phi : \|\mathbf{D}^{1/2}\boldsymbol{d}(\lambda)\| \leq \Delta \right\} \tag{87}$$

The effect is to restrict the size of $\boldsymbol{d} = \boldsymbol{d}(\lambda)$. The bound Δ is adjusted each time a *major* iteration step begins and is decreased if the solution of (87) does not provide a suitable correction. We have adopted this idea for use in PALS*fit* from the work of Moré [34], as implemented in the software package MINPACK-1 [35] for unconstrained least-squares minimization. A subroutine from this package, LMPAR, performs *minor* iterations by finding a value of λ that solves (87) approximately. The optimal λ is saved for use as an initial estimate in the next major step. Details of this technique are found in Moré [34].

4.6 Separable least-squares technique

A substantial gain in computing efficiency can be obtained when some of the k components of the parameter vector $\boldsymbol{b}$ enter our model linearly. Indeed this is the case in PALS*fit*. The least-squares problem is then called *separable* or *semilinear*. Separable procedures have been studied by several authors [36, 37, 38, 39, 40, 41] and have proved successful in this work and in many other applications as well. In the separable case (21) can be written

$$y = f(x;\boldsymbol{b}) = f(x;\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{j=1}^{p} \alpha_j f_j(x;\boldsymbol{\beta}) \tag{88}$$

This means a partitioning of the k-vector

$$\boldsymbol{b} = \begin{matrix} p \\ q \end{matrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \tag{89}$$

into a "linear" p-vector $\boldsymbol{\alpha} = (\alpha_j)$ and a "nonlinear" q-vector $\boldsymbol{\beta} = (\beta_j)$, where $p + q = k$. Corresponding to (88) we have

$$f_i(\boldsymbol{b}) = \sum_{j=1}^{p} \alpha_j f_{ij}(\boldsymbol{\beta}) = (\mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha})_i, \qquad i = 1,\ldots,n \tag{90}$$

where $\mathbf{F} = \mathbf{F}(\boldsymbol{\beta})$ is an $n \times p$ matrix with elements $f_{ij} = f_{ij}(\boldsymbol{\beta})$. With these definitions $\boldsymbol{f}(\boldsymbol{b})$ entering (25) can be written

$$\boldsymbol{f}(\boldsymbol{b}) = \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha} \tag{91}$$

In separable NLLS we consider the linear subproblems of (25) where $\boldsymbol{\beta}$ is fixed and $\boldsymbol{\alpha}$ varies:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^p} \{\|\mathbf{W}^{1/2}(\boldsymbol{y} - \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha})\|^2 : \boldsymbol{\beta} \text{ fixed}\} \tag{92}$$

Considering first the unconstrained case, the standard linear least-squares analysis tells that $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\boldsymbol{\beta})$ is the solution of the pth-order normal equation system

$$\mathbf{F}^{\mathrm{T}}\mathbf{W}\mathbf{F}\boldsymbol{\alpha} = \mathbf{F}^{\mathrm{T}}\mathbf{W}\boldsymbol{y} \tag{93}$$

cf. the linear regression case (31–37). Turning to the determination of the nonlinear part $\boldsymbol{\beta}$ of the parameter vector $\boldsymbol{b}$, we realize that an iterative method is needed. In fact, there will be an outer loop, where each step provides a correction vector $\boldsymbol{d}$ to $\boldsymbol{\beta}$, and an inner procedure which invokes a linear minimization (92–93) each time a new trial value of $\boldsymbol{\beta}$ is chosen. We can formulate the nonlinear outer minimization as follows:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^q} \{\phi(\boldsymbol{\beta}) \equiv \|\mathbf{W}^{1/2}(\boldsymbol{y} - \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha}(\boldsymbol{\beta}))\|^2\} \tag{94}$$

We solve (94) by a modified Marquardt procedure as explained in Section 4.5, where $\boldsymbol{b}$ should be replaced by $\boldsymbol{\beta}$. Indeed, equation (83) takes the form

$$(\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P} + \lambda\mathbf{D})\boldsymbol{d} = \mathbf{P}^{\mathrm{T}}\mathbf{W}(\boldsymbol{y} - \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha}) \tag{95}$$

where $\mathbf{P}$ is now the $n \times q$ matrix with elements

$$p_{ij} = \frac{\partial f_i}{\partial \beta_j} \tag{96}$$

and $\mathbf{D}$ is a diagonal matrix with the same diagonal row as $\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}$.

A crucial point in the separable procedure is the evaluation of (96), which can be accomplished by considering the vector $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{b})$ in (91). Note that $\boldsymbol{f}$ depends on $\boldsymbol{\beta}$ directly through $\mathbf{F}$ and indirectly through $\boldsymbol{\alpha}$; hence

$$\frac{\partial \boldsymbol{f}}{\partial \beta_j} = \frac{\partial \mathbf{F}}{\partial \beta_j}\boldsymbol{\alpha} + \mathbf{F}\frac{\partial \boldsymbol{\alpha}}{\partial \beta_j} \tag{97}$$

To find $\partial\boldsymbol{\alpha}/\partial\beta_j$ we take the derivative of both members of (93). This leads to

$$\mathbf{F}^{\mathrm{T}}\mathbf{W}\mathbf{F}\frac{\partial \boldsymbol{\alpha}}{\partial \beta_j} = \frac{\partial \mathbf{F}^{\mathrm{T}}}{\partial \beta_j}\mathbf{W}(\boldsymbol{y} - \mathbf{F}\boldsymbol{\alpha}) - \mathbf{F}^{\mathrm{T}}\mathbf{W}\frac{\partial \mathbf{F}}{\partial \beta_j}\boldsymbol{\alpha}, \qquad j = 1, \ldots, q \tag{98}$$

For an ideal model the term in (98) containing the residual vector $\boldsymbol{y} - \mathbf{F}\boldsymbol{\alpha}$ is negligible when the minimum is approached, but is important when the current iterate is far from convergence.

Now we can give a summary of the complete strategy for the unconstrained separable minimization of ϕ: Start the outer iterations from a guessed value of $\boldsymbol{\beta}$, and select suitable initial values for λ and the bound Δ. For each outer iteration, solve the linear subproblem (92–93) for $\boldsymbol{\alpha}$ and calculate ϕ. Compute the Jacobian elements $\partial\boldsymbol{f}/\partial\beta_j$ from (98) and (97), and form $\mathbf{P}$ and $\mathbf{D}$. Then enter an inner procedure and find near-optimal values of λ and the correction vector to $\boldsymbol{\beta}$, $\boldsymbol{d} = \boldsymbol{d}(\lambda)$, using Marquardt's method with Moré's modification. Update the bound Δ, replace $\boldsymbol{\beta}$ by $\boldsymbol{\beta}+\boldsymbol{d}$, and resume the outer iteration loop. The procedure is finished when ϕ has proved to be stationary.

When implementing our separable algorithm, there is a practical difficulty in handling $\partial\boldsymbol{f}/\partial\beta_j$ in (97). For each data value we must evaluate a $p \times q$ matrix of scalar derivatives which means altogether $n \times p \times q$ values. To reduce the memory demand we use a packed ("sparse-matrix") scheme for storing only the nonzero derivatives.

Linear constraints on linear model parameters, as they occur in PALS*fit*, are readily integrated in the separable NLLS procedure, cf. Sections 4.2 and 4.7.

The numerical solution of many of the linear-algebraic and optimization subproblems in our algorithm is accomplished by software from the standard packages LINPACK [42] and MINPACK-1 [35]. To accommodate application of this software we found it convenient to rescale the NLLS problem, cf. Section 4.7. In that section we also give some comments on the practical numerical solution of NLLS subproblems.

4.7 Various mathematics, statistics, and numerics

In this section a number of technical details are collected. They all have relevance to the previous sections in this chapter.

4.7.1 Estimation of background and weight smoothing

Considering Assumptions 6 and 7 in Section 4.4 we face the problem that we do not know $s_i^2 = \mathrm{Var}[y_i]$ when setting up the statistical weighting (2). Each count y_i is distributed in a

Poisson distribution with a certain mean value η_i,

$$y_i \in \mathrm{P}(\eta_i) \tag{99}$$

implying that

$$\mathrm{E}[y_i] = \eta_i \tag{100}$$

$$\mathrm{Var}[y_i] = \eta_i \tag{101}$$

The ideal weighting

$$w_i = \frac{1}{\eta_i} \tag{102}$$

would provide approximately central (i.e. unbiased) least-squares estimates of the model parameters. Since the η_i are unknown we must use some kind of approximation. We may simply take

$$w_i = \frac{1}{y_i} \tag{103}$$

or rather use the modified formula

$$w_i = \frac{1}{\max(y_i, 1)} \tag{104}$$

because it is possible to record $y_i = 0$. However, in the following we shall assume that the probability of this is negligible,

$$P\{y_i = 0\} \approx 0 \tag{105}$$

The "raw" weights (103) will normally fluctuate, and we shall now show that this may induce a bias on the parameter estimates. In a fitting model (3) with a free background B it is often the case that B is virtually independent of the other parameters. This means that obtaining a least-squares estimate B^* of B is essentially a one-parameter problem,

$$B^* = \mathrm{argmin}_{B\in\mathbb{R}} \sum_{i=1}^{n} w_i (y_i - B)^2 \tag{106}$$

which has the unique solution

$$B^* = \frac{\sum w_i y_i}{\sum w_i} \tag{107}$$

In the "theoretical" case (102) we evaluate $B^* = B_0^*$ by inserting (102) in (107). Because B is the only parameter we have $\eta_i = B$ and then

$$B_0^* = \frac{\sum y_i}{n} = \langle y_i \rangle \tag{108}$$

where $\langle\cdot\rangle$ stands for averaging over channels. B_0^* is indeed central as

$$\mathrm{E}[B_0^*] = \mathrm{E}[y_i] = \eta_i = B \tag{109}$$

On the other hand, in the "real" case (103), we evaluate $B^* = B_1^*$ by inserting (103) in (107), and then it turns out that B_1^* is not central. In fact we can show that B_1^* underestimates the background roughly by 1,

$$\mathrm{E}[B_1^*] \approx B - 1 \tag{110}$$

under the additional assumptions that the number of channels n in the analysis as well as the counts y_i and the background B are reasonably large. We find

$$B_1^* = \frac{n}{\sum \frac{1}{y_i}} = \frac{1}{\langle \frac{1}{y_i} \rangle} \tag{111}$$

