

PALSfit3: A software package for analysing positron lifetime spectra

Kirkegaard, Peter; Olsen, Jens V.; Eldrup, Morten Mostgaard

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Kirkegaard, P., Olsen, J. V., & Eldrup, M. M. (2017). PALSfit3: A software package for analysing positron lifetime spectra. Kgs. Lyngby: Technical University of Denmark (DTU).

DTU Library Technical Information Center of Denmark

General rights

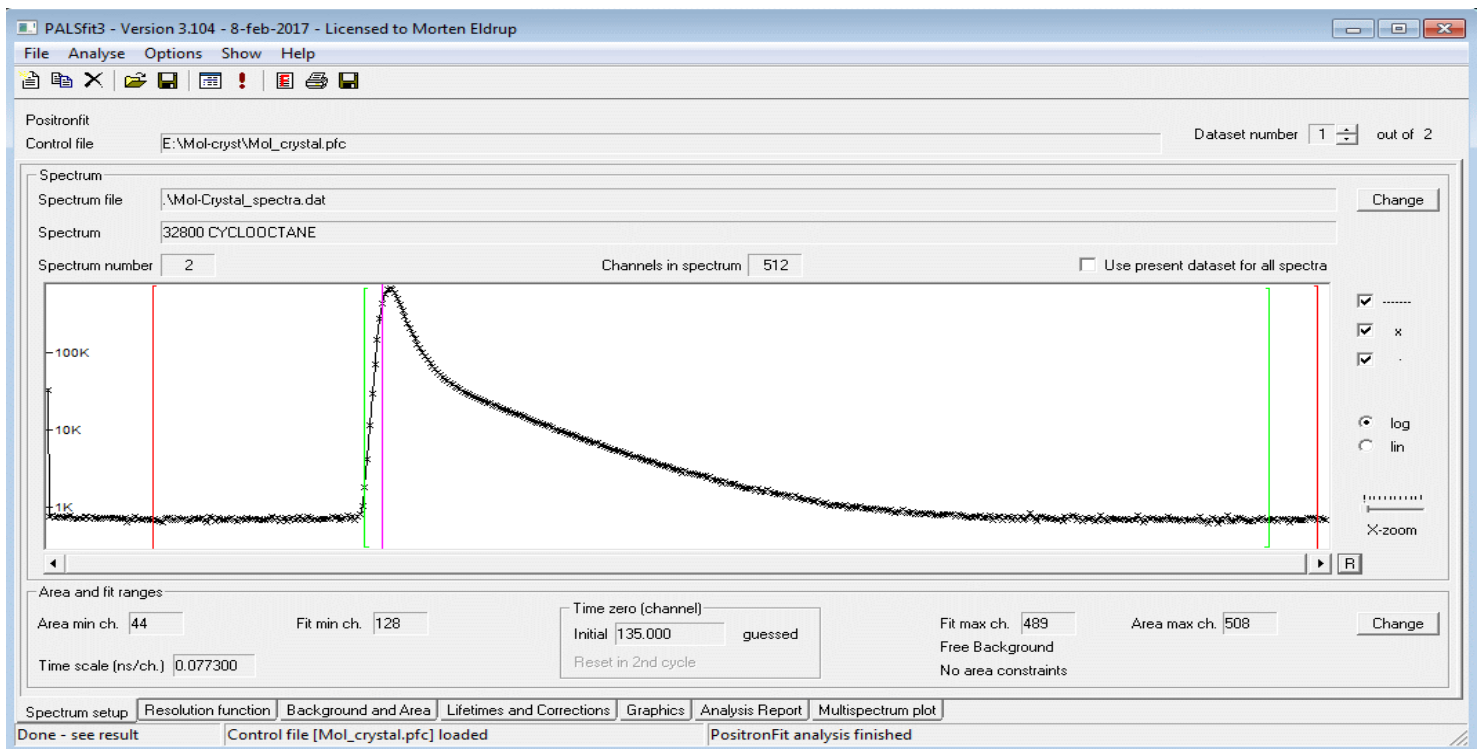
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



A software package for analysing positron lifetime spectra



Peter Kirkegaard, Jens V. Olsen and Morten Eldrup
Technical University of Denmark, Risø Campus

March 2017

Authors: Peter Kirkegaard, Jens. V. Olsen and Morten Eldrup

Title: PALS*fit3*:

A software package for analyzing positron lifetime spectra

Institutes: DTU AIT and DTU Energy

Abstract:

The present report describes a Windows based computer program called PALS*fit3*. The purpose of the program is to carry out analyses of spectra that have been measured by positron annihilation lifetime spectroscopy (PALS). PALS*fit3* is based on the well tested PATFIT and PALS*fit* programs, which have been used extensively by the positron annihilation community. The present document describes the mathematical foundation of the PALS*fit3* model as well as a number of features of the program. The cornerstones of PALS*fit3* are two least-squares fitting modules: POSITRONFIT and RESOLUTIONFIT. In both modules a model function will be fitted to a measured lifetime spectrum. This model function consists of a function representing the physics of the positron decay which is convoluted with the experimental time resolution function, plus a constant background. The 'physics function' consists of a sum of decaying exponentials each of which may be broadened by convolution with a log-normal lifetime distribution. 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. In the POSITRONFIT module, the fitting parameters to be extracted from a measured spectrum are for each lifetime component its mean lifetime and its broadening as well as its intensity. A correction for positrons annihilating outside the sample can be made as part of the analysis. In the RESOLUTIONFIT module, parameters determining the shape of the time resolution function can be fitted. The extracted resolution function may then be used in POSITRONFIT.

Graphics displays are provided to ease the selection of some of the input parameters and to display results of spectrum analysis. The results are also available in a text window.

PALS*fit3* is verified on Windows-XP and Windows 7, 8 and 10.

The PALS*fit3* software can be acquired from the Technical University of Denmark (<http://PALSfit.dk>)

DTU AIT
and

DTU Energy

March 2017

Pages: 60

References: 65

ISBN: 978-87-92986-61-0

Technical University of Denmark
Department of
Energy Conversion and Storage
Risø Campus
Frederiksborgvej 399
DK-Roskilde
Denmark

www.energy.dtu.dk

Contents

1	Introduction and overview	<i>4</i>
1.1	Acknowledgements	<i>4</i>
1.2	PALS <i>fit</i> 3	<i>4</i>
2	About PALS<i>fit</i>	<i>5</i>
2.1	General fitting criterion	<i>5</i>
2.2	The POSITRONFIT model	<i>6</i>
2.3	The log-normal extension	<i>8</i>
2.4	Source correction	<i>9</i>
2.5	The RESOLUTIONFIT model	<i>9</i>
3	Input and output	<i>10</i>
3.1	PALS <i>fit</i> input	<i>10</i>
3.2	POSITRONFIT control file	<i>11</i>
3.3	RESOLUTIONFIT control file	<i>14</i>
3.4	PALS <i>fit</i> output	<i>15</i>
3.5	POSITRONFIT main output	<i>16</i>
3.6	RESOLUTIONFIT main output	<i>18</i>
3.7	Channel ranges	<i>19</i>
4	Experience with PALS<i>fit</i>	<i>20</i>
4.1	POSITRONFIT experience	<i>20</i>
4.2	RESOLUTIONFIT experience	<i>21</i>
5	Appendix A: Fit and statistics	<i>24</i>
5.1	Unconstrained NLS data fitting	<i>24</i>
5.2	Constraints	<i>26</i>
5.3	Statistical analysis	<i>26</i>
5.4	Marquardt minimization	<i>32</i>
5.5	Separable least-squares technique	<i>34</i>
5.6	Various mathematics, statistics, and numerics	<i>35</i>
6	Appendix B: Model details	<i>41</i>
6.1	POSITRONFIT	<i>41</i>
6.2	RESOLUTIONFIT	<i>44</i>
7	Appendix C: Log-normal details	<i>46</i>
7.1	Log-normal POSITRONFIT model formulas	<i>46</i>
7.2	Fixing one of the log-normal parameters τ_m or σ	<i>47</i>
7.3	Jacobian matrix for output presentation	<i>48</i>
7.4	Numerical evaluation of log-normal integrals	<i>49</i>
7.5	Discarding log-normal lifetime components	<i>50</i>
8	Appendix D: Quality check	<i>50</i>
8.1	Simulation of lifetime spectra	<i>50</i>
8.2	Verifying POSITRONFIT by statistical analysis	<i>51</i>
8.3	Comparison of PALS <i>fit</i> 3 with LT10	<i>54</i>
	References	<i>56</i>

1 Introduction and overview

An important aspect of doing experiments by positron annihilation lifetime spectroscopy (PALS) is to be able to reliably analyse the measured spectra in order to extract physically meaningful parameters. A number of computer programs have been developed over the last many years by various authors for this purpose. Most of these programs have used various methods of least squares fitting of a model function to the experimental data [1–15], while others carry out a direct deconvolution of the measured spectra using different criteria for obtaining the optimal solution [16–23]. At our laboratory we have concentrated on developing programs for least squares fitting of positron lifetime spectra.

1.1 Acknowledgements

The development of *PALSfit* programs, including the present report, has been supported by the Technical University of Denmark (AIT Department and Department of Energy Conversion and Storage). Also the input and inspiring questions from colleagues world-wide have been much appreciated. Thanks are due to Dr. Tetsuya Hirade for allowing us to apply his PALGEN program for further development of spectrum simulation tools, and to Dr. Anne Margrethe Larsen who resolved several issues about the L^AT_EX typesetting system. Niels Jørgen Pedersen (deceased) was involved in the early phases of this project.

1.2 *PALSfit3*

PALSfit Version 3, or *PALSfit3* for short, is our most recent software of this kind. It is based on the well tested PATFIT and *PALSfit* (Version 1 and 2) software [8, 12, 24], which have been used extensively by the positron annihilation community. Now *PALSfit3* allows each lifetime component to be fitted not only with a simple decaying exponential, but also with a broadened decaying exponential function. The reason for introducing broadening of lifetimes arises from the fact that if all lifetime components are assumed to be simple decaying exponentials, it is very difficult, if not impossible, to reliably separate groups of components whose lifetime values are close. This is due to the fact that when fitting a sum of simple decaying exponentials, a very strong correlation exists between such lifetimes. Hence, instead of assuming several individual lifetime components in the analysis, we allow in *PALSfit3* that many components with close lifetimes may be joined into just one component which in principle is a decaying exponential, but with a lifetime that can have a distribution of values (this distribution has been assumed to be a so-called log-normal distribution). In addition, a number of new graphics displays are provided to ease the selection of some input parameters and to display results of spectrum analyses.

The two cornerstones in *PALSfit* 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 *PALSfit* 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. This model function consists of a function representing the physics of the positron decay which is convoluted with the experimental time resolution function, plus a constant background. The ‘physics function’ consists of a sum of decaying exponentials each of which may be broadened

by convolution with a log-normal distribution (only in POSITRONFIT). 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.

A correction for the contribution to a measured lifetime spectrum from positrons annihilating outside the sample can be made during the POSITRONFIT analysis.

The front page of the present report shows an example of a POSITRONFIT Spectrum setup window. Various additional tabs at the bottom can be selected to enter or change input data (Resolution function, Background and Area and Lifetimes and Corrections) or display results of analyses (Graphics, Text output and Multispectrum plot).

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 function may then be used in POSITRONFIT to analyse more complicated spectra. PALS*fit3* can easily feed the resolution function determined by RESOLUTIONFIT into POSITRONFIT. In the latter program the shape of the resolution function is fixed.

In the following, Chapter 2 presents a brief overview of the PALS*fit3* model. In Chapter 3 follows a detailed description of the PALS*fit3* input and output, while Chapter 4 conveys some experiences we and others have gained with PALS*fit3* and its predecessors.

Appendices A–C contain the mathematical and statistical details which constitute the foundation for the programs. Appendix D deals with quality checks of the programs, carried out by statistical analyses of the results from fitting a long series of simulated spectra as well as by comparing results obtained by both PALS*fit3* and LT programs (Giebel and Kansy [15]).

PALS*fit3* is available from the website www.palsfit.dk.

A contemporary edition of the PATFIT package, roughly equivalent to PALS*fit3* without its GUI, is available too. It contains command-driven versions of POSITRONFIT and RESOLUTIONFIT and might be useful for batch processing under Windows or in a Linux environment where it is also available.

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, \dots, b_k that minimizes

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

where n is the number of data values, $f_i(b_1, \dots, b_k)$ 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”,

$$w_i = \frac{1}{s_i^2} \quad (2)$$

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 Appendix A. 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 The POSITRONFIT model

Let us first consider the “simple” POSITRONFIT model, i.e. without any broadening of components. In this case the model function is a sum of decaying exponentials convoluted with the resolution function of the lifetime spectrometer, plus a constant background. Each exponential corresponds to a single lifetime component. 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 Appendix B, Section 6.1; here we state the model in an annotated form using the symbol $*$ for convolution:

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

where

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

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

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

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 weighted sum of k_g Gaussians which may be displaced with respect to each other:

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

where

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

and

$$\sum_{p=1}^{k_g} \omega_p = 1 \quad (8)$$

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

$$\text{FWHM}_p = 2\sqrt{2 \ln 2} s_p \approx 2.3548 s_p \quad (9)$$

We also see that

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

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 often measured in ns. In this *physical time* representation all the other temporal model parameters in (3–10), i.e. t , τ , T_0 , Δ_p , and s_p , would be in ns too, and it is natural to set $T_0 = 0$.

However, for some of the quantities it is more convenient to use a time scale directly related to the spectrum recording system. Let the spectrum be recorded in n_{ch} channels, numbered $i_{\text{ch}} = 1, 2, \dots, n_{\text{ch}}$. Each channel represents a time slot whose common width will be used as the time unit. We let channel No. i_{ch} begin at $t = i_{\text{ch}} - 1$ and end at $t = i_{\text{ch}}$. 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 $0, 1, 2, \dots$. 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 scales for physical time and channel time are assumed to be connected by a *fixed* parameter C :

$$1 \text{ channel} = C \text{ ns} \quad (11)$$

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

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

$$f_i = \int_{i-1}^i f(t) dt, \quad i = 1, \dots, n \quad (12)$$

with $f(t)$ given by (3), so that (12) 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 $i_{\text{ch}}^{\text{min}}$ and ends in $i_{\text{ch}}^{\text{max}}$ (inclusive), where

$$1 \leq i_{\text{ch}}^{\text{min}} \leq i_{\text{ch}}^{\text{max}} \leq n_{\text{ch}} \quad (13)$$

we should generalize (12) to

$$f_i = \int_{i_{\text{ch}}^{\text{min}}+i-2}^{i_{\text{ch}}^{\text{min}}+i-1} f(t) dt, \quad i = 1, \dots, n \quad (14)$$

where now

$$n = i_{\text{ch}}^{\text{max}} - i_{\text{ch}}^{\text{min}} + 1 \quad (15)$$

In any case, we obtain as the result a model for the least-squares analysis of the form

$$f_i = \sum_{j=1}^{k_0} F_{ij} + B \quad (16)$$

where F_{ij} is the contribution from lifetime component j in spectrum channel $i_{\text{ch}}^{\text{min}} + i - 1$. (We relegate the full write-up of F_{ij} to Appendix B, Section 6.1.) We recall that f_i in (12), (14), and (16) corresponds to $f_i(b_1, \dots, b_k)$ in Section 2.1, formula (1).

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

$$I_j = \frac{A_j \tau_j}{\sum_{k=1}^{k_0} A_k \tau_k} \quad (17)$$

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.

$$\sum_{j=1}^{k_0} h_{lj} I_j = 0 \quad (18)$$

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,

$$\sum_{j=1}^{k_0} A_j \tau_j + \text{background area} = \text{constant} \quad (19)$$

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

The necessary mathematical processing of the POSITRONFIT model for the least-squares analysis is outlined in Appendix B, Section 6.1.