From the Poisson distribution (99) we obtain the probability

$$P\{y_i = k\} = \frac{B^k}{k\,!} e^{-B} \tag{112}$$

We then have

$$\mathrm{E}[B_1^*] \approx \frac{1}{\mathrm{E}[\frac{1}{y_i}]} \tag{113}$$

Now (112) and (105) give

$$\mathrm{E}\Big[\frac{1}{y_i}\Big] \approx \sum_{k=1}^{\infty} \frac{1}{k} \frac{B^k}{k\,!} e^{-B} \tag{114}$$

This sum can be evaluated with the result

$$\mathrm{E}\Big[\frac{1}{y_i}\Big] \approx e^{-B}(\mathrm{Ei}(B) - \gamma - \log B) \tag{115}$$

where Ei is the exponential integral and γ is Euler's constant. Since B was assumed fairly large, we can use the asymptotic expansion

$$\mathrm{Ei}(B) \sim \frac{e^B}{B}\Big\{1 + \sum_{r=1}^{\infty} \frac{r\,!}{B^r}\Big\}, \quad B \to +\infty \tag{116}$$

We shall use the first-order approximand, and discarding γ and $\log B$ in (115) we then obtain

$$\mathrm{E}\Big[\frac{1}{y_i}\Big] \approx \frac{1}{B}\Big(1 + \frac{1}{B}\Big) \tag{117}$$

Finally (113) gives the approximate result (110) that the bias is -1.

An approximate removal of the background bias can be accomplished by *weight smoothing* which can be done in several ways. Earlier PALS*fit* versions carried out an extra iteration cycle after the first cycle had converged. (This should be done anyway in case of source correction in *PositronFit*.) Between the two iteration cycles the weight fluctuations were removed by replacing (103) by the weights

$$w_i = \frac{1}{y_i^{\mathrm{f}}} \tag{118}$$

where y_i^{f} is the model-predicted y_i-estimate at the end of the first cycle. This procedure did in fact remove the bias, but also has some drawbacks. When the fit is not perfect, the smoothing not only removes the fluctuations but also distorts the overall shape of the weight function, which results in an unreliable statistical analysis.

The present PALS*fit* version uses instead a heuristic *non-parametric* (i.e. model-independent) weight smoothing, once and for all at the beginning of the analysis. Guided by log-plots of various typical spectra $y(t)$, we make a continuous piecewise linear fit of $\log_{10} y(t)$, in which the knot abscissas and knot values are the fitting parameters. Our algorithm uses separable least squares optimization where the inner minimisation determines the ordinates for given knot abscissas, while the outer minimisation furnishes the optimal position of the knot abscissas. The latter is controlled by an "amoeba type" procedure (Nelder and Mead, in Numerical Recipes [43]). We obtain good results by dividing the spectrum in at most 4 parts and using the piecewise linear algorithm on each of these. There will be fewer than 4 parts when, for example, the peak position is outside the interval from $i_{\mathrm{ch}}^{\mathrm{min}}$ to $i_{\mathrm{ch}}^{\mathrm{max}}$, cf. (12). The number of segments used in each part is normally 4 but less when there are few channels in a part. Common fit values at contiguous parts are obtained by a simple averaging. This ensures an overall continuity of the fit. When the minimum value of the counts within some part exceeds a critical value, say 1000, we abandon the smoothing there, since the spectrum within that part may already be considered smooth. From the computed fit values we finally obtain the smoothed weights w_i by taking antilogarithm and reciprocal.

The new way of estimating the weights has very little influence on the values of the fitted parameters. Typically the parameters change by an amount which is an order of magnitude smaller than the parameter standard deviations, or less. The main effect is that the program has become more robust against some extreme choices of input parameters to the analysis. Moreover the reliability of the statistical analysis for imperfect fits is improved.

4.7.2 Implementation of linear constraints

We consider the constrained linear least-squares problem (cf. (38–39) and (35)),

$$\phi_{\min} = \min_{\boldsymbol{\alpha}\in\mathbb{R}^p} \{\|\mathbf{W}^{1/2}(\boldsymbol{y} - \mathbf{P}\boldsymbol{\alpha})\|^2 : \mathbf{H}\boldsymbol{\alpha} = \boldsymbol{\gamma}\} \tag{119}$$

This subproblem is part of the separable NLLS method discussed in Section 4.6, where an optimal linear parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^p$ was computed for a given nonlinear parameter vector $\boldsymbol{\beta} \in \mathbb{R}^q$. Thus the derivative matrix $\mathbf{P} = (p_{ij}) = (\partial f_i/\partial\alpha_j)$ is here of size $n \times p$. One way of handling this constraints problem would be to use Lagrange multipliers. This method was used in early predecessors of PALS*fit*. As a result, the normal equation system (36) was extended to a block matrix system:

$$\begin{pmatrix} \mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P} & \mathbf{H}^{\mathrm{T}} \\ \mathbf{H} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \mathbf{P}^{\mathrm{T}}\mathbf{W}\boldsymbol{y} \\ \boldsymbol{\gamma} \end{pmatrix} \tag{120}$$

where the vector $2\boldsymbol{\mu}$ contains the Lagrange multipliers. Although (120) is simple enough, there are some drawbacks in this procedure. The coefficient matrix in (120) is not positive definite as was $\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}$. This could reduce the numerical stability of the calculations. We also note that the constraints increase the size of the "effective normal equation system" from $p \times p$ to $(p + m) \times (p + m)$. Below we describe an elimination method which is now in use in PALS*fit*. It offers better stability, reduced computer time, and reduced storage demand. Since the rank of $\mathbf{H}$ is m, we can construct a nonsingular matrix by picking m independent columns from $\mathbf{H}$. A suitable permutation will bring these columns to the m first positions. This can be expressed in terms of a pth-order permutation matrix $\boldsymbol{\Pi}$ by

$$\mathbf{H}\boldsymbol{\Pi} = \mathbf{H}' = \begin{matrix} \\ m \end{matrix} \begin{matrix} \begin{matrix} m & p-m \end{matrix} \\ \begin{pmatrix} \mathbf{B} & \mathbf{N} \end{pmatrix} \end{matrix} \tag{121}$$

In the language of linear programming we call $\mathbf{B}$ a "basis matrix" for $\mathbf{H}$, whereas the columns in $\mathbf{N}$ are called "nonbasic". Because $\boldsymbol{\Pi}$ is orthogonal, $\boldsymbol{\Pi}\boldsymbol{\Pi}^{\mathrm{T}} = \mathbf{I}_p$, (39) can be written

$$\mathbf{H}'\boldsymbol{\alpha}' = \boldsymbol{\gamma} \tag{122}$$

with

$$\boldsymbol{\alpha}' = \boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\alpha} \tag{123}$$

Equation (122) has the complete solution

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha}'_0 + \mathbf{Y}'\boldsymbol{t} \tag{124}$$

where

$$\boldsymbol{\alpha}'_0 = \begin{matrix} m \\ p-m \end{matrix} \begin{pmatrix} \mathbf{B}^{-1}\boldsymbol{\gamma} \\ \mathbf{0} \end{pmatrix} \tag{125}$$

$$\mathbf{Y}' = \begin{matrix} \\ m \\ p-m \end{matrix} \begin{matrix} \begin{matrix} p-m \end{matrix} \\ \begin{pmatrix} -\mathbf{B}^{-1}\mathbf{N} \\ \mathbf{I}_{p-m} \end{pmatrix} \end{matrix} \tag{126}$$

and $\boldsymbol{t} \in \mathbb{R}^{p-m}$. From (123) we get the complete solution of (39):

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_0 + \mathbf{Y}\boldsymbol{t}, \qquad \boldsymbol{t} \in \mathbb{R}^{p-m} \tag{127}$$

where

$$\boldsymbol{\alpha}_0 = \boldsymbol{\Pi}\boldsymbol{\alpha}'_0 \tag{128}$$

and

$$\mathbf{Y} = \boldsymbol{\Pi}\mathbf{Y}' \tag{129}$$

It is practical to partition $\boldsymbol{\Pi}$ in column sections as follows:

$$\boldsymbol{\Pi} = \begin{matrix} \\ p \end{matrix} \begin{matrix} \begin{matrix} m & p-m \end{matrix} \\ \begin{pmatrix} \boldsymbol{\Pi}_1 & \boldsymbol{\Pi}_2 \end{pmatrix} \end{matrix} \tag{130}$$

Then (128) becomes

$$\boldsymbol{\alpha}_0 = \boldsymbol{\Pi}_1 \mathbf{B}^{-1} \boldsymbol{\gamma} \tag{131}$$

To express (129) we note that

$$\mathbf{N} = \mathbf{H}\boldsymbol{\Pi}_2 \tag{132}$$

and so

$$\mathbf{Y} = (\mathbf{I}_p - \boldsymbol{\Pi}_1 \mathbf{B}^{-1} \mathbf{H})\boldsymbol{\Pi}_2 \tag{133}$$

We have

$$\mathbf{H}\mathbf{Y} = \mathbf{H}'\mathbf{Y}' = \mathbf{0} \tag{134}$$

and the columns of $\mathbf{Y}$ form a basis of the null space or kernel of $\mathbf{H}$. Using (127) we can reformulate the constrained p-dimensional problem (119) to an unconstrained $(p-m)$-dimensional problem:

$$\phi_{\min} = \min_{\boldsymbol{t} \in \mathbb{R}^{p-m}} \{\|\mathbf{W}^{1/2}(\boldsymbol{y} - \mathbf{P}\boldsymbol{\alpha}_0 - \mathbf{P}\mathbf{Y}\boldsymbol{t})\|^2\} \tag{135}$$

We see that this can be derived from (35) by substituting $p-m$ for k, $\boldsymbol{y} - \mathbf{P}\boldsymbol{\alpha}_0$ for $\boldsymbol{y}$, $\mathbf{P}\mathbf{Y}$ for $\mathbf{P}$, and $\boldsymbol{t}$ for $\boldsymbol{b}$. Thus we can immediately write down the normal equation system for (135) by making the corresponding substitutions in (36):

$$(\mathbf{P}\mathbf{Y})^{\mathrm{T}} \mathbf{W} (\mathbf{P}\mathbf{Y})\, \boldsymbol{t} = (\mathbf{P}\mathbf{Y})^{\mathrm{T}} \mathbf{W} (\boldsymbol{y} - \mathbf{P}\boldsymbol{\alpha}_0) \tag{136}$$

Next we shall derive an expression for the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{b})$ of the total parameter vector $\boldsymbol{b} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ when the constraints (38) or (39) are included. Recalling that $\boldsymbol{\Sigma}(\boldsymbol{b})$ is independent of the actual fitting method, we can estimate it by perturbing the solution vector $\boldsymbol{b}$ at the end of iterations. From the normal equation system (136) we deduce in analogy with (42–53) that

$$\boldsymbol{\Sigma}(\boldsymbol{t}) = \{(\mathbf{P}_\alpha \mathbf{Y})^{\mathrm{T}} \mathbf{W} (\mathbf{P}_\alpha \mathbf{Y})\}^{-1} \tag{137}$$

We have written $\mathbf{P}_\alpha$ for $\mathbf{P}$ since we shall now reserve the notation $\mathbf{P}$ for the $n \times k$ matrix containing derivatives $\partial f_i / \partial b_j$ with respect to *all* the $p+q$ components of $\boldsymbol{b} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$. Thus we shall write

$$\mathbf{P} = \begin{array}{c} \\ n \end{array} \begin{array}{c} \begin{array}{cc} p & q \end{array} \\ \begin{pmatrix} \mathbf{P}_\alpha & \mathbf{P}_\beta \end{pmatrix} \end{array} \tag{138}$$

We note that

$$\begin{array}{c} \\ n \end{array} \begin{array}{c} \begin{array}{cc} p-m & q \end{array} \\ \begin{pmatrix} \mathbf{P}_\alpha \mathbf{Y} & \mathbf{P}_\beta \end{pmatrix} \end{array} = \mathbf{P}\mathbf{Z} \tag{139}$$

where $\mathbf{Z}$ is given by

$$\mathbf{Z} = \begin{array}{c} \\ p \\ q \end{array} \begin{array}{c} \begin{array}{cc} p-m & q \end{array} \\ \begin{pmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_q} \end{pmatrix} \end{array} \tag{140}$$

This means that (137) can be extended from $\boldsymbol{t}$ to $(\boldsymbol{t}, \boldsymbol{\beta})$ as follows:

$$\boldsymbol{\Sigma}(\boldsymbol{t}, \boldsymbol{\beta}) = \{(\mathbf{P}\mathbf{Z})^{\mathrm{T}} \mathbf{W} (\mathbf{P}\mathbf{Z})\}^{-1} \tag{141}$$

Furthermore, since

$$\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha}_0 \\ \mathbf{0} \end{pmatrix} + \mathbf{Z} \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{\beta} \end{pmatrix} \tag{142}$$

we obtain the result

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = \mathbf{Z}\{(\mathbf{P}\mathbf{Z})^{\mathrm{T}} \mathbf{W} (\mathbf{P}\mathbf{Z})\}^{-1} \mathbf{Z}^{\mathrm{T}} \tag{143}$$

4.7.3 Distribution of $\phi_{\min}$

With $\boldsymbol{b}_0 = (b_{j0})$ defined in (41) and $\boldsymbol{b} = (b_j)$ being the solution of the normal equations (28), we obtain the following approximate expression of $\phi_{\min}$ from the linear expansion (42), which is valid for small $\Delta\boldsymbol{b} = (b_j - b_{j0})$:

$$\phi_{\min} = \sum_{i=1}^{n} w_i \Big(y_i - \eta_i - \sum_{j=1}^{k} p_{ij}(b_j - b_{j0}) \Big)^2 \tag{144}$$

This can also be written

$$\phi_{\min} = \|\mathbf{W}^{1/2}(\Delta\boldsymbol{y} - \mathbf{P}\Delta\boldsymbol{b})\|^2 = (\Delta\boldsymbol{y} - \mathbf{P}\Delta\boldsymbol{b})^{\mathrm{T}}\mathbf{W}(\Delta\boldsymbol{y} - \mathbf{P}\Delta\boldsymbol{b}) \tag{145}$$

with $\Delta\boldsymbol{y} = (y_i - \eta_i)$ and $\mathbf{P}$ given by (29–30). By (45–46) $\phi_{\min}$ becomes a quadratic form in $\Delta\boldsymbol{y}$:

$$\phi_{\min} = \Delta\boldsymbol{y}^{\mathrm{T}}\mathbf{B}\,\Delta\boldsymbol{y} \tag{146}$$

where $\mathbf{B}$ is found to

$$\mathbf{B} = \mathbf{W} - \mathbf{W}\mathbf{P}(\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{W} \tag{147}$$

Defining $u_i = \Delta y_i / s_i$, we see that u_i becomes a standardized normal variable,

$$u_i \in \mathrm{N}(0,1) \tag{148}$$

Then $\phi_{\min}$ can be expressed as a quadratic form in $\boldsymbol{u} = (u_i)$:

$$\phi_{\min} = \boldsymbol{u}^{\mathrm{T}}\mathbf{C}\boldsymbol{u} \tag{149}$$

with

$$\mathbf{C} = \mathbf{W}^{-1/2}\mathbf{B}\mathbf{W}^{-1/2} = \mathbf{I}_n - \mathbf{M} \tag{150}$$

where

$$\mathbf{M} = \mathbf{W}^{\frac{1}{2}}\mathbf{P}(\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{W}^{\frac{1}{2}} \tag{151}$$

Clearly the matrix $\mathbf{M}$ is of rank k and all its nonzero eigenvalues are unity, as is easily verified by premultiplying $\mathbf{M}\boldsymbol{x} = \lambda\boldsymbol{x}$ by $\mathbf{P}^{\mathrm{T}}\mathbf{W}^{1/2}$. Hence there is an orthogonal substitution $\boldsymbol{u} = \mathbf{Q}\boldsymbol{z}$ which transforms $\phi_{\min}$ into a sum of $f = n - k$ squares:

$$\phi_{\min} = \sum_{i=1}^{n-k} z_i^2 \tag{152}$$

where the z_i are independent, and each $z_i \in \mathrm{N}(0,1)$. This means that $\phi_{\min}$ has a χ^2-distribution with f degrees of freedom. If there are m independent linear constraints on the parameters, then the expression (135) demonstrates that the number of degrees of freedom is altered to $f = n - (k - m) = n - k_{\mathrm{free}}$.

4.7.4 Proof of parameter shift formula

We shall here give a proof of the formula (66). In the following we consider $\phi(\boldsymbol{b})$ with fixed spectrum (y_i). We fix Δb_1 and seek the conditional minimum when the other parameters vary. We shall use the expansion formula (73):

$$\phi(\boldsymbol{b} + \Delta\boldsymbol{b}) = \phi(\boldsymbol{b}) + \nabla\phi(\boldsymbol{b}) \cdot \Delta\boldsymbol{b} + \Delta\boldsymbol{b}^T\mathbf{P}^T\mathbf{W}\mathbf{P}\Delta\boldsymbol{b} \tag{153}$$

We introduce the vector $\boldsymbol{z}$ with components $\Delta b_2, \ldots, \Delta b_k$. The gradient term in (153) can be written

$$\nabla\phi(\boldsymbol{b}) \cdot \Delta\boldsymbol{b} = \frac{\partial\phi}{\partial b_1}\Delta b_1 + \nabla_z\phi(\boldsymbol{b}) \cdot \boldsymbol{z} \tag{154}$$

where $\nabla_z\phi(\boldsymbol{b})$ must be zero. Making the partition

$$\mathbf{P}^T\mathbf{W}\mathbf{P} = \begin{pmatrix} a_{11} & \boldsymbol{d}^{\mathrm{T}} \\ \boldsymbol{d} & \mathbf{C} \end{pmatrix} \tag{155}$$

(153) can then be written

$$\phi(\boldsymbol{b}+\Delta\boldsymbol{b}) = \phi(\boldsymbol{b}) + \frac{\partial\phi}{\partial b_1}\Delta b_1 + a_{11}\Delta b_1^2 + 2\Delta b_1 \boldsymbol{z}^{\mathrm{T}}\boldsymbol{d} + \boldsymbol{z}^T\mathbf{C}\boldsymbol{z} \tag{156}$$

To minimize (156) we take the derivative with respect to $\boldsymbol{z}$. After equating the result to zero we deduce that

$$\boldsymbol{z} = -\Delta b_1 \mathbf{C}^{-1}\boldsymbol{d} \tag{157}$$

Next we make the similar partitioning

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = \begin{pmatrix} \sigma_{11} & \boldsymbol{s}^{\mathrm{T}} \\ \boldsymbol{s} & \boldsymbol{\Gamma} \end{pmatrix} \tag{158}$$

We assume statistical weighting which implies the identity $\mathbf{P}^T\mathbf{W}\mathbf{P}\boldsymbol{\Sigma}(\boldsymbol{b}) = \mathbf{I}_k$, cf. (53). From this we infer that

$$\mathbf{C}^{-1}\boldsymbol{d} = -\frac{1}{\sigma_{11}}\boldsymbol{s} \tag{159}$$

Inserting this in (157) yields

$$\boldsymbol{z} = \frac{\Delta b_1}{\sigma_{11}}\boldsymbol{s} \tag{160}$$

which proves formula (66) since $\boldsymbol{s}$ has components $\sigma_{21}, \ldots, \sigma_{k1}$.