2.3 The log-normal extension

Until now we have assumed that each lifetime component consists of a decaying exponential function (convoluted with a resolution function) with a single decay rate (equal to the inverse of the lifetime τ_*). We may say that its probability density function (pdf) is a delta function,

$$f(\tau) = \delta(\tau - \tau_*) \quad (20)$$

Sometimes it is more realistic to assume that a lifetime may have some continuous distribution. We shall here consider the case where one (or more) of the lifetimes obeys the *log-normal* distribution which is implemented in *PALSfit3*.

Let us recapitulate the general properties of the log-normal distribution. We say that the stochastic variable τ has a log-normal distribution if $\ln \tau$ has a normal distribution,

$$\ln \tau \in N(\ln \tau_*, \sigma_*^2) \quad (21)$$

for some positive τ_* and σ_* . Thus the mean and variance of $\ln \tau$ are $\ln \tau_*$ and σ_*^2 , respectively. To find F , the cumulative distribution function (CDF) of τ , we note that

$$F(x) = P\{\tau < x\} = P\left\{\frac{\ln \tau - \ln \tau_*}{\sigma_*} < \frac{\ln x - \ln \tau_*}{\sigma_*}\right\} \quad (22)$$

In terms of the CDF for $N(0,1)$,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}t^2} dt = \frac{1}{2} \left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \quad (23)$$

eq. (22) can be written

$$F(x) = \Phi\left(\frac{\ln x - \ln \tau_*}{\sigma_*}\right) \quad (24)$$

Taking the derivative, this gives the pdf of τ :

$$f(\tau) = \frac{1}{\tau \sigma_* \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_*^2}(\ln \tau - \ln \tau_*)^2\right) \quad (25)$$

with support $(0, \infty)$. In the limit $\sigma_* \rightarrow 0$ (25) tends to (20). We see from (24) that τ_* equals the median value for the log-normal distribution; σ_* is dimensionless and is sometimes called the scale or the shape parameter. The mean and variance are

$$E[\tau] = \tau_m = \tau_* \exp\left(\frac{1}{2}\sigma_*^2\right) \quad (26)$$

$$\operatorname{Var}[\tau] = \sigma^2 = \tau_*^2 \exp(\sigma_*^2)(\exp(\sigma_*^2) - 1) \quad (27)$$

Hence the standard deviation is

$$\sigma = \tau_* \exp\left(\frac{1}{2}\sigma_*^2\right) \sqrt{\exp(\sigma_*^2) - 1} \quad (28)$$

and the coefficient of variation is

$$\frac{\sqrt{\operatorname{Var}[\tau]}}{E[\tau]} = \frac{\sigma}{\tau_m} = \sqrt{\exp(\sigma_*^2) - 1} \quad (29)$$

We see that when the parameter σ_* is small, it is approximately equal to the coefficient of variation. By solving the equation

$$\frac{d}{d\tau} \ln(f(\tau)) = 0 \quad (30)$$

for τ , where $f(\tau)$ is given by (25), we find that the most probable τ -value (mode) for the log-normal distribution is

$$\tau_{\max} = \tau_* \exp(-\sigma_*^2) \quad (31)$$

The log-normal distribution is completely specified by either of the parameter sets (τ_*, σ_*) or (τ_m, σ) . The relation between the two sets was given in (26–27); the inverse transformation reads

$$\tau_* = \frac{\tau_m^2}{\sqrt{\sigma^2 + \tau_m^2}} \quad (32)$$

$$\sigma_*^2 = \ln \left(1 + \frac{\sigma^2}{\tau_m^2} \right) \quad (33)$$

The Full Width at Half Maximum is given by

$$\text{FWHM} = 2\tau_* \exp(-\sigma_*^2) \sinh(\sqrt{2 \ln 2} \sigma_*) \quad (34)$$

or, equivalently

$$\text{FWHM} = 2\tau_m^4 (\sigma^2 + \tau_m^2)^{-\frac{3}{2}} \sinh \left[\sqrt{2 \ln 2 \cdot \ln(1 + \sigma^2/\tau_m^2)} \right] \quad (35)$$

In POSITRONFIT we use the representation (τ_*, σ_*) for internal calculations, while (τ_m, σ) is used for input and output. More details related to POSITRONFIT and the log-normal distribution are given in Appendix C.

2.4 Source correction

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 spectrum data are fitted in a first iteration cycle. Then, the spectrum for the source correction is subtracted from the raw spectrum. The corrected spectrum is then fitted 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 (16) (with $B = 0$) as follows:

$$f_i^s = \sum_{j=1}^{k_s} F_{ij}^s \quad (36)$$

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

$$\sum_{j=1}^{k_s} A_j^s \tau_j^s = \alpha \sum_{j=1}^{k_0} A_j \tau_j \quad (37)$$

Log-normal broadening of the source-correction lifetime components is accepted.

2.5 The RESOLUTIONFIT model

The RESOLUTIONFIT model function is basically the same as for POSITRONFIT, Eqs. (3–16). A few additional formulas relevant to RESOLUTIONFIT are given in Appendix B, Section 6.2. The purpose of RESOLUTIONFIT is to extract the shape of the resolution function. The widths and shifts (Eqs. (9) and (7)) of the Gaussians in the resolution function are therefore included as

fitting parameters. In order not to have too many fitting 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. Moreover, log-normal broadening of lifetime components is not available. 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 shifts 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. At least one shift must be fixed.

3 Input and output

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. For one analysis of the spectrum, these data shall be organised in a block structured *dataset* which is saved in a so-called *control file*. In order to carry out several analyses of the same spectrum or of different spectra, several datasets may be stacked in the same control file.

The most direct way to generate datasets and control files is to run PALS*fit* and edit the input data by using the PALS*fit* menus. Nevertheless there might be situations where an inspection or an external editing of the content of a control file is required. For example, as mentioned in Chapter 1, it may be useful in certain situations (batch processing) 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. Note that PATFIT and PALS*fit* are input-compatible.

In any case, the knowledge of the structure of the control files may give the user a good overview of the capabilities of PALS*fit3*. Therefore, in the following we shall describe the contents of the control files for POSITRONFIT and RESOLUTIONFIT in some detail.

Each dataset in a control file is partitioned into a number of data blocks, corresponding roughly to the menus in PALS*fit*. Each block is initiated by a block header. For example, the first block header reads

```
POSITRONFIT DATA BLOCK 1: OUTPUT OPTIONS
```

in the case of POSITRONFIT, and similarly for RESOLUTIONFIT.

3.1 PALS*fit* input

As mentioned above, PALS*fit3* can interactively generate and/or edit the control file for either POSITRONFIT or RESOLUTIONFIT. Previously generated control files can be used as default input values. A number of checks on the consistency of the generated control data are built into PALS*fit3*. PALS*fit3* is largely self-explanatory regarding input editing.

3.2 POSITRONFIT control file

A sample PALS*fit3* control file for POSITRONFIT with a single dataset is shown below:

```
POSITRONFIT DATA BLOCK 1: OUTPUT OPTIONS
0000
POSITRONFIT DATA BLOCK 2: SPECTRUM
  2049
(10i7)
.\Polymer_Spectrum.DAT
37200 PVAC, T=414K
0
POSITRONFIT DATA BLOCK 3: CHANNEL RANGES. TIME SCALE. TIME-ZERO.
  5
  2000
  275
  2000
0.026800
G
  285.000
POSITRONFIT DATA BLOCK 4: RESOLUTION FUNCTION
  3
  0.29840  0.25460  0.23960
  12.000  13.000  75.000
 -0.10380  0.08020  0.00000
POSITRONFIT DATA BLOCK 5: LIFETIMES AND INTENSITY CONSTRAINTS
  3
GGG
  0.2000  0.5000  3.2000
FFF
  0.0000  0.0000  0.0000
  0
POSITRONFIT DATA BLOCK 6: BACKGROUND CONSTRAINTS
  1
  1400
  2000
POSITRONFIT DATA BLOCK 7: AREA CONSTRAINTS
  0
POSITRONFIT DATA BLOCK 8: SOURCE CORRECTION
  2
  0.3803  2.0000
  0.0000  0.0000
  86.9972 13.0028
  9.1957
  1
  3
GGG
  0.1500  0.4000  3.2000
FGF
  0.0000  0.2000  0.0000
  -1
 -3.0000  0.0000  1.0000
```

Block 1 contains output options:

Apart from the block header there is only one record. It contains normally 4 integer keys¹ in its first 4 positions. Each key is either 0 or 1. The value 1 causes some output action to be taken, whereas 0 omits this action. The actions of the 4 keys are:

1. Write input echo to result file
2. Write each iteration output to result file

¹In fact, the record may contain an additional key which is not an output option but an indicator of the so-called log-normal fineness, see Appendix C, Section 7.4.

3. Write residual plot to result file
4. Write correlation matrix to result file

Regardless of these keys, POSITRONFIT always produces the *Main Output*, cf. Section 3.4.

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 [25].

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 the integer INSPEC taking a value of either 0 or 1. 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 an identification label for 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 spectrum file with the file name entered above. The program opens 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 are read from the subsequent records in exactly the same way as in the case INSPEC = 1.

Block 3 contains information related to the measuring system:

The first 2 records (after the block header) contain 2 channel numbers ICHA1 and ICHA2. These numbers are lower and upper bounds for the definition of a total area range.

The next 2 records contain also 2 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 C measured in ns, cf. (11) in Section 2.2.

The last 2 records in the block deal with T_0 (time=0 channel number, may be fractional). 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 .

Rules for proper channel specifications are given in Section 3.7.

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 3 records contains k_g numbers. In the first record we have the full widths at half maxima of the Gaussians (in ns), FWHM_j , $j = 1, \dots, k_g$, in the second their relative intensities (in percent) ω_j , $j = 1, \dots, k_g$, and in the third their peak displacements (in ns) Δt_j , $j = 1, \dots, k_g$.

Block 5 contains data for lifetime components and intensity constraints:

The first record (after the block header) contains the number k_0 of lifetime components assumed in the model.

Each of the next 4 records contain k_0 data. In the first we have the constraint flags (G = guessed, F = fixed) for the lifetimes (mean values in case of log-normal broadening). The second record contains the initial values (guessed or fixed) for each of the k_0 lifetimes. In the standard case of no log-normal broadening of the lifetime components, the 3rd record should contain the flags F (=fixed), while the 4th should contain k_0 zeroes. In the case of log-normal broadening (= standard deviation), the 3rd record should contain the flags F or G, while the 4th should contain the initial values (guessed or fixed) for each of the k_0 broadenings.

The next record after this contains an integer m denoting the number and type of intensity constraints. $|m|$ is equal to the number of constraints; m itself may be positive, negative, or zero. If $m = 0$ there is no further input data in the 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 , $l = 1, \dots, m$ and I_{j_l} , $l = 1, \dots, m$; 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} , $j = 1, \dots, k_0$ to the intensities for one of the linear combinations, cf. equation (18) in Section 2.2.

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 (to be fitted); 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 2 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. $k_s = 0$ means no source correction, in which case the present block contains no more data.

If $k_s > 0$, the next 3 records contain the lifetimes τ_j^s , the lifetime broadenings σ_j^s and the relative intensities (in percent) I_j^s , $j = 1, \dots, k_s$ 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 (37) in Section 2.4.

Then follows a record with an integer ISEC. When ISEC = 0 the new iteration cycle after the source correction starts from parameter 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 dataset is completed. However, as previously mentioned, PALS*fit3* accepts multiple datasets in the same POSITRONFIT control file.

3.3 RESOLUTIONFIT control file

A sample PALS*fit3* control file for RESOLUTIONFIT with a single dataset is shown below:

```

RESOLUTIONFIT DATA BLOCK 1: OUTPUT OPTIONS
0000
RESOLUTIONFIT DATA BLOCK 2: SPECTRUM
    1023
(10i7)
.\Cu_Spectrum.DAT
51108 Cu-ann
0
RESOLUTIONFIT DATA BLOCK 3: CHANNEL RANGES. TIME SCALE. TIME-ZERO.
    2
    1023
    175
    1000
    0.015800
    194.000
RESOLUTIONFIT DATA BLOCK 4: RESOLUTION FUNCTION
    3
GGG
    0.25000  0.20000  0.35000
    70.000  20.000  10.000
FGG
    0.00000  -0.04000  -0.02500
RESOLUTIONFIT DATA BLOCK 5: LIFETIMES AND INTENSITY CONSTRAINTS
    3
FFG
    0.1100  0.1800  0.4000
    0
RESOLUTIONFIT DATA BLOCK 6: BACKGROUND CONSTRAINTS
    0

```

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:

It is identical to the corresponding block in the POSITRONFIT control file, except for the status of T_0 . In RESOLUTIONFIT the last record always contains a guessed value of T_0 , so there is no preceding G or F flag.

Rules for proper channel specifications are given in Section 3.7.

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), $\text{FWHM}_j^{\text{ini}}$, $j = 1, \dots, k_g$. The next record contains the k_g Gaussian component intensities in percent, ω_j , $j = 1, \dots, k_g$. 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} , $j = 1, \dots, k_g$.

Block 5 contains data for the lifetime components in the lifetime spectrum as well as constraints on their relative intensities:

It is similar to the corresponding block in the POSITRONFIT control file, but without the two records about the log-normal input.

Block 6 contains data related to the background:

It is identical to the corresponding block in the POSITRONFIT control file.

This completes the RESOLUTIONFIT dataset. Multiple datasets can be handled in the same way as for POSITRONFIT.

3.4 PALSfit output

After a successful POSITRONFIT or RESOLUTIONFIT analysis PALSfit presents the results of the analysis in several output files as well as in graphical displays. The most important of the files is the Analysis report (result file) the content of which is displayed automatically after the analysis comes to an end. It can also be viewed by choosing the “Analysis Report” tab. It 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 (Appendix A, Section 5.3). 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 dataset.
- 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 Appendix A, Section 5.3.

As indicated above, the outputs b)–d) are optional, while the Main Output is always produced. In addition to the Main output, two ‘graphics files’ (*.pfg and *.pft from POSITRONFIT, *.rfg and *.rft from RESOLUTIONFIT) are produced (and may optionally be saved). They contain data necessary for the generation of plots of measured and fitted spectra. These plots will be displayed by choosing the tab “Graphics” or the tab “Multispectrum plot”.

3.5 POSITRONFIT main output

In the following we give an example of the Main Output part of a POSITRONFIT analysis report produced by PALSfit3, with a brief explanation of its contents (for details about the input possibilities consult Section 3.2):

```

PALSfit - Version 3.104 8-feb-2017 - Licensed to Morten Eldrup
Input file: C:\PALSfit3\PALSfit3_test-data2017\PVAc.pfc
P O S I T R O N F I T . Version 3.104      Job time 12:00:49.06 13-MAR-17
*****
37200 PVAc, T=414K
in file: .\Polymer_Spectrum.DAT
*****
Dataset 1                               LT LN LX SX IX BG TZ AR GA
                                           3 0 0 0 0 1 0 0 3

Time scale  ns/channel      :    0.026800
Area range  starts in ch    5 and ends in ch    2000
Fit range   starts in ch   275 and ends in ch    2000

Resolution  FWHM (ns)      :    0.2984    0.2546    0.2396
Function    Intensities (%) :    12.0000   13.0000   75.0000
           Shifts (ns)     :    -0.1038    0.0802    0.0000

----- I n i t i a l   P a r a m e t e r s -----
           Time-zero (ch.no):    285.0000G
           Lifetimes (ns)  :    0.2000G    0.5000G    3.2000G

Background  fixed to mean from ch 1400 to ch 2000 =    551.1963

----- 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 5 iterations
Chi-square =    1768.66 with 1719 degrees of freedom
           Lifetimes (ns)  :    0.2191    0.4635    3.2092
           Intensities (%) :    22.4624   50.7573   26.7803
Time-zero   Channel time   :    285.0458
Total area  from fit       :    3.93911E+06      from table : 3.94045E+06

----- S o u r c e   C o r r e c t i o n -----
           Lifetimes (ns)  :    0.3803    2.0000
           Intensities (%) :    86.9972   13.0028
           Total (%)      :    9.1957

----- I n i t i a l   2 n d   c y c l e   P a r a m e t e r s -----
           Lifetimes (ns)  :    0.1500G    0.4000G    3.2000G
           Sigma (ns)     :    0.2000G
           Lin.comb.coeff. :    -3.0000    0.0000    1.0000

```

```

##### F i n a l   R e s u l t s #####
Dataset 1                               LT LN LX SX IX BG TZ AR GA
                                      3  1  0  0 -1  1  0  0  3

Convergence obtained after 8 additional iterations
Chi-square =      1776.83 with 1719 degrees of freedom
Reduced chi-square = chi-square/dof =      1.034 with std deviation 0.034
Significance of imperfect model = 83.81 %

      Lifetimes (ns)   :      0.1667    0.4265    3.2757
      Std deviations   :      0.0055    0.0015    0.0142

      Sigma (ns)      :                   0.1102
      Std deviations   :                   0.0063

      Intensities(%) LC:    9.3713   62.5150   28.1138
      Std deviations   :    0.0420    0.1679    0.1259

Background  counts/channel : 551.1963
            Std deviations  :      mean

Time-zero   channel time   : 285.0628
            Std deviations  :    0.0149

Total area  from fit       : 3.67870E+06    from table : 3.67939E+06

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

```

This output was obtained by running *PALSfit3* with the dataset in Section 3.2. It does not represent a typical analysis of a spectrum, but rather illustrates a number of program features.