4.7.5 Scaling in separable NLLS

In order to facilitate the application of standard minimization software we found it appropriate to make a scaling of the problem formulation. We recast the original minimization problem (25) to

$$\phi_{\min} = \min_{\boldsymbol{b}\in\mathbb{R}^k}\{\|\boldsymbol{r}(\boldsymbol{b})\|^2\} \tag{161}$$

where $\boldsymbol{r}(\boldsymbol{b})$ is a (scaled) *residual vector* with components

$$r_i = w_i^{1/2}(y_i - f_i(\boldsymbol{b})) \tag{162}$$

This induces a number of vector and matrix transformations containing the matrix scaling factor $\mathbf{W}^{1/2}$:

$$\boldsymbol{z} = \mathbf{W}^{1/2}\boldsymbol{y} \tag{163}$$

$$\boldsymbol{e} = \mathbf{W}^{1/2}\boldsymbol{f} \tag{164}$$

$$\mathbf{E} = \mathbf{W}^{1/2}\mathbf{F} \tag{165}$$

$$\mathbf{G} = \mathbf{W}^{1/2}\mathbf{P} \tag{166}$$

Then the counterparts of (91–95) become:

$$\boldsymbol{e}(\boldsymbol{b}) = \mathbf{E}(\boldsymbol{\beta})\boldsymbol{\alpha} \tag{167}$$

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^p}\{\|\boldsymbol{z} - \mathbf{E}(\boldsymbol{\beta})\boldsymbol{\alpha}\|^2 : \boldsymbol{\beta} \text{ fixed}\} \tag{168}$$

$$\mathbf{E}^{\mathrm{T}}\mathbf{E}\boldsymbol{\alpha} = \mathbf{E}^{\mathrm{T}}\boldsymbol{z} \tag{169}$$

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^q}\{\phi(\boldsymbol{\beta}) \equiv \|\boldsymbol{z} - \mathbf{E}(\boldsymbol{\beta})\boldsymbol{\alpha}(\boldsymbol{\beta})\|^2\} \tag{170}$$

$$(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{D})\boldsymbol{d} = \mathbf{G}^{\mathrm{T}}(\boldsymbol{z} - \mathbf{E}\boldsymbol{\alpha}) \tag{171}$$

Moreover (97–98) are replaced by:

$$\frac{\partial \boldsymbol{e}}{\partial\beta_j} = \frac{\partial\mathbf{E}}{\partial\beta_j}\boldsymbol{\alpha} + \mathbf{E}\frac{\partial\boldsymbol{\alpha}}{\partial\beta_j} \tag{172}$$

$$\mathbf{E}^{\mathrm{T}}\mathbf{E}\frac{\partial\boldsymbol{\alpha}}{\partial\beta_j} = \frac{\partial\mathbf{E}^{\mathrm{T}}}{\partial\beta_j}(\boldsymbol{z} - \mathbf{E}\boldsymbol{\alpha}) - \mathbf{E}^{\mathrm{T}}\frac{\partial\mathbf{E}}{\partial\beta_j}\boldsymbol{\alpha} \tag{173}$$

We see that the effect of these transformations is to "hide" the weights w_i entirely. To include the effect of the linear constraints on (172–173) we use (127) and find:

$$\frac{\partial \boldsymbol{e}}{\partial \beta_j} = \frac{\partial \mathbf{E}}{\partial \beta_j}\boldsymbol{\alpha} + \mathbf{E}\mathbf{Y}\frac{\partial \boldsymbol{t}}{\partial \beta_j} \tag{174}$$

$$(\mathbf{E}\mathbf{Y})^{\mathrm{T}}\mathbf{E}\mathbf{Y}\frac{\partial \boldsymbol{t}}{\partial \beta_j} = \frac{\partial (\mathbf{E}\mathbf{Y})^{\mathrm{T}}}{\partial \beta_j}(\boldsymbol{z} - \mathbf{E}\boldsymbol{\alpha}) - (\mathbf{E}\mathbf{Y})^{T}\frac{\partial \mathbf{E}}{\partial \beta_j}\boldsymbol{\alpha} \tag{175}$$

4.7.6 QR decomposition

A direct solution of normal equations, even by Choleski decomposition, may present numerical difficulties inherent with the ill-conditioning of the positive-definite coefficient matrix, say $\mathbf{E}^{\mathrm{T}}\mathbf{E}$ in (169). Instead we use a procedure based on the so-called QR decomposition of $\mathbf{E}$, viz.

$$\mathbf{E} = \mathbf{Q}\mathbf{R} \tag{176}$$

where $\mathbf{Q}$ is an $n \times p$ matrix with orthonormal columns and $\mathbf{R}$ is a $p \times p$ upper triangular matrix (see, e.g., Chapter 9 in [42]). Using (176) the system (169) is reformulated to $\mathbf{R}\boldsymbol{\alpha} = \mathbf{Q}^{\mathrm{T}}\boldsymbol{z}$, which can be easily solved by back-substitution. The same procedure is used when solving (173) for $\partial\boldsymbol{\alpha}/\partial\beta_j$, with $\mathbf{R}$ being saved after the solution of (169).

The numerical computation of the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{b})$ can also be done by QR technique. We start with the expression (143) which includes the constraints and performs the usual scaling by defining $\mathbf{T} = \mathbf{W}^{1/2}\mathbf{P}$. Then we obtain

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = \mathbf{Z}((\mathbf{T}\mathbf{Z})^{\mathrm{T}}\mathbf{T}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}} \tag{177}$$

and can now make the decomposition (of course the factors are different from those in (176)),

$$\mathbf{T}\mathbf{Z} = \mathbf{Q}\mathbf{R} \tag{178}$$

which leads to

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = \mathbf{Z}\mathbf{R}^{-1}(\mathbf{Z}\mathbf{R}^{-1})^{\mathrm{T}} \tag{179}$$

In some ill-conditioned problems the diagonal row of $\mathbf{R}$ may contain very small elements, which would render the evaluation of $\boldsymbol{\Sigma}(\boldsymbol{b})$ by (179) completely erratic. There exists a variant of the QR decomposition with column scaling and pivoting that admits a judicious discarding rule for insignificant elements in the $\mathbf{R}$-diagonal [42, 35]. Following this idea, we shall replace (178) with

$$\mathbf{T}\mathbf{Z}\boldsymbol{\Lambda}\boldsymbol{\Pi} = \mathbf{Q}\mathbf{R} \tag{180}$$

where $\boldsymbol{\Lambda}$ is a diagonal scaling matrix, $\boldsymbol{\Pi}$ a permutation matrix, and the diagonal elements of $\mathbf{R}$ are in non-increasing order of magnitude. The entries in $\boldsymbol{\Lambda}$ are chosen as the inverse Euclidean norms of the column vectors of $\mathbf{T}\mathbf{Z}$ and might be called "uncoupled standard deviations". Instead of (179) we obtain

$$\boldsymbol{\Sigma}(\boldsymbol{b}) = \mathbf{Z}\boldsymbol{\Lambda}\boldsymbol{\Pi}\mathbf{R}^{-1}(\mathbf{Z}\boldsymbol{\Lambda}\boldsymbol{\Pi}\mathbf{R}^{-1})^{\mathrm{T}} \tag{181}$$

The expression (181) is only used for the "significant" parameters which corresponds to the upper part of $\mathbf{R}$. The variance of the "insignificant" parameters are estimated by their uncoupled standard deviations, while the covariance calculation for such parameters are abandonned. (It is easier to grasp the essential features of this procedure if we simplify and replace $\mathbf{T}\mathbf{Z}$ with $\mathbf{P} = \mathbf{Q}\mathbf{R}$.)

5 Mathematical model details

In Sections 2.4 and 2.6 we gave a short presentation of the theoretical models used in *PositronFit* and *ResolutionFit*. Below we shall try to fill the gap between the rather brief description given there of the underlying mathematical models, and the least-squares theory in Chapter 4.

5.1 *PositronFit*

Writing formula (3) as

$$f(t) = \sum_{j=1}^{k_0} \sum_{p=1}^{k_g} \omega_p (a_j * G_p)(t) \; + \; B \tag{182}$$

we must evaluate the convolution integral

$$(a_j * G_p)(t) = \int_{-\infty}^{\infty} a_j(v) G_p(t-v) \mathrm{d}v \tag{183}$$

where a_j and G_p were defined in (4) and (7), respectively. Henceforward, we prefer to describe the decay of a lifetime component in terms of the annihilation rate

$$\lambda_j = 1/\tau_j \tag{184}$$

instead of the lifetime τ_j itself. It can be shown that

$$(a_j * G_p)(t) = \frac{1}{2} A_j \phi(t - T_0 - \Delta t_p, \lambda_j, \sigma_p) \tag{185}$$

Here the function ϕ acts as a building block for our lifetime spectral model and is defined by

$$\phi(u, \lambda, \sigma) = \exp\Big(-\lambda u + \frac{1}{2}\lambda^2\sigma^2\Big) \operatorname{erfc}\Big(\frac{\lambda\sigma^2 - u}{\sqrt{2}\sigma}\Big) \tag{186}$$

where erfc stands for the complementary error function

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) \tag{187}$$

and erf in turn is defined by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) \mathrm{d}t \tag{188}$$

Regarding the practical computation of (186), that formula may from a numerical point of view be dangerous to use as it stands. The difficulty arises when $u \ll 0$. Then $\phi(u, \lambda, \sigma)$ itself is small; nevertheless, the first factor of (186) is large and may cause an overflow on the computer. At the same time, the second factor is very small and likely to underflow. A better alternative is to restate (186) as

$$\phi(u, \lambda, \sigma) = \exp\Big(-\frac{u^2}{2\sigma^2}\Big) \operatorname{eerfc}\Big(\frac{\lambda\sigma^2 - u}{\sqrt{2}\sigma}\Big) \tag{189}$$

where eerfc stands for the normalized complementary error function

$$\operatorname{eerfc}(x) = \exp(x^2) \operatorname{erfc}(x) \tag{190}$$

It is not hard to develop robust and accurate numerical approximations for this slowly varying function, which is decreasing when $x > 0$ and behaves asymptotically as $1/(\sqrt{\pi}x)$. When x is negative we use the formula

$$\operatorname{eerfc}(-x) = 2\exp(x^2) - \operatorname{eerfc}(x) \tag{191}$$

Inserting (185) in (182) we get

$$f(t) = \frac{1}{2}\sum_{j=1}^{k_0} A_j \sum_{p=1}^{k_g} \omega_p \phi(t - T_0 - \Delta_p, \lambda_j, \sigma_p) \;+\; B \tag{192}$$

Finally, we compute the integrated model-predicted count f_i defined by equation (11) in Section 2.4. We use the identity

$$\int \phi(u,\lambda,\sigma)\, du = -\frac{1}{\lambda}\Big(\phi(u,\lambda,\sigma) + \mathrm{erfc}\,\Big(\frac{u}{\sqrt{2}\sigma}\Big)\Big) \tag{193}$$

and obtain

$$f_i = \sum_{j=1}^{k_0} F_{ij} + B = \sum_{j=1}^{k_0} \alpha_j f_{ij} + B \tag{194}$$

where

$$\alpha_j = \tfrac{1}{2} A_j/\lambda_j = \tfrac{1}{2} A_j \tau_j \tag{195}$$

is half the absolute intensity of lifetime component j,

$$f_{ij} = \sum_{p=1}^{k_g} \omega_p \Big\{ \phi(t_{i-1,p}, \lambda_j, \sigma_p) - \phi(t_{ip}, \lambda_j, \sigma_p) + \mathrm{erfc}\,\Big(\frac{t_{i-1,p}}{\sqrt{2}\sigma_p}\Big) - \mathrm{erfc}\,\Big(\frac{t_{ip}}{\sqrt{2}\sigma_p}\Big)\Big\} \tag{196}$$

and where we use the shorthand notation

$$t_{ip} = i_{\mathrm{ch}}^{\min} + i - 1 - T_0 - \Delta_p \tag{197}$$

By now we have arrived at the model expression $f_i = f_i(\boldsymbol{b})$ entering the least-squares formulation of the fitting problem given in Chapter 4. We also see that (194) is separable as required; the parameter vector $\boldsymbol{b}$ splits into a "linear" parameter $\boldsymbol{\alpha}$ and a "nonlinear" one $\boldsymbol{\beta}$ given by

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{k_0}, B) \tag{198}$$

and

$$\boldsymbol{\beta} = (\lambda_1, \ldots, \lambda_{k_0}, T_0) \tag{199}$$

Thus the separable fitting theory of Section 4.6 applies. To perform the computations outlined there, we must evaluate the derivatives of f_{ij} in (196) with respect to λ_j and T_0; this job is facilitated by the following two formulas:

$$\frac{\partial \phi}{\partial u} = -\lambda\phi(u,\lambda,\sigma) + \sqrt{\frac{2}{\pi}}\frac{1}{\sigma}\exp\Big(-\frac{u^2}{2\sigma^2}\Big) \tag{200}$$

and

$$\frac{\partial \phi}{\partial \lambda} = (\lambda\sigma^2 - u)\phi(u,\lambda,\sigma) - \sqrt{\frac{2}{\pi}}\sigma\exp\Big(-\frac{u^2}{2\sigma^2}\Big) \tag{201}$$

We introduce the abbreviations

$$\phi_{ijp} = \phi(t_{ip}, \lambda_j, \sigma_p) \tag{202}$$

$$\delta\phi_{ijp} = \phi_{ijp} - \phi_{i-1,jp} \tag{203}$$

$$\delta_{ip}^{\exp} = \exp\Big(-\frac{t_{ip}^2}{2\sigma_p^2}\Big) - \exp\Big(-\frac{t_{i-1,p}^2}{2\sigma_p^2}\Big) \tag{204}$$

and then obtain from (200–201)

$$\frac{\partial f_{ij}}{\partial T_0} = -\lambda_j \sum_{p=1}^{k_g} \omega_p \delta\phi_{ijp} \tag{205}$$

$$\frac{\partial f_{ij}}{\partial \lambda_j} = \sum_{p=1}^{k_g} \omega_p \{\phi_{i-1,jp} - (\lambda_j\sigma_p^2 - t_{ip})\delta\phi_{ijp} + \sigma_p\sqrt{2/\pi}\,\delta_{ip}^{\exp}\} \tag{206}$$

In Section 2.4 we mentioned the types of constraints which could be imposed on the parameters in *PositronFit*. Those constraints that fix one of the "primary" fitting parameters listed in (198) and (199) are realized by deleting the corresponding components from $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$. This may apply to B, λ_j, and T_0. Constraints of the type "fixed relative intensity" are not of this simple type because the relative intensities $\alpha_j / \sum \alpha_{j'}$ are not primary parameters. But obviously such constraints are expressible as linear constraints on the linear parameters α_j, i.e. relations of the form

$$\sum h_{ij} \alpha_j = \gamma_i \tag{207}$$

where h_{ij} are known coefficients, cf. (38). The same holds good for constraints of the type "a linear combination of the relative intensities = 0", as well as the total-area constraint (18).

5.2 *ResolutionFit*

Although the basic model in *ResolutionFit* is the same as in *PositronFit*, there are certain differences regarding which parameters enter as fitting parameters since the widths σ_p and the shifts Δt_p are fitting parameters in *ResolutionFit*. Hence (199) should be replaced by

$$\boldsymbol{\beta} = (\lambda_1, \ldots, \lambda_{k_0}, T_0, \sigma_1, \ldots, \sigma_{k_g}, \Delta t_1, \ldots, \Delta t_{k_g}) \tag{208}$$

In Section 2.6 we mentioned the types of constraints which could be imposed on the parameters in *ResolutionFit*. Some of the parameters of (208) can be fixed and in that case should be deleted from $\boldsymbol{\beta}$. This may apply to λ_j, σ_p, and Δt_p. Notice that T_0 in *ResolutionFit* is always a free parameter. As a consequence, we must require that at least one of the shifts Δt_p be fixed.

In addition to (200–206) we shall need the following formulas:

$$\frac{\partial \phi}{\partial \sigma} = \lambda^2 \sigma \phi(u, \lambda, \sigma) - \sqrt{\frac{2}{\pi}} \Big(\lambda + \frac{u}{\sigma^2} \Big) \exp\Big(- \frac{u^2}{2\sigma^2} \Big) \tag{209}$$

$$\frac{\partial f_{ij}}{\partial \Delta_p} = -\lambda_j \omega_p \delta \phi_{ijp} \tag{210}$$

$$\frac{\partial f_{ij}}{\partial \sigma_p} = \omega_p \{ -\lambda_j^2 \sigma_p \delta \phi_{ijp} + \lambda_j \sqrt{2/\pi}\, \delta_{ip}^{\exp} \} \tag{211}$$

In *ResolutionFit* we compute shape parameters for the fitted resolution curve. This leads to nonlinear equations involving the ϕ function of (186). We use a Newton–Raphson procedure for the numerical solution of these.

6 Control files

When running PALS*fit* the program produces a control file, which contains all the input instructions for either a *PositronFit* or a *ResolutionFit* analysis. Normally the editing of this file is a task which is done automatically by using the PALS*fit* menus. Nevertheless there might be situations where an inspection or an external editing of the file is appropriate.

In Chapter 1 it was mentioned that in certain situations (batch processing or running under Linux) it may be useful to run the command-driven PATFIT programs *PositronFit* and *ResolutionFit* directly. In that case you will also need to know the structure of the input files [2]. Note that PATFIT and PALS*fit* are input-compatible.

Anyway, the knowledge of the structure of the control files may give the user a good overview of the capabilities of PALS*fit*. Therefore, in the following we shall describe the contents of

the control files for PositronFit and ResolutionFit in some detail. For convenience, we will in some cases use parameter names that occur in the programs.