After a heading which contains the spectrum headline the key numbers are displayed in the upper right hand corner. LT indicates the number of lifetime components (k_0), LN of these are log-normally distributed, LX is the number of fixed lifetimes, SX the number of fixed lifetime broadenings, IX 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 3.2, Block 5), BG the type of background constraint (KB, Section 3.2, Block 6), TZ whether time-zero is free or fixed (0 = free, 1 = fixed), AR the type of area constraint (KAR, Section 3.2, Block 7), and GA 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, such as 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 for the first iteration cycle.

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 (Appendix A, Section 5.3).

The next part (“Source correction” and “Initial 2nd Cycle Parameters”) shows the parameters of the chosen source correction, which accounts for those positrons that annihilate outside the sample. It is followed by 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 (Appendix A, Section 5.3). 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 IX 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”.

3.6 RESOLUTIONFIT main output

Below you find an example of the Main Output from a RESOLUTIONFIT analysis by PALSfit, with a brief explanation of its contents (for more details about the input possibilities consult Section 3.3):

```
PALSfit - Version 3.104 8-feb-2017 - Licensed to Morten Eldrup
Input file: C:\PALSfit3\PALSfit3_test-data2017\Cu-ann.rfc
Dataset 1

R E S O L U T I O N F I T Version 3.104   Job time 11:48:35.89 13-MAR-17
*****
51108 Cu-ann
in file: .\Cu_Spectrum.DAT
*****
Dataset 1

Time scale   ns/channel       :   0.015800
Area range   starts in ch    2 and ends in ch 1023
Fit range    starts in ch   175 and ends in ch 1000

Initial      FWHM (ns)       :   0.2500G   0.2000G   0.3500G
Resolution   Intensities (%) :   70.0000  20.0000  10.0000
Function     Shifts (ns)    :   0.0000F  -0.0400G  -0.0250G

Other init.  Time-zero(ctime): 194.0000
Parameters   Lifetimes (ns)  :   0.1100F   0.1800F   0.4000G

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

Convergence obtained after 6 iterations
Chi-square = 842.59 with 815 degrees of freedom
Reduced chi-square = Chi-square/dof = 1.034 with std deviation 0.050
Significance of imperfect model = 75.56 %

-----
Resolution function:
                                                GA  WX  SX
                                                3   0  1
      FWHM (ns)       :   0.2504   0.1921   0.2969
      Std deviations  :   0.0026   0.0048   0.0385

      Intensities (%) :   70.0000  20.0000  10.0000

      Shifts (ns)    :   0.0000  -0.0438  -0.0607
      Std deviations  :   fixed    0.0093   0.0412

-----
Lifetime components:
                                                LT  LX  IX
                                                3   2  0
      Lifetimes (ns)  :   0.1100   0.1800   0.4594
      Std deviations  :   fixed    fixed    0.0160

      Intensities (%) :   82.2079  14.8386   2.9535
      Std deviations  :   0.7790   0.9797   0.2206

-----
Background:
                                                B
                                                0
      Counts/channel : 150.6070
      Std deviation   :   0.4937

-----
Time-zero   Channel time : 193.5894
            Std deviations :   0.3573

Total area  From fit       : 2.81759E+06   From table : 2.82172E+06

Shape parameters for resolution curve (ns):
      N      2      5      10     30     100     300     1000
FW at  1/N   0.2435  0.3756  0.4532  0.5584  0.6598  0.7444  0.8301
MIDP at 1/N   0.0031  0.0057  0.0063  0.0054  0.0022 -0.0021 -0.0073

Peak position of resolution curve: Channel # 192.3999
##### R e s o l u t i o n f i t #####
```

This output was obtained by running `PALSfit3` with the dataset listed in Section 3.3.

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 shifts except one, may be fitting parameters. If the background is fixed to a mean value between certain channel limits, these limits as well as the resulting background value are displayed.

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, similar to the main output from `POSITRONFIT` (for definition of the terms see Appendix A, Section 5.3).

Next follows the estimated values of the fitted (and fixed) parameters and their standard deviations (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 `GA` indicates the number of Gaussians (k_g), `WX` the number of fixed widths, and `SX` the number of fixed shifts. For the lifetime components the `LT` indicates the number of these (k_0), `LX` the number of fixed lifetimes, and `IX` the number and type of intensity constraints. As in `POSITRONFIT`, a positive value of `IX` means fixed intensities, while a negative value indicates constraints on linear combinations of intensities, the absolute value giving the number of constraints. Finally the background output follows, where `B` indicates the type of background constraint (`KB`, Section 3.3, 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`.

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.

3.7 Channel ranges

As we saw in the description of the control files for `POSITRONFIT` (`P`) and `RESOLUTIONFIT` (`R`) in Sections 3.2 and 3.3, 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 acronyms for the channel limits as before, we can collect all the channel ranges in the following table:

Name	Definition	Program	Symbol	Lines in <code>PALSfit3</code> graphics
Total range	[1,NCH]	P R	T	None
Area range	[ICHA1,ICHA2]	P R	A	Red
Fit range	[ICHMIN,ICHMAX]	P R	F	Green
Background range	[ICHBG1,ICHBG2]	P R	B	Blue
Fixed-area range	[ICHBEG,ICHEND]	P	A_f	Brown

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:

$$\emptyset \subset A \subseteq T \tag{38}$$

$$\emptyset \subset F \subseteq A \tag{39}$$

$$\emptyset \subset B \subseteq A \tag{40}$$

$$\emptyset \subset A_f \subseteq A \tag{41}$$

These restrictions are exhaustive. If a restriction is violated, *PALSfit3* 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.)

4 Experience with *PALSfit*

In this section we shall give a short account of some of the experiences we (and others) have had with *PALSfit3* and its predecessor versions, in particular 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 [8, 24], from which we shall quote frequently in the remaining part of the present Chapter.

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. [26–34]. 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.

4.1 *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 [35–37] while in some insulators where positronium is formed, up to four components (unconstrained analysis) may be extracted e.g. [38–41]. (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. [26, 33]. Various other aspects of the analysis of positron lifetime spectra are discussed in for example [42–45]). 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 of 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.

The important, particular feature of *PALSfit3* is that each lifetime component may be broadened by a log-normal distribution, whose width is characterized by its standard deviation σ . These widths may be fixed or act as fitting parameters. During the fitting procedure it may happen that a σ becomes increasingly smaller (i.e. approaching zero) thus indicating that the lifetime component is essentially a simple exponential decay. In such cases *POSITRONFIT* neglects the broadening and continues the fitting procedure with a simple decaying exponential instead (Appendix C, Section 7.5). If such an event has taken place during the fitting, it will be indicated on the top of the Main output.

4.2 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 that of others [6, 7, 9, 33, 41] suggest that this separation is possible, although in general great care is necessary to obtain well-defined results [8, 33]. 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 (Sections 3.3 and 3.6), 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. [46]). 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 80–90 % intensity or more.

On the other hand, one cannot be sure that the lifetimes can always be separated easily from the resolution function. 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 come out with 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 [6, 9], 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 [9, 33], and the better the resolution is, the less does a small difference influence the extracted lifetime parameters [33]. The sum-of-Gaussians used in *PALSfit* 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 [33, 47, 48]. (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 it seems possible to discriminate more efficiently against some of these undesired distortions of measured spectra [49–51].

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 on 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”. 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 3.6) the 0.110 ns is the annealed-Cu lifetime, while the 0.18 ns, 14.8386% component is the estimated lifetime and intensity component for the positrons annihilating in the 0.5 mg/cm² nickel foil surrounding the source material. The 0.4594 ns, 2.9535% 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 spectra in any subsequent POSITRONFIT analyses (when the same source and similar sample material have been used for all measurements).

5 Appendix A: Fit and statistics

The first three sections of Appendix A contain general information about nonlinear least-squares (NLLS) methods and their statistical interpretations with relevance for *PALSfit*, but without going into details with the specific models involved; these are discussed in Chapter 2 and also in Appendices B and C.

The remaining sections are of a more technical nature. Section 5.4 presents essential principles of NLLS solution methods. Section 5.5 documents the separable least-squares technique which is of utmost importance for the efficiency and robustness of *PALSfit*, and Section 5.6 contains various mathematical and numerical details.

5.1 Unconstrained NLLS data fitting

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

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

$$y = f(x; b_1, b_2, \dots, b_k) = f(x; \mathbf{b}) \quad (42)$$

where x and y are the independent and dependent variable, respectively, and $\mathbf{b} = (b_j)$ is a parameter vector with k components which may enter linearly or nonlinearly in (42), and so we may talk about linear and nonlinear parameters b_j . Further, a set of n data points (x_i, y_i) ($i = 1, \dots, n$) is given, x_i being the independent and y_i the dependent variable; we shall here introduce the data vector $\mathbf{y} = (y_i)$, 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 $\mathbf{b} \in \mathbb{R}^k$ such that

$$\phi(\mathbf{b}) = \sum_{i=1}^n w_i (y_i - f(x_i; \mathbf{b}))^2 \quad (43)$$

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

When setting up equation (43) it was assumed that the x_i were fixed points corresponding to the independent variable x in (42). 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 (42) over (t_{i-1}, t_i) . Hence it is appropriate to replace (43) by

$$\phi(\mathbf{b}) = \sum_{i=1}^n w_i (y_i - f_i(\mathbf{b}))^2 \quad (44)$$

In general all we need is a “recipe” $f_i(\mathbf{b})$ to compute the model values to be compared with the data values y_i . The reformulation (44) is just a generalisation of the pointwise formulation (43) who has $f_i(\mathbf{b}) = f(x_i, \mathbf{b})$. 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 \mathbf{b} .

By introducing the matrix $\mathbf{W} = \text{diag}(w_i)$ and the n -vector $\mathbf{f}(\mathbf{b}) = (f_i(\mathbf{b}))$ we can express (44) in vector notation as follows:

$$\phi(\mathbf{b}) = \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{f}(\mathbf{b}))\|^2 \quad (45)$$

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

$$\phi_{\min} = \min_{\mathbf{b} \in \mathbb{R}^k} \{\|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{f}(\mathbf{b}))\|^2\} \quad (46)$$

A solution \mathbf{b} to (46) satisfies the gradient equation

$$\nabla\phi(\mathbf{b}) = \mathbf{0} \quad (47)$$

which is equivalent to the k scalar equations

$$\frac{\partial\phi(\mathbf{b})}{\partial b_j} = 0, \quad j = 1, \dots, k \quad (48)$$

By (44) and (48) we obtain

$$\sum_{i=1}^n w_i(y_i - f_i(\mathbf{b}))p_{ij} = 0, \quad j = 1, \dots, k \quad (49)$$

where

$$p_{ij} = \frac{\partial f_i(\mathbf{b})}{\partial b_j} \quad (50)$$

It is practical to collect the derivatives (50) in the $n \times k$ matrix

$$\mathbf{P} = (p_{ij}) \quad (51)$$

The equations (49) 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 5.4 and 5.5. Only for linear or linearized models the normal equations are linear.

It is instructive to consider the linear case in some detail. Here (42) takes the form

$$y = \sum_{j=1}^k g_j(x)b_j \quad (52)$$

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 (43) or (44) can be written

$$\phi(\mathbf{b}) = \sum_{i=1}^n w_i \left(y_i - \sum_{j=1}^k p_{ij}b_j \right)^2 \quad (53)$$

The normal equations take the classical form

$$\sum_{j'=1}^k \sum_{i=1}^n w_i p_{ij} p_{ij'} b_{j'} = \sum_{i=1}^n w_i y_i p_{ij}, \quad j = 1, \dots, k \quad (54)$$

We have $\mathbf{f}(\mathbf{b}) = \mathbf{P}\mathbf{b}$, and the equations (45–46) can be written

$$\phi(\mathbf{b}) = \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{P}\mathbf{b})\|^2 \quad (55)$$

$$\phi_{\min} = \min_{\mathbf{b} \in \mathbb{R}^k} \{\|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{P}\mathbf{b})\|^2\} \quad (56)$$

The problem (56) is solved by (54) which can be written

$$\mathbf{P}^T \mathbf{W} \mathbf{P} \mathbf{b} = \mathbf{P}^T \mathbf{W} \mathbf{y} \quad (57)$$

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

$$\mathbf{P}^T \mathbf{P} \mathbf{b} = \mathbf{P}^T \mathbf{y} \quad (58)$$

Assuming that the coefficient matrix $\mathbf{P}^T \mathbf{W} \mathbf{P}$ in (57) is nonsingular, it must be positive definite too. The same applies to $\mathbf{P}^T \mathbf{P}$ in (58). The case described in (52–58) 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 (49). Here we just assume that a usable solution \mathbf{b} can be found.

5.2 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, $h(\mathbf{b}) = 0$, as well as an inequality $h(\mathbf{b}) \geq 0$, where $h(\mathbf{b})$ is an arbitrary function of the parameter vector.

Although inequality constraints 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; in practice the sign excursions are often a simple and robust way of removing redundant parameters. 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, and are indeed used in Appendix C, Section 7.2. Otherwise, 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 5.5 we shall describe the separable least-squares technique used in *PALSfit*. 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 (52) where $\phi(\mathbf{b})$ is given by (53). 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 \mathbf{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 ($m \leq p$):

$$h_{l1}\alpha_1 + \cdots + h_{lp}\alpha_p = \gamma_l, \quad l = 1, \dots, m \quad (59)$$

In vector form (59) reads

$$\mathbf{H}\boldsymbol{\alpha} = \boldsymbol{\gamma} \quad (60)$$

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 (59) or (60) influence the NLLS procedure will be discussed in Section 5.6.

5.3 Statistical analysis

In this section we address the question of the statistical scatter in the parameters and ϕ_{\min} that can be expected in NLLS parameter estimation. In particular we are interested in the standard deviations of the parameters and in their correlation coefficients.

Covariance matrix of the parameters

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 $\mathbf{y} = (y_i)$ be given. Let us first consider the unconstrained case. Through solution of the normal equations (49) each spectrum \mathbf{y} gives rise to a parameter estimate $\mathbf{b} = \mathbf{b}(\mathbf{y})$. Hence also \mathbf{b} becomes a random (vector) variable with a certain joint distribution.