The control file is composed of one or more control data sets. A control data set is partitioned into a number of data blocks, corresponding roughly to the menus in PALS*fit*. Each block is initiated by a so-called block header. For example, the first block header reads

```
POSITRONFIT DATA BLOCK: OUTPUT OPTIONS
```

in the case of *PositronFit*, and similarly for *ResolutionFit*.

6.1 *PositronFit* control file

A sample PALS*fit* control file for *PositronFit* with a single control data set is shown below.

```
POSITRONFIT DATA BLOCK: OUTPUT OPTIONS
0000
POSITRONFIT DATA BLOCK: SPECTRUM
       512
(10i7)
Mol_Crystal spectra.dat
32800 CYCLOOCTANE
 1 0
32800 CYCLOOCTANE
  32800     764     778     728     754     755     733     725     743     769
    700     735     694     737     714     766     756     692     741     728
    705     715     729     716     717     723     699     717     719     702
    696     729     637     710     678     745     702     708     686     708
    708     670     677     712     679     664     642     708     718     720
    677     688     722     704     705     701     651     675     703     720
    696     710     723     683     628     746     725     647     730     703
    677     688     691     700     730     676     658     708     714     690
    719     705     707     708     673     701     684     689     692     697
    743     688     721     646     712     716     665     707     730     690
    706     743     713     682     758     720     683     712     679     700
    692     699     692     708     742     740     742     690     708     727
    770     704     766     683     743     822    1049    1810    4146   11412
  30056   71498  151210  277754  433291  581716  677357  705264  669401  595116
 509460  426352  354314  294562  245828  208141  175625  151785  131162  114887
 100962   90149   80698   73201   66518   60971   56076   52147   48976   45968
  43147   40678   39125   36522   35059   33784   32103   30988   30125   29086
  27905   26888   25935   25090   24441   23366   22792   22072   21308   21178
  20144   19616   18959   18302   17973   17247   16784   16432   15678   15607
  15241   14667   14085   13685   13468   12871   12522   12277   11915   11533
  11407   11174   10807   10156    9894    9835    9590    9237    9142    8789
   8694    8424    8249    7828    7537    7623    7368    7025    6916    6836
   6748    6475    6307    6079    6018    5730    5628    5619    5377    5274
   5190    5019    4829    4747    4560    4529    4541    4336    4194    4083
   4007    3865    3855    3751    3701    3568    3546    3390    3363    3262
   3218    3061    3007    2932    2786    2918    2759    2738    2722    2611
   2631    2524    2551    2533    2363    2437    2350    2220    2214    2157
   2026    2155    2008    2009    1998    2018    1898    1847    1834    1764
   1774    1724    1747    1639    1728    1572    1601    1581    1679    1566
   1512    1501    1486    1477    1480    1359    1404    1406    1335    1397
   1337    1299    1258    1240    1281    1240    1203    1195    1217    1159
   1117    1163    1079    1049    1077    1080    1038    1086    1020    1051
    974     968    1039     996    1001    1018     967     957     964     927
    975     953     885     942     921     915     911     935     859     851
    880     873     945     863     844     827     852     806     822     838
    830     802     825     825     810     774     772     784     796     801
    803     799     815     760     796     778     768     752     733     760
    719     744     765     786     729     717     719     771     688     730
    744     742     672     733     756     719     689     749     686     754
    686     693     716     830     716     705     666     743     753     734
    710     715     737     705     670     712     650     758     705     764
    778     746     688     728     662     716     707     697     733     660
    687     768     704     690     644     730     692     691     694     675
```

```
    697    695    678    675    710    685    767    711    642    683
    723    702    732    703    678    686    694    693    745    671
    701    699    676    671    659    691    672    669    677    682
    701    752    664    651    620    670    717    708    616    700
    655    706    791    652    665    670    726    722    713    682
    662    650    651    712    680    694    721    676    677    671
    694    701    660    696    711    715    682    635    705    715
    674    676    680    681    734    737    721    699    675    717
    694    667
POSITRONFIT DATA BLOCK: CHANNEL RANGES. TIME SCALE. TIME-ZERO.
        35
       512
       136
       501
  0.077300
G
   136.300
POSITRONFIT DATA BLOCK: RESOLUTION FUNCTION
         3
    0.5395    0.3539    0.4036
    20.000    20.000    60.000
    0.0412   -0.0749    0.0000
POSITRONFIT DATA BLOCK: LIFETIMES AND INTENSITY CONSTRAINTS
         3
FGG
    0.2300    0.4000    2.6500
         1
         3
   28.0000
POSITRONFIT DATA BLOCK: BACKGROUND CONSTRAINTS
         2
   680.0000
POSITRONFIT DATA BLOCK: AREA CONSTRAINTS
         1
        35
       512
POSITRONFIT DATA BLOCK: SOURCE CORRECTION
         2
    0.3840    0.9056
    8.0000    0.4527
  100.0000
         1
         4
GGGG
    0.1200    0.3600    1.2000    2.8000
        -1
   -3.0000    0.0000    1.0000    1.0000
```

Block 1 contains output options. Apart from the block header there is only one record. It contains four integer keys. Each key is either 0 or 1. The value 1 causes some output action to be taken, whereas 0 omits the action. The actions of the 4 keys are:

1. Write input echo to result file
2. Write each iteration output to result file
3. Write residual plot to result file
4. Write correlation matrix to result file

Regardless of the setting of these keys, *PositronFit* always produces the Main Output.

Block 2 contains the spectrum. The first record (after the block header) contains the integer NCH, which is the total number of channels in the spectrum. Next record contains a description of precisely how the spectrum values are "formatted" in the file—expressed as a so-called FORMAT in the programming language FORTRAN [44]; an asterisk $*$ means free format. After this, two text records follow. In the first a name of a spectrum file is given. (Even when INSPEC = 1 (see below) this name should be present, but is in that case not

used by the program.) In the other record an identification label of the spectrum is given. The next record contains two integers, INSPEC and XXX, both taking a value of either 0 or 1. (XXX is specific for the PALS*fit*/GUI and is not used by PATFIT.) INSPEC = 1 means that the spectrum is an intrinsic part of the present control file. In this case the next record should be a text line with a description of the spectrum. The subsequent records are supposed to hold the NCH spectrum values. On the other hand, INSPEC = 0 means that the spectrum is expected to reside in an external file with the spectrum file name entered above. The program tries to open this file (which may contain several spectra) and scans it for a record whose start matches the identification label. After a successful match, the matching (text) line and the spectrum itself is read from the subsequent records in exactly the same way as in the case INSPEC = 1. If XXX = 1 all spectra in the spectrum file will be analysed with the parameters defined in the present data set. If XXX = 0, only the spectrum defined in the present data set will be analysed.

Block 3 contains information related to the measuring system. The first two records (after the block header) contain two channel numbers ICHA1 and ICHA2. These numbers are lower and upper bounds for the definition of a total area range. The next two records contain also two channel numbers ICHMIN and ICHMAX. These define in the same way the channel range which is used in the least-squares analysis. The next record contains the channel width (in ns). The last two records in this block deal with T_0 (time=0 channel number). First comes a constraint flag being either a G or an F. G stands for guessed (i.e. free) T_0, F stands for fixed T_0. The other record contains the initial (guessed or fixed) value of T_0.

Block 4 contains input for definition of the resolution function. The first record (after the block header) contains the number k_g of Gaussian components in the resolution function. Each of the next three records contains k_g numbers. In the first we have the full widths at half maxima of the Gaussians (in ns), FWHM_j, $j = 1, \ldots, k_\mathrm{g}$, in the second their relative intensities (in percent) ω_j, $j = 1, \ldots, k_\mathrm{g}$, and in the third their peak displacements (in ns) Δt_j, $j = 1, \ldots, k_\mathrm{g}$.

Block 5 contains data for the lifetime components in the lifetime spectrum as well as constraints on their relative intensities. The first record (after the block header) holds the number k_0 of lifetime components assumed in the model. Each of the next two records contains k_0 data. In the first we have the constraint flags (G = guessed, F = fixed) for the lifetimes. The other record contains the initial values (guessed or fixed) of the k_0 lifetimes. After this comes a record with an integer m telling the number and type of intensity constraints. $|m|$ is equal to the number of constraints, but m itself may be positive, negative, or zero. If $m = 0$ there is no further input data in this block. If $m > 0$, m of the relative intensities are fixed. In this case the next data item is a pair of records with the numbers j_l, $l = 1, \ldots, m$ and I_{j_l}, $l = 1, \ldots, m$; here j_l is the term number (the succession agreeing with the lifetimes on the previous record) associated with constraint number l, and I_{j_l} is the corresponding fixed relative intensity (in percent). If $m < 0$, $|m|$ linear combinations of the intensities are equal to zero. In this case $|m|$ records follow, each containing the k_0 coefficients h_{lj}, $j = 1, \ldots, k_0$ to the intensities for one of the linear combinations, cf. equation (17) in Section 2.4.

Block 6 contains data related to the background. The first record (after the block header) contains an integer indicator KB, assuming one of the values 0, 1, or 2. KB = 0 means a free background; in this case no more data follows in this block. If KB = 1 the background is fixed to the spectrum average from channel ICHBG1 to channel ICHBG2. These two channel numbers follow on the next two records. If KB = 2, the background is fixed to an input value which is entered on the next record.

Block 7 contains input for constraining the total area. The first record (after the block header) holds an integer indicator KAR, assuming one of the values 0, 1, or 2. KAR =

0 means no area constraint; in this case no more data follows in this block. If KAR > 0, the area between two specified channel limits ICHBEG and ICHEND will be fixed, and these channel numbers follow on the next two records. If KAR $= 1$, the area is fixed to the measured spectrum, and no more input will be needed. If KAR $= 2$ the area is fixed to an input value which is entered on the next record.