We shall use the symbol $E[\cdot]$ for expected value (ensemble mean) and $\text{Var}[\cdot]$ for variance. We

introduce the “ensemble-mean spectrum”

$$\boldsymbol{\eta} = (\eta_i) = \mathbb{E}[\mathbf{y}] \quad (61)$$

and the corresponding hypothetical estimate

$$\mathbf{b}_0 = (b_{j0}) = \mathbf{b}(\boldsymbol{\eta}) \quad (62)$$

Thus \mathbf{b}_0 is the solution of (49) corresponding to the particular spectrum (η_i) . Now, given an arbitrary spectrum (y_i) , let the corresponding parameter vector be $\mathbf{b} = (b_j)$. If we assume that $\mathbf{b} - \mathbf{b}_0 = \Delta\mathbf{b} = (\Delta b_j)$ is so small that our model is locally linear in \mathbf{b} around \mathbf{b}_0 , we have to a first-order approximation

$$f_i(\mathbf{b}) = f_i(\mathbf{b}_0) + \sum_{j=1}^k p_{ij} \Delta b_j \quad (63)$$

where p_{ij} are the derivatives (50) evaluated at \mathbf{b}_0 . We insert (63) into the normal equations (49) and obtain a linear equation system of order k with Δb_j as unknowns. In vector notation this system reads

$$\mathbf{P}^T \mathbf{W} \mathbf{P} \Delta \mathbf{b} = \mathbf{P}^T \mathbf{W} \Delta \mathbf{y} \quad (64)$$

where $\Delta \mathbf{y} = \mathbf{y} - \mathbf{f}(\mathbf{b}_0)$ has the components

$$\Delta y_i = y_i - f_i(\mathbf{b}_0), \quad i = 1, \dots, n \quad (65)$$

We note the similarity with the linear case (57). The system (64) has the solution

$$\Delta \mathbf{b} = \mathbf{K} \Delta \mathbf{y} \quad (66)$$

where

$$\mathbf{K} = (\mathbf{P}^T \mathbf{W} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{W} \quad (67)$$

The *covariance matrix* of a vector variable \mathbf{v} will here be denoted $\boldsymbol{\Sigma}(\mathbf{v})$. (Other names are dispersion matrix and variance-covariance matrix, since the diagonal row contains the component variances.) It is well-known that if two vectors \mathbf{v} and \mathbf{w} are related by a linear transformation

$$\mathbf{w} = \mathbf{A} \mathbf{v} \quad (68)$$

then

$$\boldsymbol{\Sigma}(\mathbf{w}) = \mathbf{A} \boldsymbol{\Sigma}(\mathbf{v}) \mathbf{A}^T \quad (69)$$

Our primary goal is to estimate the covariance matrix

$$\boldsymbol{\Sigma}(\mathbf{b}) = (\sigma_{jj'}) = \begin{pmatrix} \sigma_{11} & \dots & \sigma_{1k} \\ \dots & \dots & \dots \\ \sigma_{k1} & \dots & \sigma_{kk} \end{pmatrix} \quad (70)$$

for the parameter vector \mathbf{b} obtained from the NLLS. 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

$$\mathbf{R} = \begin{pmatrix} 1 & \dots & \rho_{1k} \\ \dots & \dots & \dots \\ \rho_{k1} & \dots & 1 \end{pmatrix} \quad (71)$$

by the formula

$$\rho_{jj'} = \sigma_{jj'} / (\sigma_j \sigma_{j'}) \quad (72)$$

Equation (66) shows that $\Delta \mathbf{b}$ is related to $\Delta \mathbf{y}$ by a (locally) linear transformation, and so we obtain from (69) the approximate result

$$\boldsymbol{\Sigma}(\mathbf{b}) = \mathbf{K} \boldsymbol{\Sigma}(\mathbf{y}) \mathbf{K}^T \quad (73)$$

We now assume that the measurements y_i are independent. Let

$$\text{Var}[y_i] = s_i^2, \quad i = 1, \dots, n \quad (74)$$

such that s_i is the standard deviation of y_i . Then $\mathbf{\Sigma}(\mathbf{y}) = \text{diag}(s_i^2)$. We also assume that the variances s_i^2 ($i = 1, \dots, n$) 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 $\mathbf{\Sigma}(\mathbf{b})$. By using (73) and observing that (2) implies

$$\mathbf{W}\mathbf{\Sigma}(\mathbf{y}) = \mathbf{I}_n \quad (75)$$

we obtain after reduction the formula

$$\mathbf{\Sigma}(\mathbf{b}) = (\mathbf{P}^T \mathbf{W} \mathbf{P})^{-1} \quad (76)$$

which holds at least approximately. It is exact in the linear regression case (57).

If we assume a normal distribution of y_i , then 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.

Distribution of ϕ_{\min}

Still under the assumption of a locally linear model and of statistical weighting as described, we shall next study the distribution of ϕ_{\min} in (46). Here we make the additional assumptions that we have an *ideal model*, i.e.

$$f_i(\mathbf{b}_0) = \eta_i \quad (77)$$

and that each measurement y_i has a Gaussian distribution,

$$y_i \in \mathbf{N}(\eta_i, s_i^2) \quad (78)$$

Then by (65) and (77)

$$\Delta y_i \in \mathbf{N}(0, s_i^2) \quad (79)$$

Moreover \mathbf{b} will be approximately joint Gaussian. Let $\mathbf{b} = (b_j)$ be the solution of the normal equations (49). Then we obtain the following approximate expression of ϕ_{\min} from the linear expansion (63), which is valid for small $\Delta \mathbf{b} = (b_j - b_{j0})$:

$$\phi_{\min} = \sum_{i=1}^n w_i \left(y_i - \eta_i - \sum_{j=1}^k p_{ij} (b_j - b_{j0}) \right)^2 \quad (80)$$

This can also be written

$$\phi_{\min} = \|\mathbf{W}^{1/2}(\Delta \mathbf{y} - \mathbf{P}\Delta \mathbf{b})\|^2 = (\Delta \mathbf{y} - \mathbf{P}\Delta \mathbf{b})^T \mathbf{W}(\Delta \mathbf{y} - \mathbf{P}\Delta \mathbf{b}) \quad (81)$$

with $\Delta \mathbf{y} = (y_i - \eta_i)$ and \mathbf{P} given by (50–51). By (66–67) ϕ_{\min} becomes a quadratic form in $\Delta \mathbf{y}$:

$$\phi_{\min} = \Delta \mathbf{y}^T \mathbf{B} \Delta \mathbf{y} \quad (82)$$

where \mathbf{B} is found to

$$\mathbf{B} = \mathbf{W} - \mathbf{W} \mathbf{P} (\mathbf{P}^T \mathbf{W} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{W} \quad (83)$$

Defining $u_i = \Delta y_i / s_i$, we see from (79) that u_i becomes a standardized normal variable,

$$u_i \in \mathbf{N}(0, 1) \quad (84)$$

Since the statistical weighting (2) was assumed, we have

$$\Delta \mathbf{y} = \mathbf{W}^{-1/2} \mathbf{u} \quad (85)$$

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

$$\phi_{\min} = \mathbf{u}^T \mathbf{C} \mathbf{u} \quad (86)$$

with

$$\mathbf{C} = \mathbf{W}^{-1/2} \mathbf{B} \mathbf{W}^{-1/2} = \mathbf{I}_n - \mathbf{M} \quad (87)$$

where

$$\mathbf{M} = \mathbf{W}^{\frac{1}{2}} \mathbf{P} (\mathbf{P}^T \mathbf{W} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{W}^{\frac{1}{2}} \quad (88)$$

Considering first the unconstrained case, the matrix \mathbf{M} has its full rank k . All its k nonzero eigenvalues are unity, as is easily verified by premultiplying $\mathbf{M}\mathbf{x} = \lambda\mathbf{x}$ by $\mathbf{P}^T \mathbf{W}^{1/2}$. Thus \mathbf{C} is of rank $n - k$ with $n - k$ unity eigenvalues, and so there is an orthogonal substitution $\mathbf{u} = \mathbf{Q}\mathbf{z}$ which transforms ϕ_{\min} into a sum of $f = n - k$ squares:

$$\phi_{\min} = \sum_{i=1}^{n-k} z_i^2 \quad (89)$$

where the z_i are independent, and each $z_i \in \mathbf{N}(0, 1)$. This means that ϕ_{\min} has a χ^2 -distribution with $f = n - k$ degrees of freedom,

$$\phi_{\min} \in \chi^2(f) \quad (90)$$

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

$$\chi^2 = \phi_{\min} = \min_{\mathbf{b} \in \mathbb{R}^k} \sum_{i=1}^n \left(\frac{y_i - f_i(\mathbf{b})}{s_i} \right)^2 \quad (91)$$

Influence of linear constraints on the statistics

The results derived for $\Sigma(\mathbf{b})$ and ϕ_{\min} are independent of the applied fitting technique. But we have assumed an unconstrained variation of all components of the k -vector \mathbf{b} . If there are m independent linear constraints on the parameters, then the analysis still holds good for a “basic subset” of k_{free} independent parameter components, where

$$k_{\text{free}} = k - m \quad (92)$$

In this case we obtain $\Sigma(\mathbf{b})$ by incorporating the linear constraints (59) or (60) and expressing the remaining components (deterministically) in terms of the free ones. Consequently, (90) is still valid, but now we have

$$f = n - k_{\text{free}} \quad (93)$$

Details of these operations are found in Section 5.6.

Covariance matrix for any vector that is expressed in terms of \mathbf{b}

Suppose now that the solution vector $\mathbf{b} \in \mathbb{R}^k$ is applied to calculate a new vector $\mathbf{b}' = \mathbf{b}'(\mathbf{b}) \in \mathbb{R}^{k'}$. Then the covariance matrix of \mathbf{b}' is approximately

$$\Sigma(\mathbf{b}') = \mathbf{J} \Sigma(\mathbf{b}) \mathbf{J}^T \quad (94)$$

where

$$\mathbf{J} = d\mathbf{b}'/d\mathbf{b} \quad (95)$$

is the $k' \times k$ Jacobian matrix of the transformation. This follows from the differential relation

$$db'_j = \frac{\partial b'_j}{\partial b_1} db_1 + \dots + \frac{\partial b'_j}{\partial b_k} db_k, \quad j = 1, \dots, k' \quad (96)$$

together with (69). In *PALSfit* we use rather simple parameter transformations when passing from \mathbf{b} to \mathbf{b}' , 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 $I_j(\delta_{jj'} - I_j)/J_j$, $\delta_{jj'}$ being the Kronecker delta.

Standard deviation of a derived parameter

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

$$\sigma_z^2 = \sum_{j=1}^k \left(\frac{\partial z}{\partial b_j} \right)^2 \sigma_j^2 + 2 \sum_{j=1}^{k-1} \sum_{j'=j+1}^k \frac{\partial z}{\partial b_j} \frac{\partial z}{\partial b_{j'}} \rho_{jj'} \sigma_j \sigma_{j'} \quad (97)$$

This result follows by the transformation rule (94).

Forcing a single parameter to be shifted

There is another property of the correlation matrix which might be useful. 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

$$\Delta b_j = (\sigma_j / \sigma_1) \rho_{1j} \Delta b_1, \quad j = 2, \dots, k \quad (98)$$

The formula (98) 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 $\Delta b_1 \approx 2-3 \times \sigma_1$ for well-defined fitting problems with small σ_j . For fits with large σ_j it seems to be valid only up to $\Delta b_1 \approx 0.1-0.2 \times \sigma_1$, and in certain pathological cases it fails completely; such failures may be ascribed to imperfect models or strong nonlinearities. We shall here give a proof of (98). In the following we consider $\phi(\mathbf{b})$ with fixed spectrum (y_i). We fix Δb_1 and seek the conditional minimum when the other parameters vary. We shall use the approximative formula (112) (proved in Section 5.4) with $\mathbf{d} = \Delta \mathbf{b}$:

$$\phi(\mathbf{b} + \Delta \mathbf{b}) \approx \phi(\mathbf{b}) + \nabla \phi(\mathbf{b}) \cdot \Delta \mathbf{b} + \Delta \mathbf{b}^T \mathbf{P}^T \mathbf{W} \mathbf{P} \Delta \mathbf{b} \quad (99)$$

We introduce the vector \mathbf{z} with components $\Delta b_2, \dots, \Delta b_k$. The gradient term in (99) can be written

$$\nabla \phi(\mathbf{b}) \cdot \Delta \mathbf{b} = \frac{\partial \phi}{\partial b_1} \Delta b_1 + \nabla_z \phi(\mathbf{b}) \cdot \mathbf{z} \quad (100)$$

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

$$\mathbf{P}^T \mathbf{W} \mathbf{P} = \begin{pmatrix} a_{11} & \mathbf{d}^T \\ \mathbf{d} & \mathbf{C} \end{pmatrix} \quad (101)$$

(99) can then be written

$$\phi(\mathbf{b} + \Delta \mathbf{b}) = \phi(\mathbf{b}) + \frac{\partial \phi}{\partial b_1} \Delta b_1 + a_{11} (\Delta b_1)^2 + 2 \Delta b_1 \mathbf{z}^T \mathbf{d} + \mathbf{z}^T \mathbf{C} \mathbf{z} \quad (102)$$

To minimize (102) we take the derivative with respect to \mathbf{z} . After equating the result to zero we deduce that

$$\mathbf{z} = -\Delta b_1 \mathbf{C}^{-1} \mathbf{d} \quad (103)$$

Next we make the similar partitioning

$$\mathbf{\Sigma}(\mathbf{b}) = \begin{pmatrix} \sigma_{11} & \mathbf{s}^T \\ \mathbf{s} & \mathbf{\Gamma} \end{pmatrix} \quad (104)$$

We assume statistical weighting which implies the identity $\mathbf{P}^T \mathbf{W} \mathbf{P} \mathbf{\Sigma}(\mathbf{b}) = \mathbf{I}_k$, cf. (76). From this we infer that

$$\mathbf{C}^{-1} \mathbf{d} = -\frac{1}{\sigma_{11}} \mathbf{s} \quad (105)$$

Inserting this in (103) yields

$$\mathbf{z} = \frac{\Delta b_1}{\sigma_{11}} \mathbf{s} \quad (106)$$

which proves formula (98) since \mathbf{s} has components $\sigma_{21}, \dots, \sigma_{k1}$.

The χ^2 test for goodness of fit

We saw that $\chi^2 = \phi_{\min}$ under certain assumptions has a χ^2 -distribution with $f = n - k_{\text{free}} = n - (k - m)$ degrees of freedom, where m is the number of constraints and $k_{\text{free}} = k - m$ is the effective number of free parameters in the estimation. The mean and variance of $\phi_{\min} = \chi^2$ are

$$\text{E}[\chi^2] = f \tag{107}$$

and

$$\text{Var}[\chi^2] = 2f \tag{108}$$

From the χ^2 statistics one can derive a “goodness-of-fit” significance test for the validity of the asserted ideal model, cf. (77). In such a χ^2 -test we compute the probability $P\{\chi^2 < \chi_{\text{obs}}^2\}$ that a χ^2 -distributed variable with f degrees of freedom will not exceed the observed value χ_{obs}^2 . 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

$$V = \chi^2/f, \tag{109}$$

with mean

$$\text{E}[V] = 1 \tag{110}$$

and variance

$$\text{Var}[V] = 2/f \tag{111}$$

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.

Statistical assumptions in the NLLS analysis

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. $\text{Var}[y_i]$ small.
2. Our model is only weakly nonlinear in the parameter vector \mathbf{b} .
3. An ideal model which means that (77) 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 $\text{Var}[y_i]$ 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 (63).

Assumption 1 is a fair approximation in *PALSfit* 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 $\text{Var}[y_i]$ (see also Assumption 7).

Regarding Assumption 7, the theoretical values of the population variances $\text{Var}[y_i]$ 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.

5.4 Marquardt minimization

As mentioned in Section 5.1 the normal equations (49) are in general nonlinear and must be solved iteratively. We now describe such an iterative method called Marquardt's principle, which is a combination of two classical unconstrained minimization procedures; constraints will be taken care of as described in Sections 5.2 and 5.6.

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 \mathbf{d} is small, the fit is good, and the model locally linear:

$$\phi(\mathbf{b} + \mathbf{d}) \approx \phi(\mathbf{b}) + \nabla\phi(\mathbf{b}) \cdot \mathbf{d} + \mathbf{d}^T \mathbf{P}^T \mathbf{W} \mathbf{P} \mathbf{d} \quad (112)$$

with the usual meaning of \mathbf{W} and \mathbf{P} . We use Taylor's quadratic limit formula

$$\phi(\mathbf{b} + \mathbf{d}) = \phi(\mathbf{b}) + \nabla\phi(\mathbf{b}) \cdot \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{S} \mathbf{d} + o(\|\mathbf{d}\|^2) \quad (113)$$

where $\mathbf{S} = \{s_{jj'}\}$ is the Hessian matrix of $\phi(\mathbf{b})$. From the expression (44) we find

$$s_{jj'} = \frac{\partial^2 \phi(\mathbf{b})}{\partial b_j \partial b_{j'}} = 2 \sum_{i=1}^n w_i \left(\frac{\partial f_i}{\partial b_j} \frac{\partial f_i}{\partial b_{j'}} - (y_i - f_i) \frac{\partial^2 f_i}{\partial b_j \partial b_{j'}} \right) \quad (114)$$

with $f_i = f_i(\mathbf{b})$. 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 (114). 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

$$\mathbf{S} = 2\mathbf{P}^T \mathbf{W} \mathbf{P} \quad (115)$$

Inserting this in (113) we establish (112). Returning to Newton's method, let \mathbf{b} be a guessed or previously iterated parameter vector. Newton's correction step \mathbf{d} now solves the local minimization problem

$$\min_{\mathbf{d} \in \mathbb{R}^k} \{\phi(\mathbf{b} + \mathbf{d})\} \quad (116)$$

or

$$\nabla\phi(\mathbf{b} + \mathbf{d}) = \mathbf{0} \quad (117)$$

for fixed \mathbf{b} . We approximate $\phi(\mathbf{b} + \mathbf{d})$ by (112). For brevity we shall write

$$\mathbf{A} = \mathbf{P}^T \mathbf{W} \mathbf{P} \quad (118)$$

Assuming that \mathbf{P} has full rank, \mathbf{A} will be positive definite. We shall need the formula

$$\nabla(\nabla\phi(\mathbf{b}) \cdot \mathbf{d}) \approx 2\mathbf{A} \mathbf{d} \quad (119)$$

which is proved by applying the same approximation as in (114). Then we obtain

$$\nabla\phi(\mathbf{b}) + 2\mathbf{A}\mathbf{d} + \mathcal{O}(\|\mathbf{d}\|^2) = \mathbf{0} \quad (120)$$

Thus the Newton step \mathbf{d} is computed from the normal equation system

$$\mathbf{A}\mathbf{d} = \mathbf{g} \quad (121)$$

where

$$\mathbf{g} = -\frac{1}{2}\nabla\phi(\mathbf{b}) \quad (122)$$

According to (44) the components of \mathbf{g} are

$$g_j = \sum_{i=1}^n w_i (y_i - f_i(\mathbf{b})) p_{ij}, \quad j = 1, \dots, k \quad (123)$$

(We see that the normal equations (49) are satisfied when $\mathbf{g} = \mathbf{0}$.) Subsequently $\mathbf{b} + \mathbf{d}$ replaces \mathbf{b} as the new iterate, and the iterations continue. With the pure Newton method we cannot guarantee that the new $\phi = \phi(\mathbf{b} + \mathbf{d})$ 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 modify (121) as follows,

$$(\mathbf{A} + \lambda\mathbf{D})\mathbf{d} = \mathbf{g} \quad (124)$$

where \mathbf{D} is a diagonal matrix with the same diagonal row as the positive definite matrix \mathbf{A} . This is Marquardt's equation [52]. Written out in full, we see from (118) and (123) that it takes the form:

$$(\mathbf{P}^T\mathbf{W}\mathbf{P} + \lambda\mathbf{D})\mathbf{d} = \mathbf{P}^T\mathbf{W}(\mathbf{y} - \mathbf{f}(\mathbf{b})) \quad (125)$$

where $\mathbf{y} = (y_i)$ and $\mathbf{f}(\mathbf{b}) = (f_i(\mathbf{b}))$. 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. (121). On the other hand, when $\lambda \rightarrow \infty$ we obtain a solution vector proportional to $\mathbf{D}^{-1}\mathbf{g}$. According to (122) \mathbf{g} is proportional to the negative gradient vector $-\nabla\phi$, so $\mathbf{D}^{-1}\mathbf{g}$ 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

$$(\mathbf{A}^{(r)} + \lambda^{(r)}\mathbf{D}^{(r)})\mathbf{d}^{(r)} = \mathbf{g}^{(r)} \quad (126)$$

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

$$\mathbf{b}^{(r+1)} = \mathbf{b}^{(r)} + \mathbf{d}^{(r)} \quad (127)$$

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

$$\phi^{(r+1)} \leq \phi^{(r)} \quad (128)$$

If we are not already at the minimum, it is always possible to satisfy (128) 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 $\mathbf{b}^{(r)}$ are stationary with increasing r . The algorithm is sometimes called the Levenberg-Marquardt method (LM) since Levenberg [53] already in 1944 put forward essential parts of the ideas taken up by Marquardt in 1963 [52].

Over the years LM has undergone a number of refinements, adding more robustness to it. In earlier versions of *PALSfit* we used LM as in [52]. But pure LM puts no bounds on the step vector $\mathbf{d}(\lambda)$. Newer LM implementations use a "trust-region" enhancement and replace the unrestricted minimization of ϕ by the quadratic programming problem

$$\min_{\mathbf{d}(\lambda)} \left\{ \phi : \|\mathbf{D}^{1/2}\mathbf{d}(\lambda)\| \leq \Delta \right\} \quad (129)$$

where \mathbf{D} is the matrix entering (124). The effect of (129) is to restrict the size of $\mathbf{d} = \mathbf{d}(\lambda)$. The bound Δ is adjusted each time a *major* iteration step begins and is decreased if the solution of (129) does not provide a suitable correction. We have adopted this idea for use in *PALSfit* from the work of Moré [54], as implemented in the software package MINPACK-1 [55] for unconstrained least-squares minimization. A subroutine from this package, LMPAR, performs *minor* iterations by finding a value of λ that solves (129) 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 [54].

5.5 Separable least-squares technique

An efficiency gain in solving the NLLS problem (46) can be obtained when some of the parameters b_j enter the model $\mathbf{f}(\mathbf{b}) = (f_i(\mathbf{b}))$ linearly. Indeed this is the case in *PALSfit*. The least-squares problem is then called *separable* (or semilinear). Separable procedures have been studied by several authors [3, 56–60] and have proved successful in this work and in many other applications as well. They provide a decomposition of the space spanned by the model parameters. Thus we can rearrange the parameters to comply with the partitioning of

$$\mathbf{b} = \begin{matrix} p \\ q \end{matrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \quad (130)$$

into a “linear” p -vector $\boldsymbol{\alpha} = (\alpha_j)$ and a “nonlinear” q -vector $\boldsymbol{\beta} = (\beta_j)$, where $p + q = k$. The model functions $f_i(\mathbf{b})$ can then be written

$$f_i(\mathbf{b}) = \sum_{j=1}^p \alpha_j f_{ij}(\boldsymbol{\beta}) = (\mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha})_i, \quad i = 1, \dots, n \quad (131)$$

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

$$\mathbf{f}(\mathbf{b}) = \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha} \quad (132)$$

In separable NLLS we consider the linear subproblems of (46) where $\boldsymbol{\beta}$ is fixed and $\boldsymbol{\alpha}$ varies:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^p} \{ \phi(\boldsymbol{\alpha}) = \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha})\|^2 : \boldsymbol{\beta} \text{ fixed} \} \quad (133)$$

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

$$\mathbf{F}^T \mathbf{W} \mathbf{F} \boldsymbol{\alpha} = \mathbf{F}^T \mathbf{W} \mathbf{y} \quad (134)$$

cf. the linear regression case (52–58). Turning to the determination of the nonlinear part $\boldsymbol{\beta}$ of the parameter vector \mathbf{b} , we realize that an iterative method is needed. In fact, there will be an outer loop, where each step provides a correction vector \mathbf{d} to $\boldsymbol{\beta}$, and an inner procedure which invokes a linear minimization (133–134) each time a new trial value of $\boldsymbol{\beta}$ is chosen. We can formulate the nonlinear outer minimization as follows:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^q} \{ \phi(\boldsymbol{\beta}) = \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha}(\boldsymbol{\beta}))\|^2 \} \quad (135)$$

We solve (135) by a modified Marquardt procedure as explained in Section 5.4, where \mathbf{b} should be replaced by $\boldsymbol{\beta}$. Indeed, equation (125) takes the form

$$(\mathbf{P}^T \mathbf{W} \mathbf{P} + \lambda \mathbf{D})\mathbf{d} = \mathbf{P}^T \mathbf{W}(\mathbf{y} - \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha}(\boldsymbol{\beta})) \quad (136)$$

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

$$p_{ij} = \frac{\partial f_i}{\partial \beta_j} \quad (137)$$

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

A crucial point in the separable procedure is the evaluation of (137), which can be accomplished by considering the vector $\mathbf{f} = \mathbf{f}(\boldsymbol{\beta})$ in (132). Note that \mathbf{f} depends on $\boldsymbol{\beta}$ directly through \mathbf{F} and indirectly through $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\boldsymbol{\beta})$; hence

$$\frac{\partial \mathbf{f}}{\partial \beta_j} = \frac{\partial \mathbf{F}}{\partial \beta_j} \boldsymbol{\alpha} + \mathbf{F} \frac{\partial \boldsymbol{\alpha}}{\partial \beta_j} \quad (138)$$

To find $\partial \boldsymbol{\alpha} / \partial \beta_j$ we take the derivative of both members of (134). This leads to

$$\mathbf{F}^T \mathbf{W} \mathbf{F} \frac{\partial \boldsymbol{\alpha}}{\partial \beta_j} = \frac{\partial \mathbf{F}^T}{\partial \beta_j} \mathbf{W} (\mathbf{y} - \mathbf{F} \boldsymbol{\alpha}) - \mathbf{F}^T \mathbf{W} \frac{\partial \mathbf{F}}{\partial \beta_j} \boldsymbol{\alpha}, \quad j = 1, \dots, q \quad (139)$$

For an ideal model the term in (139) containing the residual vector $\mathbf{y} - \mathbf{F} \boldsymbol{\alpha}$ 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 Moré bound Δ . For each outer iteration, solve the linear subproblem (133–134) for $\boldsymbol{\alpha}$ and calculate ϕ . Compute the Jacobian elements $\partial \mathbf{f} / \partial \beta_j$ from (139) and (138), and form \mathbf{P} and \mathbf{D} . Then enter an inner procedure and find near-optimal values of λ and the correction vector to $\boldsymbol{\beta}$, $\mathbf{d} = \mathbf{d}(\lambda)$, using Marquardt’s method with Moré’s modification. Update the bound Δ , replace $\boldsymbol{\beta}$ by $\boldsymbol{\beta} + \mathbf{d}$, and resume the outer iteration loop. The procedure is finished when convergence is obtained.

When implementing our separable algorithm, there is a practical difficulty in handling $\partial \mathbf{f} / \partial \beta_j$ in (138). 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 *PALSfit*, are readily integrated in the separable NLS procedure, cf. Sections 5.2 and 5.6.

5.6 Various mathematics, statistics, and numerics

In this section a number of technical details are collected. They all have relevance to previously discussed subjects.

Estimation of background and weight smoothing

Considering Assumptions 6 and 7 in Section 5.3 we face the problem that we do not know $\sigma_i^2 = \text{Var}[y_i]$ when setting up the statistical weighting (2). Each count y_i is distributed in a Poisson distribution with a certain mean value η_i ,

$$y_i \in \text{P}(\eta_i) \quad (140)$$

implying that

$$\text{E}[y_i] = \eta_i \quad (141)$$

$$\text{Var}[y_i] = \eta_i \quad (142)$$

The ideal weighting

$$w_i = \frac{1}{\eta_i} \quad (143)$$

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

$$w_i = \frac{1}{y_i} \quad (144)$$

or rather use the modified formula

$$w_i = \frac{1}{\max(y_i, 1)} \quad (145)$$

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

$$P\{y_i = 0\} \approx 0 \quad (146)$$

The “raw” weights (144) will normally fluctuate, and we shall now show that this may induce a bias on the estimate of the background. In a fitting model like (3) with a free background B it is often the case that the major contribution to (1) comes from channels where the impact from other fitting parameters are negligible compared to B . This implies that B is virtually independent of the other parameters, such that obtaining a least-squares estimate B^* of B is essentially a one-parameter problem,

$$B^* = \operatorname{argmin}_{B \in \mathbb{R}} \sum_{i=1}^n w_i (y_i - B)^2 \quad (147)$$

where n should be properly adjusted. Assuming this situation, we find the unique solution

$$B^* = \frac{\sum w_i y_i}{\sum w_i} \quad (148)$$

In the “theoretical” case (143) we evaluate $B^* = B_0^*$ by inserting (143) in (148). Because B is the only parameter, we have $\eta_i = B$ and then

$$B_0^* = \frac{\sum y_i}{n} = \langle y_i \rangle \quad (149)$$

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

$$\mathbb{E}[B_0^*] = \mathbb{E}[y_i] = \eta_i = B \quad (150)$$

On the other hand, in the “real” case (144), we evaluate $B^* = B_1^*$ by inserting (144) in (148), 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,

$$\mathbb{E}[B_1^*] \approx B - 1 \quad (151)$$

under the additional assumptions that n as well as the counts y_i and the background B are reasonably large. We then obtain

$$B_1^* = \frac{n}{\sum \frac{1}{y_i}} = \frac{1}{\langle \frac{1}{y_i} \rangle} \quad (152)$$

and

$$\mathbb{E}[B_1^*] \approx \frac{1}{\mathbb{E}[\frac{1}{y_i}]} \quad (153)$$

From the Poisson distribution (140) we have the probability

$$P\{y_i = k\} = \frac{B^k}{k!} e^{-B} \quad (154)$$

and, taking (146) into account, this gives

$$\mathbb{E}\left[\frac{1}{y_i}\right] \approx \sum_{k=1}^{\infty} \frac{1}{k} \frac{B^k}{k!} e^{-B} \quad (155)$$

This sum can be evaluated with the result

$$\mathbb{E}\left[\frac{1}{y_i}\right] \approx e^{-B} (\operatorname{Ei}(B) - \gamma - \log B) \quad (156)$$

where Ei is the exponential integral defined by

$$\operatorname{Ei}(x) = - \int_{-x}^{\infty} \frac{e^{-t}}{t} dt \quad (\text{Cauchy principal value when } x > 0) \quad (157)$$

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

$$\text{Ei}(B) \sim \frac{e^B}{B} \left\{ 1 + \sum_{r=1}^{\infty} \frac{r!}{B^r} \right\}, \quad B \rightarrow +\infty \quad (158)$$

We shall use the first-order approximand, and discarding γ and $\log B$ in (156) we then obtain

$$\text{E} \left[\frac{1}{y_i} \right] \approx \frac{1}{B} \left(1 + \frac{1}{B} \right) \quad (159)$$

Finally (153) gives the approximate result (151) that the bias is -1 . Monte Carlo simulations working with the Poisson process confirms (151).

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 POSITRON-FIT.) Between the two iteration cycles the weight fluctuations were removed by replacing (144) by the weights

$$w_i = \frac{1}{y_i^f} \quad (160)$$

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.

Later PALS*fit* versions use 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 [61]). 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_{ch}^{\min} to i_{ch}^{\max} , cf. (13). 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.

This 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 becomes more robust against some extreme choices of input parameters to the analysis. Moreover the reliability of the statistical analysis for imperfect fits is improved.