Block 8 contains source correction data. The first record (after the block header) contains an integer k_{s} denoting the number of components in the source correction spectrum. $k_{\mathrm{s}} = 0$ means no source correction, in which case the present block contains no more data. The next record contains the lifetimes τ_j^{s}, $j = 1, \ldots, k_{\mathrm{s}}$, and the following the relative intensities I_j^{s}, $j = 1, \ldots, k_{\mathrm{s}}$ for the source correction terms. On the next record is the number α which is the percentage of positrons that annihilate in the source, cf. equation (20) in Section 2.4. Then there follows a record with an integer ISEC. When ISEC $= 0$ the new iteration cycle after the source correction starts from lifetime guesses equal to the converged values from the first (correction-free) cycle. ISEC $= 1$ tells that the second cycle starts from new input data. These 2nd-cycle input data are now entered in exactly the same way as the 1st-cycle data in Block 5. ISEC $= 2$ works as ISEC $= 1$, but with the additional possibility of changing the status of T_0; in this case two more records follow, the first containing the constraint flag (G = guessed, F = fixed) for T_0 and the second the value of T_0.

With the end of Block 8 the entire *PositronFit* control data set is completed. However, as previously mentioned, PALS*fit* accepts multiple control data sets in the same *PositronFit* control file.

6.2 *ResolutionFit* control file

A sample PALS*fit* control file for *ResolutionFit* with a single control data set is shown below.

```
RESOLUTIONFIT DATA BLOCK: OUTPUT OPTIONS
0000
RESOLUTIONFIT DATA BLOCK: SPECTRUM
      1023
(/,(10i7))
.\Metal defects spectra.DAT
39699 CU-ANNEALED
 0 0
RESOLUTIONFIT DATA BLOCK: CHANNEL RANGES. TIME SCALE. TIME-ZERO.
         5
      1000
       140
       500
  0.013400
   172.000
RESOLUTIONFIT DATA BLOCK: RESOLUTION FUNCTION
         3
GGG
    0.2600    0.3000    0.4000
    77.000    19.000     4.000
FGG
    0.0000    0.0223   -0.0462
RESOLUTIONFIT DATA BLOCK: LIFETIMES AND INTENSITY CONSTRAINTS
         3
FFG
    0.1100    0.1800    0.4000
         0
RESOLUTIONFIT DATA BLOCK: BACKGROUND CONSTRAINTS
         1
       650
      1000
```

Block 1 contains output options. It is identical to the corresponding block in the *PositronFit*

control file (but of course the name RESOLUTIONFIT must appear in the block header).

Block 2 contains the spectrum. It is identical to the corresponding block in the *PositronFit* control file.

Block 3 contains information related to the measuring system. The first two records (after the block header) contain two channel numbers ICHA1 and ICHA2. These numbers are lower and upper bounds for the definition of a total area range. The next two records contain also two channel numbers ICHMIN and ICHMAX. These define in the same way the channel range which is used in the least-squares analysis. The next record contains the channel width (in ns). The last record in this block contains the initial (guessed) value of T_0.

Block 4 contains input for definition and initialization of the resolution function. The first record (after the block header) contains the number k_g of Gaussian components in the resolution function. Each of the next two records contains k_g data. In the first we have the constraint flags (G=guessed, F=fixed) for the Gaussian widths. The second contains the initial values (guessed or fixed) of the full widths at half maxima of the Gaussians (in ns), $\mathrm{FWHM}_j^{\mathrm{ini}}$, $j = 1, \ldots, k_\mathrm{g}$. The next record contains the k_g Gaussian component intensities in percent, ω_j, $j = 1, \ldots, k_\mathrm{g}$. The last two records in the block contain again k_g data each. First, we have the constraint flags (G=guessed, F=fixed) for the Gaussian shifts; notice that not all the shifts can be free. Next, we have the initial (guessed or fixed) peak displacements (in ns), Δ_j^{ini}, $j = 1, \ldots, k_\mathrm{g}$.

Block 5 contains data for the lifetime components in the lifetime spectrum as well as constraints on their relative intensities. It is identical to the corresponding block in the *PositronFit* control file.

Block 6 contains data related to the background. It is identical to the corresponding block in the *PositronFit* control file.

This completes the *ResolutionFit* control data set. Multiple data control data sets can be handled in the same way as for *PositronFit*.

6.3 Channel ranges in PALS*fit*

As we saw in the description of the control files for *PositronFit* (P) and *ResolutionFit* (R) in Sections 6.1 and 6.2, the data blocks contain a number of integers defining various kinds of *channel ranges*. Each range is an interval $[M, N]$ and is thus equal to the set of integers i satisfying $M \leq i \leq N$. Using the same program acronyms for the channel limits as before, we can collect all the channel ranges in the following table:

Name	Definition	Program use	Symbol
Total range	[1,NCH]	PR	T
Fit range	[ICHMIN,ICHMAX]	PR	F
Area range	[ICHA1,ICHA2]	PR	A
Background range	[ICHBG1,ICHBG2]	PR	B
Fixed-area range	[ICHBEG,ICHEND]	P	A_f

The first three ranges are always present while the existence of the two latter depends on optional constraints. There are certain restrictions on the ranges T, F, A, B, A_f (when present). All must be nonempty subsets of T, and all the remaining must be subsets of A. Thus in formal terms

$$\emptyset \subset A \subseteq T \tag{212}$$

$$\emptyset \subset F \subseteq A \tag{213}$$

$$\emptyset \quad \subset \quad B \quad \subseteq \quad A \tag{214}$$

$$\emptyset \quad \subset \quad A_f \quad \subseteq \quad A \tag{215}$$

These restrictions are exhaustive. If a restriction is violated, PALS*fit* makes a suitable modification of the range in order to amend the problem and tries to continue. On the other hand, PATFIT refuses to make an analysis in such a case.

7 Simulation of lifetime spectra

In the following we shall very briefly describe the auxiliary programs POSGEN and PALGEN. Both use Monte Carlo simulation to generate spectra from given lifetime parameters. They admit a possibility of checking the PALS*fit* software, since we may subsequently ask the latter to try to retrieve the original parameters. POSGEN takes the formula apparatus in Section 5.1 as granted. Thus this program is primarily meant as an internal test and development tool.

In 1997 Hirade[1] developed a simulation program PALGEN, written in the BASIC language. PALGEN uses Monte Carlo in its simplest version, the so-called *direct simulation*. This is close to the real-world physical setup and admits an independent assessment of the capability of PALS*fit* to recover correct lifetime values. Cheung *et al.* [45], also in 1997, describe a simulation tool which is equivalent to PALGEN; they used their program to study the merits of the *PositronFit* software as an analysis tool. At Risø DTU we have later created a modern FORTRAN-based version of PALGEN. Its functionality is incorporated in PALS*fit*.

The main sampling principle in PALGEN is to split the recorded annihilation time t as follows:

$$t = T_0 + t_{\mathrm{exp}} + t_{\mathrm{gauss}} + t_{\mathrm{shift}} \tag{216}$$

Here T_0 is the given time-zero, t_{exp} the true annihilation time, t_{gauss} the smear component, and t_{shift} a deterministic shift value associated with the latter. We begin with the sampling of t_{exp}. First a random number determines the lifetime component from the given intensities. Then, with the lifetime τ, we have the exponential distribution function

$$F(x) = 1 - e^{-x/\tau} \tag{217}$$

To sample $x = t_{\mathrm{exp}}$ we use the classical Monte Carlo formula,

$$x = F^{-1}(\xi) \tag{218}$$

which is the inverse form of

$$F(x) = \xi \tag{219}$$

with $\xi \in (0, 1)$ being a uniform random number. Then, by replacing ξ with $1 - \xi$ we obtain the sampling formula

$$t_{\mathrm{exp}} = -\tau \ln \xi \tag{220}$$

Next we perform the sampling of t_{gauss}. First we pick one of the gaussian components. Let the standard deviation of this be σ. Then we use the well-known polar method of Box and Muller [46] to generate a normal variate η with variance 1, and obtain

$$t_{\mathrm{gauss}} = \eta\, \sigma \tag{221}$$

Finally the sampling of the background proceeds independently of the remaining spectrum and is accomplished by simple uniform multinomial sampling with bins = channels.

[1] Personal communication, Tetsuya Hirade, Department of Materials Science, Japan Atomic Energy Research Institute.

Apart from statistical fluctuations the two programs POSGEN and PALGEN are found to be equivalent to each other. This provides (yet another) confirmation of the consistence of the PALS*fit* model.

References

[1] P. Kirkegaard, M. Eldrup, O. E. Mogensen, and N. J. Pedersen, "Program system for analysing positron lifetime spectra and angular correlation curves," *Computer Physics Communications*, vol. 23, pp. 307–335, 1981.

[2] P. Kirkegaard, N. J. Pedersen, and M. Eldrup, "PATFIT-88: a data-processing system for positron annihilation spectra on mainframe and personal computers," Tech. Rep. Risø-M-2740, Risø National Laboratory, DK-4000 Roskilde, Denmark, February 1989. Available as pdf-file from `www.palsfit.dk`.

[3] J. V. Olsen, P. Kirkegaard, N. J. Pedersen, and M. Eldrup, "PALSfit: A new program for the evaluation of positron lifetime spectra," *Phys. Stat. Sol. (c)*, vol. 4, pp. 4004–4006, 2007.

[4] M. Eldrup, O. Mogensen, and G. Trumpy, "Positron lifetimes in pure and doped ice and in water," *J. Chem. Phys.*, vol. 57, pp. 495–504, 1972.

[5] M. Eldrup, Y. M. Huang, and B. T. A. McKee, "Estimates of uncertainties in analysis of positron lifetime spectra for metals," *Appl. Phys.*, vol. 15, pp. 65–71, 1978.

[6] W. Puff, "The influence of several parameters on the lifetimes and intensities of positron lifetime spectra of metals," *Appl. Phys.*, vol. 18, pp. 165–168, 1979.

[7] H. Sormann, P. Kindl, and W. Puff, "Investigations on the reliability of a multi-component analysis of positron lifetime spectra, using a new method of producing computer-simulated test spectra," *Nucl. Instr. and Meth.*, vol. 206, pp. 203–209, 1983.

[8] H. Sormann, P. Kindl, and W. Puff, "Reliability tests of the multi-component analysis of positron lifetime spectra," in *Positron Annihilation* (P. C. Jain, R. M. Singru, and K. P. Gopinathan, eds.), pp. 848–850, World Scientific, 1985.

[9] H. Sormann, P. Kindl, and G. Reiter, "Numerical evaluation of disturbed positron lifetime spectra," in *Proc. 8'th Int. Conf. on Positron Annihilation*, (Gent, Belgium), 1988. World Scientific 1989.

[10] M. Eldrup, I. K. MacKenzie, B. T. A. McKee, and D. Segers, "Open discussion on experimental techniques and data analysis (for bulk systems)," in *Proc. 8'th Int. Conf. on Positron Annihilation*, (Gent, Belgium), 1988. World Scientific 1989.

[11] M. Franz, T. Hehenkamp, J.-E. Kluin, and J. D. Gervey, "Computer simulation of positron-lifetime spectroscopy on thermally generated vacancies in copper and comparison with experimental results," *Phys. Rev. B*, vol. 48, pp. 3507–3510, 1993.

[12] M. Eldrup, O. E. Mogensen, and J. H. Evans, "A positron annihilation study of the annealing of electron irradiated molybdenum," *J. Phys. F: Metal Phys.*, vol. 6, pp. 499–521, 1976.

[13] H. E. Hansen, R. Talja, H. Rajainmäki, H. K. Nielsen, B. Nielsen, and R. M. Nieminen, "Positron studies of hydrogen-defect interactions in proton irradiated molybdenum," *Appl. Phys.*, vol. A36, pp. 81–92, 1985.

[14] K. O. Jensen, M. Eldrup, N. J. Pedersen, and J. H. Evans, "Annealing behaviour of copper and nickel containing high concentrations of krypton studied by positron annihilation and other techniques," *J. Phys. F: Metal Phys.*, vol. 18, pp. 1703–1724, 1988.

[15] F. M. Jacobsen, M. Eldrup, and O. E. Mogensen, "The temperature dependence of the two positronium bubble states in liquid SF_6," *Chem. Phys.*, vol. 50, pp. 393–403, 1980.

[16] D. Lightbody, J. N. Sherwood, and M. Eldrup, "Temperature and phase dependence of positron lifetimes in solid cyclohexane," *Chem. Phys.*, vol. 93, pp. 475–484, 1985.

[17] F. M. Jacobsen, O. E. Mogensen, and N. J. Pedersen, "High resolution positron lifetime measurements in non-polar liquids," in *Proc. 8'th Int. Conf. on Positron Annihilation*, (Gent, Belgium), 1988. World Scientific 1989.

[18] P. Kindl, W. Puff, and H. Sormann, "A free four-term analysis of positron lifetime spectra of γ-irradiated teflon," *Phys. stat. sol. (a)*, vol. 58, pp. 489–494, 1980.

[19] M. Eldrup, B. Singh, S. Zinkle, T. Byun, and K. Farrell, "Dose dependence of defect accumulation in neutron irradiated copper and iron," *J. Nucl. Mater.*, vol. 307-311, pp. 912–917, 2002.

[20] B. Somieski, T. E. M. Staab, and R. Krause-Rehberg, "The data treatment influence on the spectra decomposition in positron lifetime spectroscopy. part 1," *Nucl. Instr. and Meth. A*, vol. 381, pp. 128–140, 1996.

[21] T. E. M. Staab, B. Somieski, and R. Krause-Rehberg, "The data treatment influence on the spectra decomposition in positron lifetime spectroscopy. part 2," *Nucl. Instr. and Meth. A*, vol. 381, pp. 141–151, 1996.

[22] G. Dlubek, "Do MELT and CONTIN programs accurately reveal the lifetime distribution in polymers," *Nucl. Instr. and Meth. B*, vol. 142, pp. 191–202, 1998.

[23] W. K. Warburton, "DBLCON: A version of Positronfit with non-gaussian prompt for analysing positron lifetime spectra," *Computer Physics Communications*, vol. 13, pp. 371–379, 1978.

[24] C. J. Virtue, R. J. Douglas, and B. T. A. McKee, "Interactive Positronfit: A new version of a program for analysing positron lifetime spectra," *Computer Physics Communications*, vol. 15, pp. 97–105, 1978.

[25] W. Puff, "PFPOSFIT: A new version of a program for analysing positron lifetime spectra with non-gaussian prompt curve," *Computer Physics Communications*, vol. 30, pp. 359–368, 1983.

[26] W. Lühr-Tanck, H. Bosse, T. Kurschat, M. Ederhof, A. Sager, and T. Hehenkamp, "Positron lifetime and Doppler-broadening measurements in noble metals and their alloys," *Appl. Phys.*, vol. A44, pp. 209–211, 1987.

[27] L. Dorikens-Vanpraet, D. Segers, and M. Dorikens, "The influence of geometry on the resolution of a positron annihilation lifetime spectromenter," *Appl. Phys.*, vol. 23, pp. 149–152, 1980.

[28] S. Dannefaer, "On the effect of backscattering of γ-quanta and statistics in positron-annihilated lifetime measurements," *Appl. Phys.*, vol. A26, pp. 255–259, 1981.

[29] H. Saito, Y. Nagashima, T. Kuvihara, and T. Hyodo, "A new positron lifetime spectrometer using a fast digital oscilloscope and BaF_2 scintillators," *Nucl. Instr. and Meth. in Phys. Res. A*, vol. 487, pp. 612–617, 2002.

[30] J. Nissilä, K. Rytsölä, R. Aavikko, A. Laakso, K. Saarinen, and P. Hautojärvi, "Performance analysis of a digital positron lifetime spectrometer," *Nucl. Instr. and Meth. in Phys. Res. A*, vol. 538, pp. 778–789, 2005.

[31] F. Becvar, J. Cizek, I. Prochazka, and J. Janotova, "The asset of ultra-fast digitizers for positron-lifetime spectroscopy," *Nucl. Instr. and Meth. in Phys. Res. A*, vol. 539, pp. 372–385, 2005.

[32] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, pp. 431–441, 1963.

[33] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quart. Appl. Math.*, vol. 2, pp. 164–168, 1944.

[34] J. J. Moré, "The Levenberg–Marquardt algorithm: Implementation and theory," in *Lecture Notes in Mathematics, 630, Proceedings, Biennial Conference, Dundee 1977* (A. Dold, B. Eckmann, and G. A. Watson, eds.), pp. 105–116, Springer-Verlag, 1978.

[35] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "User guide for MINPACK-1," Tech. Rep. ANL-80-74, Argonne National Laboratory, Applied Mathematics Division, 9700 South Cass Avenue, Argonne, Illinois 60439, 1980.

[36] P. Kirkegaard and M. Eldrup, "POSITRONFIT: A versatile program for analysing positron lifetime spectra," *Computer Physics Communications*, vol. 3, pp. 240–255, 1972.

[37] P. Kirkegaard, "A FORTRAN IV version of the sum-of-exponential least-squares code EXPOSUM," Tech. Rep. Risø-M-1279, Risø National Laboratory, DK-4000 Roskilde, Denmark, September 1970. Available in pdf from `www.palsfit.dk`.

[38] P. Kirkegaard, "Some aspects of the general least-squares problem for data fitting," Tech. Rep. Risø-M-1399, Risø National Laboratory, DK-4000 Roskilde, Denmark, August 1971. Available in pdf from `www.palsfit.dk`.

[39] P. Kirkegaard and M. Eldrup, "The least-squares fitting programme POSITRONFIT: principles and formulas," Tech. Rep. Risø-M-1400, Risø National Laboratory, DK-4000 Roskilde, Denmark, September 1971. Available in pdf from `www.palsfit.dk`.

[40] L. Kaufman, "A variable projection method for solving separable nonlinear least squares problems," *BIT*, vol. 15, pp. 49–57, 1975.

[41] L. Kaufman and V. Pereyra, "A method for separable nonlinear least squares problems with separable nonlinear equality constraints," *SIAM J. Num. An.*, vol. 15, pp. 12–20, 1978.

[42] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPACK, User's guide*. Philadelphia: SIAM, 1979.

[43] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*. Cambridge: Cambridge University Press, 1986.

[44] M. Metcalf and J. Reid, *FORTRAN 90/95 explained*. Oxford: Oxford University Press, 1996.

[45] S. H. Cheung, C. D. Beling, S. H. Y. Fung, and P. K. MacKeown, "Investigation into the use of POSITRONFIT in the recovery of lifetime parameters using Monte-Carlo simulated lifetime spectra," *Materials Science Forum*, vol. 255-257, pp. 738–740, 1997.

[46] G. Box and M. Muller, "A note on the generation of random normal deviates," *Annals Math. Statistics*, vol. 29, pp. 610–611, 1958.