Implementation of linear constraints

We consider the constrained linear least-squares problem (cf. (59–60)),

$$\phi_{\min} = \min_{\alpha \in \mathbb{R}^p} \{ \phi(\alpha) = \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{F}\alpha)\|^2 : \mathbf{H}\alpha = \boldsymbol{\gamma} \} \quad (161)$$

This problem is a constrained version of the separable NLLS subproblem in (133), where an optimal linear parameter vector $\alpha \in \mathbb{R}^p$ was computed for a given nonlinear parameter vector $\boldsymbol{\beta} \in \mathbb{R}^q$. Thus the derivative matrix $\mathbf{F} = (\partial f_i / \partial \alpha_j)$ is of size $n \times p$. One way of handling this

constraint problem would be to use Lagrange multipliers. This method was used in early versions of POSITRONFIT. As a result, the unconstrained normal equation system (cf. (57)),

$$\mathbf{F}^T \mathbf{W} \mathbf{F} \boldsymbol{\alpha} = \mathbf{F}^T \mathbf{W} \mathbf{y} \quad (162)$$

was extended to a block matrix system

$$\begin{pmatrix} \mathbf{F}^T \mathbf{W} \mathbf{F} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \mathbf{F}^T \mathbf{W} \mathbf{y} \\ \boldsymbol{\gamma} \end{pmatrix} \quad (163)$$

where the vector $2\boldsymbol{\mu}$ contains the Lagrange multipliers. Although (163) is simple enough, there are some drawbacks in this procedure. The coefficient matrix in (163) is not positive definite as is $\mathbf{F}^T \mathbf{W} \mathbf{F}$. This follows by considering the quadratic form Q associated with LHS(163),

$$Q = (\mathbf{F}\boldsymbol{\alpha})^T \mathbf{W} \mathbf{F} \boldsymbol{\alpha} + 2\boldsymbol{\mu} \cdot \boldsymbol{\gamma} \quad (164)$$

which may become negative. As a result the numerical stability of the calculations might be degraded. 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 PALSfit. 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 move these columns to the m first positions. This can be expressed in terms of a p th-order permutation matrix $\mathbf{\Pi}_p$ by

$$\mathbf{H}\mathbf{\Pi}_p = \mathbf{H}' = \begin{matrix} & m & p-m \\ \begin{pmatrix} \mathbf{B} & \mathbf{N} \end{pmatrix} \end{matrix} \quad (165)$$

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 $\mathbf{\Pi}_p$ is orthogonal, $\mathbf{\Pi}_p \mathbf{\Pi}_p^T = \mathbf{I}_p$, (60) can be written

$$\mathbf{H}' \boldsymbol{\alpha}' = \boldsymbol{\gamma} \quad (166)$$

with

$$\boldsymbol{\alpha}' = \mathbf{\Pi}_p^T \boldsymbol{\alpha} \quad (167)$$

The homogeneous equation $\mathbf{H}' \boldsymbol{\alpha}' = \mathbf{0}$ corresponding to (166) has the complete solution

$$\boldsymbol{\alpha}' = \mathbf{Y}' \mathbf{t}, \quad \mathbf{t} \in \mathbb{R}^{p-m} \quad (168)$$

where

$$\mathbf{Y}' = \begin{matrix} & p-m \\ \begin{matrix} m \\ p-m \end{matrix} \begin{pmatrix} -\mathbf{B}^{-1} \mathbf{N} \\ \mathbf{I}_{p-m} \end{pmatrix} \end{matrix} \quad (169)$$

A particular solution to (166) is

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha}'_0 = \begin{matrix} m \\ p-m \end{matrix} \begin{pmatrix} \mathbf{B}^{-1} \boldsymbol{\gamma} \\ \mathbf{0} \end{pmatrix} \quad (170)$$

Hence the complete solution of (166) is

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha}'_0 + \mathbf{Y}' \mathbf{t}, \quad \mathbf{t} \in \mathbb{R}^{p-m} \quad (171)$$

Finally from (167) we get the complete solution of (60):

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_0 + \mathbf{Y} \mathbf{t}, \quad \mathbf{t} \in \mathbb{R}^{p-m} \quad (172)$$

where

$$\boldsymbol{\alpha}_0 = \mathbf{\Pi}_p \boldsymbol{\alpha}'_0 \quad (173)$$

and

$$\mathbf{Y} = \mathbf{\Pi}_p \mathbf{Y}' \quad (174)$$

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

$$\mathbf{\Pi}_p = \begin{matrix} & m & p-m \\ p & (\mathbf{\Pi}_1 & \mathbf{\Pi}_2) \end{matrix} \quad (175)$$

Then (173) becomes

$$\boldsymbol{\alpha}_0 = \mathbf{\Pi}_1 \mathbf{B}^{-1} \boldsymbol{\gamma} \quad (176)$$

To express (174) we note that

$$\mathbf{N} = \mathbf{H} \mathbf{\Pi}_2 \quad (177)$$

and so

$$\mathbf{Y} = (\mathbf{I}_p - \mathbf{\Pi}_1 \mathbf{B}^{-1} \mathbf{H}) \mathbf{\Pi}_2 \quad (178)$$

Moreover

$$\mathbf{B} = \mathbf{H} \mathbf{\Pi}_1 \quad (179)$$

Then, by (165) and (174) we have

$$\mathbf{H} \mathbf{Y} = \mathbf{H}' \mathbf{\Pi}_p^T \mathbf{\Pi}_p \mathbf{Y}' = \mathbf{H}' \mathbf{Y}' = \mathbf{0} \quad (180)$$

and the columns of \mathbf{Y} form a basis of the null space or kernel of \mathbf{H} . Using (172) we can reformulate the constrained p -dimensional problem (161) to an unconstrained $(p - m)$ -dimensional problem:

$$\phi_{\min} = \min_{\mathbf{t} \in \mathbb{R}^{p-m}} \{ \|\mathbf{W}^{1/2}(\mathbf{y} - \mathbf{F} \boldsymbol{\alpha}_0 - \mathbf{F} \mathbf{Y} \mathbf{t})\|^2 \} \quad (181)$$

We see that this can be derived from (161) by removing the constraints, substituting $p - m$ for p , $\mathbf{y} - \mathbf{F} \boldsymbol{\alpha}_0$ for \mathbf{y} , $\mathbf{F} \mathbf{Y}$ for \mathbf{F} , and \mathbf{t} for $\boldsymbol{\alpha}$. Thus we can immediately write down the normal equation system for (181) by making the corresponding substitutions in (162):

$$(\mathbf{F} \mathbf{Y})^T \mathbf{W} (\mathbf{F} \mathbf{Y}) \mathbf{t} = (\mathbf{F} \mathbf{Y})^T \mathbf{W} (\mathbf{y} - \mathbf{F} \boldsymbol{\alpha}_0) \quad (182)$$

Turning to our seperable setup, this equation replaces (134). Equation (138) becomes

$$\frac{\partial \mathbf{f}}{\partial \beta_j} = \frac{\partial \mathbf{F}}{\partial \beta_j} \boldsymbol{\alpha} + \mathbf{F} \mathbf{Y} \frac{\partial \mathbf{t}}{\partial \beta_j} \quad (183)$$

while the counterpart to (139) is obtained by deriving (182) and using (172). The result is

$$(\mathbf{F} \mathbf{Y})^T \mathbf{W} (\mathbf{F} \mathbf{Y}) \frac{\partial \mathbf{t}}{\partial \beta_j} = \frac{\partial (\mathbf{F} \mathbf{Y})^T}{\partial \beta_j} \mathbf{W} (\mathbf{y} - \mathbf{F} \boldsymbol{\alpha}) - (\mathbf{F} \mathbf{Y})^T \mathbf{W} \frac{\partial \mathbf{F}}{\partial \beta_j} \boldsymbol{\alpha} \quad (184)$$

Next we shall derive an expression for the covariance matrix $\boldsymbol{\Sigma}(\mathbf{b})$ of the total parameter vector $\mathbf{b} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ when the constraints (59) or (60) are included. Realizing that $\boldsymbol{\Sigma}(\mathbf{b})$ is independent of the actual fitting method, we can estimate it by perturbing the solution vector \mathbf{b} at the end of iterations. From the normal equation system (182) we deduce in analogy with (76) that

$$\boldsymbol{\Sigma}(\mathbf{t}) = \{ (\mathbf{F} \mathbf{Y})^T \mathbf{W} (\mathbf{F} \mathbf{Y}) \}^{-1} \quad (185)$$

Let \mathbf{P} denote the $n \times k$ matrix containing the derivatives $\partial f_i / \partial b_j$ with respect to *all* the $p + q$ components of $\mathbf{b} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$. We may then make the partition

$$\mathbf{P} = \begin{matrix} & p & q \\ n & (\mathbf{F} & \mathbf{P}_\beta) \end{matrix} \quad (186)$$

We note that

$$\begin{matrix} & p-m & q \\ n & (\mathbf{F} \mathbf{Y} & \mathbf{P}_\beta) \end{matrix} = \mathbf{P} \mathbf{Z} \quad (187)$$

where \mathbf{Z} is given by

$$\mathbf{Z} = \begin{matrix} & p-m & q \\ p & (\mathbf{Y} & \mathbf{0}) \\ q & (\mathbf{0} & \mathbf{I}_q) \end{matrix} \quad (188)$$

This means that (185) can be extended from \mathbf{t} to $(\mathbf{t}, \boldsymbol{\beta})$ as follows:

$$\boldsymbol{\Sigma}(\mathbf{t}, \boldsymbol{\beta}) = \{(\mathbf{P}\mathbf{Z})^T \mathbf{W}(\mathbf{P}\mathbf{Z})\}^{-1} \quad (189)$$

Furthermore, since

$$\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha}_0 \\ \mathbf{0} \end{pmatrix} + \mathbf{Z} \begin{pmatrix} \mathbf{t} \\ \boldsymbol{\beta} \end{pmatrix} \quad (190)$$

we obtain the result

$$\boldsymbol{\Sigma}(\mathbf{b}) = \mathbf{Z}\{(\mathbf{P}\mathbf{Z})^T \mathbf{W}(\mathbf{P}\mathbf{Z})\}^{-1} \mathbf{Z}^T \quad (191)$$

Scaling in separable NLLS

The numerical solution of many of the linear-algebraic and optimization subproblems in our algorithm is accomplished by software from the standard packages LINPACK [62] and MINPACK-1 [55]. To accommodate application of this software we found it convenient to rescale the NLLS problem formulation. We shall recast the original minimization problem (46) to

$$\phi_{\min} = \min_{\mathbf{b} \in \mathbb{R}^k} \{\|\mathbf{r}(\mathbf{b})\|^2\} \quad (192)$$

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

$$r_i = w_i^{1/2}(y_i - f_i(\mathbf{b})) \quad (193)$$

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

$$\mathbf{z} = \mathbf{W}^{1/2} \mathbf{y} \quad (194)$$

$$\mathbf{e} = \mathbf{W}^{1/2} \mathbf{f} \quad (195)$$

$$\mathbf{E} = \mathbf{W}^{1/2} \mathbf{F} \quad (196)$$

$$\mathbf{G} = \mathbf{W}^{1/2} \mathbf{P} \quad (197)$$

Then the counterparts of (132–136) become:

$$\mathbf{e}(\boldsymbol{\beta}) = \mathbf{E}(\boldsymbol{\beta}) \boldsymbol{\alpha} \quad (198)$$

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^p} \{\phi(\boldsymbol{\alpha}) = \|\mathbf{z} - \mathbf{E}(\boldsymbol{\beta}) \boldsymbol{\alpha}\|^2 : \boldsymbol{\beta} \text{ fixed}\} \quad (199)$$

$$\mathbf{E}^T \mathbf{E} \boldsymbol{\alpha} = \mathbf{E}^T \mathbf{z} \quad (200)$$

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^q} \{\phi(\boldsymbol{\beta}) = \|\mathbf{z} - \mathbf{E}(\boldsymbol{\beta}) \boldsymbol{\alpha}(\boldsymbol{\beta})\|^2\} \quad (201)$$

$$(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{D}) \mathbf{d} = \mathbf{G}^T (\mathbf{z} - \mathbf{E}(\boldsymbol{\beta}) \boldsymbol{\alpha}) \quad (202)$$

Moreover (138–139) are replaced by

$$\frac{\partial \mathbf{e}}{\partial \beta_j} = \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha} + \mathbf{E} \frac{\partial \boldsymbol{\alpha}}{\partial \beta_j} \quad (203)$$

$$\mathbf{E}^T \mathbf{E} \frac{\partial \boldsymbol{\alpha}}{\partial \beta_j} = \frac{\partial \mathbf{E}^T}{\partial \beta_j} (\mathbf{z} - \mathbf{E} \boldsymbol{\alpha}) - \mathbf{E}^T \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha} \quad (204)$$

while the corresponding equations (183–184) with constraints are replaced by

$$\frac{\partial \mathbf{e}}{\partial \beta_j} = \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha} + \mathbf{E} \mathbf{Y} \frac{\partial \mathbf{t}}{\partial \beta_j} \quad (205)$$

$$(\mathbf{E} \mathbf{Y})^T \mathbf{E} \mathbf{Y} \frac{\partial \mathbf{t}}{\partial \beta_j} = \frac{\partial (\mathbf{E} \mathbf{Y})^T}{\partial \beta_j} (\mathbf{z} - \mathbf{E} \boldsymbol{\alpha}) - (\mathbf{E} \mathbf{Y})^T \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha} \quad (206)$$

We see that the effect of these transformations is to “hide” the weights w_i entirely.

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 $\mathbf{E}^T\mathbf{E}$ in (200) which is just used here as an illustrative example. Instead we use a procedure based on the so-called QR decomposition of the $n \times p$ matrix \mathbf{E} , viz.

$$\mathbf{E} = \mathbf{QR} \quad (207)$$

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 [62]). Using (207) the system (200) is reformulated to $\mathbf{R}\boldsymbol{\alpha} = \mathbf{Q}^T\mathbf{z}$, which can be easily solved by back-substitution. A similar procedure can be used when solving (204) for $\partial\boldsymbol{\alpha}/\partial\beta_j$, with \mathbf{R} being saved after the solution of (200).

The numerical computation of the covariance matrix $\boldsymbol{\Sigma}(\mathbf{b})$ in (191) can also be done by QR technique. In a general context we may ignore the scaling and constraint matrix factors and just assume that the covariance matrix is obtained by inverting the matrix $\mathbf{P}^T\mathbf{P}$ where \mathbf{P} is some $n \times p$ matrix. As in (207) we make a QR decomposition

$$\mathbf{P} = \mathbf{QR} \quad (208)$$

which leads to

$$(\mathbf{P}^T\mathbf{P})^{-1} = \mathbf{R}^{-1}(\mathbf{R}^{-1})^T \quad (209)$$

In some ill-conditioned problems the diagonal row of \mathbf{R} may contain very small elements, which would render the evaluation of (209) 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 [55,62]. Following this idea, we shall replace (208) with

$$\mathbf{P}\boldsymbol{\Lambda}\boldsymbol{\Pi} = \mathbf{QR} \quad (210)$$

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{P} and might be called ‘‘uncoupled standard deviations’’. Instead of (209) we obtain

$$(\mathbf{P}^T\mathbf{P})^{-1} = \boldsymbol{\Lambda}\boldsymbol{\Pi}\mathbf{R}^{-1}(\boldsymbol{\Lambda}\boldsymbol{\Pi}\mathbf{R}^{-1})^T \quad (211)$$

which is verified by solving (210) for \mathbf{P} , then calculating \mathbf{P}^T and $\mathbf{P}^T\mathbf{P}$, and finally $(\mathbf{P}^T\mathbf{P})^{-1}$. The expression (211) 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 abandoned.

6 Appendix B: Model details

In Sections 2.2 and 2.5 we gave a short presentation of the theoretical models used in POSITRONFIT and RESOLUTIONFIT. Below we shall fill the gap between the rather brief description given there of the underlying mathematical models, and the least-squares theory in Appendix A.

6.1 POSITRONFIT

Writing formula (3) as

$$f(t) = \sum_{j=1}^{k_0} \sum_{p=1}^{k_g} \omega_p (a_j * G_p)(t) + B \quad (212)$$

we must evaluate the convolution integral

$$(a_j * G_p)(t) = \int_{-\infty}^{\infty} a_j(v)G_p(t-v)dv \quad (213)$$

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

$$\lambda_j = 1/\tau_j \quad (214)$$

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

$$(a_j * G_p)(t) = \frac{1}{2} A_j \phi(t - T_0 - \Delta_p, \lambda_j, s_p) \quad (215)$$

Here

$$\phi(u, \lambda, s) = \exp\left(-\lambda u + \frac{1}{2} \lambda^2 s^2\right) \operatorname{erfc}\left(\frac{\lambda s^2 - u}{\sqrt{2}s}\right) \quad (216)$$

where erfc stands for the complementary error function

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) \quad (217)$$

and erf in turn is defined by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt \quad (218)$$

Inserting (215) in (212) we get

$$f(t) = \frac{1}{2} \sum_{j=1}^{k_0} A_j \sum_{p=1}^{k_g} \omega_p \phi(t - T_0 - \Delta_p, \lambda_j, s_p) + B \quad (219)$$

Next, we compute the integrated model-predicted count f_i defined by equation (12) in Section 2.2. We use that, up to a constant,

$$\int \phi(u, \lambda, s) du = -\frac{1}{\lambda} \psi(u, \lambda, s) \quad (220)$$

where

$$\psi(u, \lambda, s) = \phi(u, \lambda, s) + \operatorname{erfc}\left(\frac{u}{\sqrt{2}s}\right) \quad (221)$$

The functions ϕ and ψ are building blocks in the POSITRONFIT model. For the predicted channel counts we obtain

$$f_i = \sum_{j=1}^{k_0} F_{ij} + B = \sum_{j=1}^{k_0} \alpha_j f_{ij} + B \quad (222)$$

where

$$\alpha_j = \frac{1}{2} A_j / \lambda_j = \frac{1}{2} A_j \tau_j \quad (223)$$

is half the absolute intensity of lifetime component j ,

$$f_{ij} = \sum_{p=1}^{k_g} \omega_p \{\psi(t_{i-1,p}, \lambda_j, s_p) - \psi(t_{ip}, \lambda_j, s_p)\} \quad (224)$$

and where we use the shorthand notation

$$t_{ip} = i_{\text{ch}}^{\min} + i - 1 - T_0 - \Delta_p \quad (225)$$

By now we have arrived at the model expression $f_i = f_i(\mathbf{b})$ entering the least-squares formulation of the fitting problem given in Appendix A. We also see that (222) is separable as required; the parameter vector \mathbf{b} splits into a “linear” parameter $\boldsymbol{\alpha}$ and a “nonlinear” one $\boldsymbol{\beta}$ given by

$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{k_0}, B) \quad (226)$$

and

$$\boldsymbol{\beta} = (\lambda_1, \dots, \lambda_{k_0}, T_0) \quad (227)$$

Thus the separable fitting theory of Section 5.5 applies. To perform the computations outlined there, we must evaluate the derivatives of f_{ij} in (224) with respect to λ_j and T_0 ; this job is facilitated by the following two formulas:

$$\frac{\partial \psi}{\partial u} = -\lambda \phi(u, \lambda, s) \quad (228)$$

and

$$\frac{\partial \psi}{\partial \lambda} = (\lambda s^2 - u)\phi(u, \lambda, s) - \sqrt{\frac{2}{\pi}} s \exp\left(-\frac{u^2}{2s^2}\right) \quad (229)$$

(228) shows that ψ is a decreasing function of u . From (224–225) and (228–229) we obtain

$$\frac{\partial f_{ij}}{\partial T_0} = \sum_{p=1}^{k_g} \omega_p f_{ijp}^T \quad (230)$$

$$\frac{\partial f_{ij}}{\partial \lambda_j} = \sum_{p=1}^{k_g} \omega_p f_{ijp}^\lambda \quad (231)$$

where

$$f_{ijp}^T = -\lambda_j \delta \phi_{ijp} \quad (232)$$

$$f_{ijp}^\lambda = \phi_{i-1,jp} - (\lambda_j s_p^2 - t_{ip}) \delta \phi_{ijp} + s_p \sqrt{2/\pi} \delta_{ip}^{\text{exp}} \quad (233)$$

Moreover

$$\phi_{ijp} = \phi(t_{ip}, \lambda_j, s_p) \quad (234)$$

$$\delta \phi_{ijp} = \phi_{ijp} - \phi_{i-1,jp} \quad (235)$$

$$\delta_{ip}^{\text{exp}} = \exp\left(-\frac{t_{ip}^2}{2s_p^2}\right) - \exp\left(-\frac{t_{i-1,p}^2}{2s_p^2}\right) \quad (236)$$

Considering now the practical computation of ϕ by (216), we reparametrize the arguments by the substitution

$$(x, y) = \left(\frac{\lambda s^2 - u}{\sqrt{2}s}, \frac{u}{\sqrt{2}s}\right) \quad (237)$$

This gives

$$\phi = \exp(x^2 - y^2) \operatorname{erfc}(x) \quad (238)$$

where

$$x + y > 0 \quad (239)$$

If $x \leq 0$ it is safe to use (238) since in that case $x^2 - y^2 < 0$ due to (239). But when $x \gg 0$ a numerical problem may occur. Let us express $\operatorname{erfc}(x)$ in terms of a confluent hypergeometric function. Using the Whittaker notation [63] we get

$$\operatorname{erfc}(x) = \frac{1}{\sqrt{\pi}} x^{-\frac{1}{2}} \exp(-\frac{1}{2}x^2) W_{-\frac{1}{4}, \frac{1}{4}}(x^2) \quad (240)$$

which has the asymptotic expression

$$\operatorname{erfc}(x) \approx \frac{1}{\sqrt{\pi x}} (1 + \mathcal{O}(x^{-2})) e^{-x^2}, \quad x \rightarrow +\infty \quad (241)$$

Now suppose that also $x^2 - y^2 \gg 0$. Then (241) shows that ϕ itself is small; nevertheless, the first factor of (238) is large and may cause an overflow in the computer. At the same time, the second factor is very small and likely to underflow. A remedy is to replace (238) by

$$\phi = \exp(-y^2) \operatorname{erfcx}(x) \quad (242)$$

when $x > 0$, where erfcx stands for the *scaled complementary error function*

$$\operatorname{erfcx}(x) = \exp(x^2) \operatorname{erfc}(x) \quad (243)$$

It is not hard to develop robust and accurate numerical approximations for this slowly varying function. Since

$$\operatorname{erfcx}(x) = \frac{1}{\sqrt{\pi}} x^{-\frac{1}{2}} \exp(\frac{1}{2}x^2) W_{-\frac{1}{4}, \frac{1}{4}}(x^2) \quad (244)$$

we have the integral representation [63],

$$\operatorname{erfcx}(x) = \frac{1}{\sqrt{\pi}} \int_0^\infty (x^2 + t)^{-\frac{1}{2}} e^{-t} dt \quad (245)$$

which decreases steadily for increasing $x > 0$. The function $\text{erfcx}(x)$ behaves asymptotically as $1/(\sqrt{\pi}x)$ owing to (241).

We may include the special case $s_p = 0$ (no instrumental smearing) by letting $s \rightarrow 0$ in (216), (221), (228), (229). Using the limit formulas

$$\lim_{s \rightarrow 0} \exp\left(-\frac{u^2}{2s^2}\right) = 0 \quad (246)$$

and

$$\lim_{s \rightarrow 0} \text{erfc}\left(\frac{u}{\sqrt{2}s}\right) = \begin{cases} 2 & u < 0 \\ 0 & u > 0 \end{cases} \quad (247)$$

where (246) holds good for almost all u , we may collect the necessary limit functions in the following scheme:

Function	$u < 0$	$u > 0$	
ϕ	0	$2e^{-\lambda u}$	
ψ	2	$2e^{-\lambda u}$	
$\partial\psi/\partial u$	0	$-2\lambda e^{-\lambda u}$	
$\partial\psi/\partial\lambda$	0	$-2ue^{-\lambda u}$	(248)

We also want to be able to handle the special case of a fixed zero lifetime $\tau = 0$, i.e. $\lambda = 1/\tau = \infty$. This requires that we compute the limiting forms of (216) and (228) as $\lambda \rightarrow \infty$. We find:

$$\lim_{\lambda \rightarrow +\infty} \phi(u, \lambda, s) = 0 \quad (249)$$

$$\lim_{\lambda \rightarrow +\infty} \left[\frac{\partial\psi}{\partial u}(u, \lambda, s) = -\lambda\phi(u, \lambda, s) \right] = -\frac{1}{s} \sqrt{\frac{2}{\pi}} \exp\left(-\frac{u^2}{2s^2}\right) \quad (250)$$

which implies that (230) becomes

$$\lim_{\lambda \rightarrow +\infty} \frac{\partial f_{ij}}{\partial T_0} = -\sqrt{\frac{2}{\pi}} \sum_{p=1}^{k_g} \frac{\omega_p}{s_p} \delta_{ip}^{\text{exp}} \quad (251)$$

In Section 2.2 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 (226) and (227) are realized by deleting the corresponding components from α or β . 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

$$\sum h_{ij} \alpha_j = \gamma_i \quad (252)$$

where h_{ij} are known coefficients, cf. (59). The same holds good for constraints of the type “a linear combination of the relative intensities = 0”, as well as the total-area constraint (19).

6.2 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 standard deviations (widths) s_p and the shifts Δt_p are fitting parameters in RESOLUTIONFIT. Hence (227) should be replaced by

$$\beta = (\lambda_1, \dots, \lambda_{k_0}, T_0, s_1, \dots, s_{k_g}, \Delta_1, \dots, \Delta_{k_g}) \quad (253)$$

In Section 2.5 we mentioned the types of constraints which could be imposed on the parameters in RESOLUTIONFIT. Some of the parameters of (253) can be fixed and in that case should

be deleted from β . This may apply to λ_j , s_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 (228–236) we shall need the following formulas:

$$\frac{\partial \psi}{\partial s} = \lambda^2 s \phi(u, \lambda, s) - \sqrt{\frac{2}{\pi}} \lambda \exp\left(-\frac{u^2}{2s^2}\right) \quad (254)$$

$$\frac{\partial f_{ij}}{\partial \Delta_p} = -\lambda_j \omega_p \delta \phi_{ijp} \quad (255)$$

$$\frac{\partial f_{ij}}{\partial s_p} = \omega_p \{-\lambda_j^2 s_p \delta \phi_{ijp} + \lambda_j \sqrt{2/\pi} \delta_{ip}^{\text{exp}}\} \quad (256)$$

RESOLUTIONFIT also requires some extra limit formulas for $\lambda = \infty$. Applying (250) to (255) we obtain

$$\lim_{\lambda \rightarrow +\infty} \frac{\partial f_{ij}}{\partial \Delta_p} = -\sqrt{\frac{2}{\pi}} \frac{\omega_p}{s_p} \delta_{ip}^{\text{exp}} \quad (257)$$

The limit of (254) is found to

$$\lim_{\lambda \rightarrow +\infty} \left[\frac{\partial \psi}{\partial s}(u, \lambda, s) = \lambda^2 s \phi(u, \lambda, s) - \sqrt{\frac{2}{\pi}} \lambda \exp\left(-\frac{u^2}{2s^2}\right) \right] = \frac{u}{s^2} \sqrt{\frac{2}{\pi}} \exp\left(-\frac{u^2}{2s^2}\right) \quad (258)$$

and applying this to (256) we get

$$\lim_{\lambda \rightarrow +\infty} \frac{\partial f_{ij}}{\partial s_p} = \sqrt{\frac{2}{\pi}} \frac{\omega_p}{s_p^2} \left\{ t_{i-1,p} \exp\left(-\frac{t_{i-1,p}^2}{2s_p^2}\right) - t_{ip} \exp\left(-\frac{t_{ip}^2}{2s_p^2}\right) \right\} \quad (259)$$

The limit formulas (249), (250), (258) can be verified by using (237–238) and (241).

In RESOLUTIONFIT we compute shape parameters for the fitted resolution curve. To explain our method, we consider a general composite Gaussian resolution function,

$$f(t) = \sum_{j=1}^k \alpha_j G(t, \sigma_j, \Delta_j) \quad (260)$$

where $k \in \mathbb{N}$, $\alpha_j > 0$, $t \in \mathbb{R}$, $\sigma_j > 0$, $\Delta_j \in \mathbb{R}$, and $\sum_{j=1}^k \alpha_j = 1$. Moreover,

$$G(t, \sigma, \Delta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(t-\Delta)^2}{2\sigma^2}\right) \quad (261)$$

is a normalized Gaussian satisfying

$$\int_{-\infty}^{\infty} G(t, \sigma, \Delta) dt = 1 \quad (262)$$

We assume that (260) is a unimodal function. Then the peak position t_p is uniquely determined by solving the equation $f'(t) = 0$ wrt t . Seeking a numerical solution we shall use the Newton-Raphson method,

$$t := t - \frac{f'(t)}{f''(t)} \quad (263)$$

We have

$$f'(t) = \sum_{j=1}^k \alpha_j G'(t, \sigma_j, \Delta_j), \quad f''(t) = \sum_{j=1}^k \alpha_j G''(t, \sigma_j, \Delta_j) \quad (264)$$

where

$$G'(t, \sigma, \Delta) = -\frac{t-\Delta}{\sqrt{2\pi}\sigma^3} \exp\left(-\frac{(t-\Delta)^2}{2\sigma^2}\right), \quad G''(t, \sigma, \Delta) = \frac{(t-\Delta)^2 - \sigma^2}{\sqrt{2\pi}\sigma^5} \exp\left(-\frac{(t-\Delta)^2}{2\sigma^2}\right) \quad (265)$$

An initial guess of t to start the NR procedure may be provided from a pre-tabulation of $f(t)$ in (260), using a suitable fineness of the t -entries. Such a table may also present a numerical verification that (260) is indeed unimodal. We also want to calculate a table of full-width-at-1/ n -max values for a series n_1, n_2, \dots, n_m of n -values. We take $m = 7$ and use the values in the following table:

i	n_i
1	2
2	5
3	10
4	30
5	100
6	300
7	1000

Each of the equations

$$f(t) = \frac{1}{n_i} f(t_p), \quad i = 1, \dots, m \quad (266)$$

will be satisfied by 2 values of t such that we have

$$f(t_p - \tau_i^-) = f(t_p + \tau_i^+) = \frac{1}{n_i} f(t_p), \quad i = 1, \dots, m \quad (267)$$

where τ_i^- and τ_i^+ are both positive. The corresponding full-width and midpoint are

$$\text{FW}_i = \tau_i^+ + \tau_i^-, \quad \mu_i = t_p + \frac{1}{2}(\tau_i^+ - \tau_i^-) \quad (268)$$

It is problematic to use the Newton-Raphson method directly to obtain a numerical solution of (266). The reason is the inflexion points of the resolution curve (260) which may trap the NR-procedure. We may still use NR but we shall work with the logarithmic counterpart to (266),

$$\psi(t) \equiv \ln f(t) - \ln f(t_p) + \ln n_i = 0, \quad i = 1, \dots, m \quad (269)$$

In analogy with (263) the NR rule here becomes

$$t := t - \frac{\psi(t)}{\psi'(t)} \quad \text{where} \quad \psi'(t) = \frac{f'(t)}{f(t)} \quad (270)$$

Again, our pre-tabulation of $f(t)$ will be useful in providing initial t -values for the NR iterations.

7 Appendix C: Log-normal details

7.1 Log-normal POSITRONFIT model formulas

We shall here discuss the *PALSfit3* extension of the POSITRONFIT model to cope with the situation where some of the lifetimes obey log-normal distributions. Such an extension requires a modification of the model description in Section 6.1.

When implementing the log-normal extension in POSITRONFIT we shall still work with annihilation rates rather than lifetimes in the internal model formulation. A stochastic annihilation rate λ is related to the corresponding stochastic lifetime τ by

$$\lambda = \frac{1}{\tau}, \quad \ln \lambda = -\ln \tau \quad (271)$$

cf. (214). If τ has a log-normal distribution defined by (21–25), it follows from (271) that

$$\ln \lambda \in \text{N}(\ln \lambda_*, \sigma_*^2) \quad (272)$$

where

$$\lambda_* = \frac{1}{\tau_*} \quad (273)$$

and hence λ is log-normally distributed with the same σ_* parameter as τ . Then the pdf for λ becomes

$$f(\lambda) = f(\lambda; \lambda_*, \sigma_*) = \frac{1}{\lambda \sigma_* \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_*^2} (\ln \lambda - \ln \lambda_*)^2\right) \quad (274)$$

while the CDF is

$$F(\lambda) = \Phi\left(\frac{\ln \lambda - \ln \lambda_*}{\sigma_*}\right) \quad (275)$$

Extension of the classical POSITRONFIT model (Appendix B, Section 6.1) to the log-normal case requires evaluation of the log-normal probability-weighted integral of (221),

$$I = \int_0^\infty f(\lambda; \lambda_*, \sigma_*) \psi(u, \lambda, s) d\lambda \quad (276)$$

We shall also need the derivatives $\partial I/\partial u$, $\partial I/\partial \lambda_*$, $\partial I/\partial \sigma_*$. From (228) we obtain

$$\frac{\partial I}{\partial u} = - \int_0^\infty f(\lambda; \lambda_*, \sigma_*) \lambda \phi(u, \lambda, s) d\lambda \quad (277)$$

In the following we use the abbreviation

$$\mu(\lambda) = \ln \lambda - \ln \lambda_* = \ln(\lambda/\lambda_*) \quad (278)$$

To find $\partial I/\partial \lambda_*$ we must compute $\partial f(\lambda; \lambda_*, \sigma_*)/\partial \lambda_*$. The result is:

$$\frac{\partial}{\partial \lambda_*} f(\lambda; \lambda_*, \sigma_*) = f(\lambda; \lambda_*, \sigma_*) \frac{\mu(\lambda)}{\lambda_* \sigma_*^2} \quad (279)$$

Thus

$$\frac{\partial I}{\partial \lambda_*} = \int_0^\infty f(\lambda; \lambda_*, \sigma_*) \frac{\mu(\lambda)}{\lambda_* \sigma_*^2} \psi(u, \lambda, s) d\lambda \quad (280)$$

Similarly for $\partial I/\partial \sigma_*$:

$$\frac{\partial}{\partial \sigma_*} f(\lambda; \lambda_*, \sigma_*) = -f(\lambda; \lambda_*, \sigma_*) \frac{\sigma_*^2 - \mu(\lambda)^2}{\sigma_*^3} \quad (281)$$

$$\frac{\partial I}{\partial \sigma_*} = - \int_0^\infty f(\lambda; \lambda_*, \sigma_*) \frac{\sigma_*^2 - \mu(\lambda)^2}{\sigma_*^3} \psi(u, \lambda, s) d\lambda \quad (282)$$

7.2 Fixing one of the log-normal parameters τ_m or σ

In our least squares fitting procedures we want to be able to fix either the mean τ_m of the log-normal distribution or its standard deviation σ , or both. The cases (τ_m fixed, σ free) and (τ_m free, σ fixed) require special attention. Considering first τ_m fixed to τ_{m0} and σ free, this implies that the intrinsic parameters τ_* (or $\lambda_* = 1/\tau_*$) and σ_* cannot vary freely but are bounded by the constraint

$$\frac{1}{\lambda_*} \exp\left(\frac{1}{2}\sigma_*^2\right) = \tau_{m0} \quad (283)$$

Equation (283) defines the function

$$\lambda_*(\sigma_*) = \frac{1}{\tau_{m0}} \exp\left(\frac{1}{2}\sigma_*^2\right) \quad (284)$$

We may eliminate λ_* from the pdf $f(\lambda; \lambda_*, \sigma_*)$ by considering σ_* as the only free parameter. This gives the new pdf

$$f(\lambda; \lambda_*(\sigma_*), \sigma_*) = f_1(\lambda; \sigma_*) \quad (285)$$

If we interpret $f(\lambda; \lambda_*, \sigma_*)$ in (276) and (277) according to (285), these formulas still hold good. To obtain $\partial I/\partial \sigma_*$ we compute $\partial f_1(\lambda; \sigma_*)/\partial \sigma_*$. We have

$$\frac{\partial f_1(\lambda; \sigma_*)}{\partial \sigma_*} = \frac{\partial f(\lambda; \lambda_*, \sigma_*)}{\partial \lambda_*} \frac{d\lambda_*}{d\sigma_*} + \frac{\partial f(\lambda; \lambda_*, \sigma_*)}{\partial \sigma_*} \quad (286)$$

where, by (284),

$$\frac{d\lambda_*}{d\sigma_*} = \lambda_* \sigma_* \quad (287)$$

By (279) and (281) we find that (282) should be replaced by

$$\frac{\partial I}{\partial \sigma_*} = \int_0^\infty f(\lambda; \lambda_*, \sigma_*) \frac{\mu(\lambda)\sigma_*^2 - \sigma_*^2 + \mu(\lambda)^2}{\sigma_*^3} \psi(u, \lambda, s) d\lambda \quad (288)$$

where $\mu(\lambda)$ was defined in (278). When the iteration on σ_* has converged in the least squares iteration process, we may improve the fit by a final recalculation of λ_* by (284) before using (28) to evaluate σ . We proceed similarly when keeping τ_m free and fixing σ to σ_0 . Here we have the constraint

$$\frac{1}{\lambda_*} \exp\left(\frac{1}{2}\sigma_*^2\right) \sqrt{\exp(\sigma_*^2) - 1} = \sigma_0 \quad (289)$$

which defines the function

$$\sigma_*(\lambda_*) = \sqrt{\ln\left(\frac{1}{2}\left(1 + \sqrt{1 + 4\lambda_*^2\sigma_0^2}\right)\right)} \quad (290)$$

We obtain

$$f(\lambda; \lambda_*, \sigma_*(\lambda_*)) = f_2(\lambda; \lambda_*) \quad (291)$$

$$\frac{\partial f_2(\lambda; \lambda_*)}{\partial \lambda_*} = \frac{\partial f(\lambda; \lambda_*, \sigma_*)}{\partial \lambda_*} + \frac{\partial f(\lambda; \lambda_*, \sigma_*)}{\partial \sigma_*} \frac{d\sigma_*}{d\lambda_*} \quad (292)$$

and then, with the abbreviation

$$\kappa = 1 - \exp(-\sigma_*^2) \quad (293)$$

we get

$$\frac{d\sigma_*}{d\lambda_*} = \frac{\kappa}{1 + \kappa} \frac{1}{\lambda_* \sigma_*} \quad (294)$$

$$\frac{\partial I}{\partial \lambda_*} = - \int_0^\infty f(\lambda; \lambda_*, \sigma_*) \frac{\kappa \sigma_*^2 - (1 + \kappa) \sigma_*^2 \mu(\lambda) - \kappa \mu(\lambda)^2}{(1 + \kappa) \lambda_* \sigma_*^4} \psi(u, \lambda, s) d\lambda \quad (295)$$

In this case we should use (290) to make a final recalculation of σ_* before using (26) to evaluate τ_m .

7.3 Jacobian matrix for output presentation

We shall also need the Jacobian matrix

$$\frac{\partial(\tau_m, \sigma)}{\partial(\tau_*, \sigma_*)} = \begin{pmatrix} \frac{\partial \tau_m}{\partial \tau_*} & \frac{\partial \tau_m}{\partial \sigma_*} \\ \frac{\partial \sigma}{\partial \tau_*} & \frac{\partial \sigma}{\partial \sigma_*} \end{pmatrix} \quad (296)$$

which is used for covariance estimation for the output presentation. From (26) and (28) we obtain

$$\frac{\partial \tau_m}{\partial \tau_*} = \exp\left(\frac{1}{2}\sigma_*^2\right) \quad (297)$$

$$\frac{\partial \tau_m}{\partial \sigma_*} = \tau_* \sigma_* \exp\left(\frac{1}{2}\sigma_*^2\right) \quad (298)$$

$$\frac{\partial \sigma}{\partial \tau_*} = \exp\left(\frac{1}{2}\sigma_*^2\right) \sqrt{\exp(\sigma_*^2) - 1} \quad (299)$$

$$\frac{\partial \sigma}{\partial \sigma_*} = \frac{\tau_* \sigma_* \exp\left(\frac{1}{2}\sigma_*^2\right) (2 \exp(\sigma_*^2) - 1)}{\sqrt{\exp(\sigma_*^2) - 1}} \quad (300)$$

From (300) we derive

$$\lim_{\sigma_* \rightarrow 0} \frac{\partial \sigma}{\partial \sigma_*} = \tau_* \quad (301)$$

In the special cases where either τ_m or σ is fixed, (297–300) hold good for the free parameter.

7.4 Numerical evaluation of log-normal integrals

All the integrals (276), (277), (280), (282), (288), (295) must be evaluated numerically. In a broader context, let us consider the evaluation of integrals of the form

$$J = \int_0^{\infty} f(x)g(x)dx \quad (302)$$

where f is a general pdf satisfying

$$\int_0^{\infty} f(x)dx = 1, \quad f(x) > 0 \text{ in } (0, \infty) \quad (303)$$

and g is an arbitrary function. The integral (302) may be thought of as a probability-weighted mean of g , and so we shall be inspired by the Monte Carlo method when devising a numerical integration scheme for it. We write

$$J = \int_{x=0}^{\infty} g(x)dF(x) \quad (304)$$

where F is the CDF corresponding to f . Then we substitute

$$F(x) = \xi, \quad x = F^{-1}(\xi), \quad \xi \in (0, 1) \quad (305)$$

(cf. description of the PALGEN program in Appendix D, Section 8.1) and obtain

$$J = \int_0^1 g(F^{-1}(\xi))d\xi \quad (306)$$

We may now apply a quadrature rule on (306) i.e. some approximation formula of the type

$$J \approx \sum_{\nu=1}^N w_{\nu} g(F^{-1}(\xi_{\nu})) \quad (307)$$

where N is the number of quadrature nodes, w_{ν} are positive weights, and ξ_{ν} the node abscissas. Moreover

$$\sum_{\nu=1}^N w_{\nu} = 1 \quad (308)$$

and

$$0 < \xi_1 < \dots < \xi_N < 1 \quad (309)$$

We could use the simple rectangular rule

$$w_{\nu} = \frac{1}{N}, \quad \xi_{\nu} = \frac{2\nu - 1}{2N} \quad (310)$$

but prefer the more efficient Gauss-Legendre (GL) method [61]. Returning to the specific pdf (274), we see from (306–307) that we shall need the inverse function F^{-1} of the function F in (275). This means that we must solve the equation

$$F(x) = \Phi\left(\frac{\ln x - \ln \lambda_*}{\sigma_*}\right) = \xi \quad (311)$$

for x . The result is

$$x = F^{-1}(\xi) = \lambda_* \exp(\sigma_* \Phi^{-1}(\xi)) \quad (312)$$

Thus, finding the inverse of F is reduced to computing the inverse Φ^{-1} of the CDF Φ for $N(0,1)$. Standard software is available for this purpose. We can then write

$$J = \int_0^1 g(\lambda_* \exp(\sigma_* \Phi^{-1}(\xi)))d\xi \quad (313)$$

The corresponding numerical approximation is

$$J \approx \sum_{\nu=1}^N w_{\nu} g(\lambda_* \exp(\sigma_* \Phi^{-1}(\xi_{\nu}))) \quad (314)$$

It is this formula that is used in the numerical approximation of all the log-normal integrals in Sections 7.1 and 7.2. The number of quadrature nodes N is called the *log-normal fineness* of our approximation. In `PALSfit3` the fineness defaults to $N = 32$ which provides a reasonable compromise between speed and accuracy, but other values might be entered by the user, either automatically through `PALSfit3` or by external means. Indeed, this could be done by inserting a value of N in the output option record for `POSITRONFIT`, cf. Section 3.2 Block 1. The number may be placed anywhere in position 5–80; omission or zero implies the default value.

7.5 Discarding log-normal lifetime components

During the least squares fitting procedure it may sometimes happen that σ_* (or, equivalently, σ) tends to zero as the iterations proceed. In such a case `PALSfit3` automatically discards the broadened component and replaces it by a “classic” component with a simple decaying exponential and then resumes the iterations from there. In other words `PALSfit3` may detect situations where log-normal broadenings do not contribute to the fitting of the observations.

8 Appendix D: Quality check

8.1 Simulation of lifetime spectra

In 1997 Hirade² 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 `POSITRONFIT` to recover correct lifetime values. Cheung *et al.* [64], 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 DTU Risø Campus we have later created a FORTRAN-based version of `PALGEN`.

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

$$t = T_0 + t_{\text{anni}} + t_{\text{gauss}} + t_{\text{shift}} \quad (315)$$

Here T_0 is the given time-zero, t_{anni} the true annihilation time, t_{gauss} the instrumental smear component, and t_{shift} a deterministic shift value associated with the latter. We begin with the sampling of t_{anni} .

First a random number determines the actual lifetime component from the given intensities. Then, assuming first a fixed lifetime τ , we have the exponential probability density function (pdf) for $x = t_{\text{anni}}$,

$$f(x) = \frac{1}{\tau} \exp\left(-\frac{x}{\tau}\right) \quad (316)$$

This corresponds to the cumulative distribution function (CDF)

$$F(x) = 1 - \exp\left(-\frac{x}{\tau}\right) \quad (317)$$

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

$$x = F^{-1}(\xi) \quad (318)$$

which is the inverse form of

$$F(x) = \xi \quad (319)$$

²Personal communication, Tetsuya Hirade, then at Department of Materials Science, Japan Atomic Energy Research Institute.

where $\xi \in (0, 1)$ is a uniform random number. By replacing ξ with $1 - \xi$ we then obtain the sampling formula

$$t_{\text{anni}} = -\tau \ln \xi \quad (320)$$

Next we perform the sampling of t_{gauss} . Assuming a composite Gaussian distribution, we first 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 [65] to generate a standardized normal variate η with mean 0 and variance 1, and obtain

$$t_{\text{gauss}} = \eta \sigma \quad (321)$$

Finally, the selected Gaussian component determines t_{shift} . We have now obtained a new sample value of t as given by (315). Using a binning procedure this is converted to a count in a certain channel of the simulation spectrum to be generated. The sampling of the background proceeds independently of the remaining spectrum and is accomplished by simple uniform multinomial sampling with bins = channels.

In Section 2.3 we discussed a log-normal extension of the POSITRONFIT model, in which the lifetime was assumed to be a stochastic variable τ distributed with the pdf given in (25). To cover this situation, only a minor addition to PALGEN was needed. In order to sample τ we must know τ_* and σ_* . Then we can use (21) to generate $\ln \tau$ by the Box-Muller method. Having sampled τ we continue as before from (316) and onwards.

8.2 Verifying POSITRONFIT by statistical analysis

We have developed an ad-hoc tool POSCHECK which is a batch of software components intended to run consecutively under Windows, cf. the following `bat` file:

```
rem Check POSITRONFIT by using simulated spectra
PALGEN
rem PALGEN reads generate.inp, produces spectrum.dat and generate.log
FILECOPY
rem FILECOPY reads template.pfc and generate.inp, produces generate.pfc
rem template.pfc describes how the POSITRONFIT analysis should be done
POSITRONFIT generate.pfc positron.out
rem run POSITRONFIT with input generate.pfc and output positron.out
PANALYSE
rem PANALYSE makes statistical analysis of POSITRONFIT output
rem PANALYSE reads generate.inp and positron.out, produces psummary.txt
if errorlevel 1 goto quit
:quit
```

We restrict our scope to verification of POSITRONFIT without source correction. POSCHECK has 3 main components:

- Simulation program PALGEN as described in Section 8.1
- POSITRONFIT
- Statistical analyser PANALYSE

From given input data, PALGEN produces a random set of say n individual counting spectra, collected in a single spectrum file. Each spectrum is then analysed by POSITRONFIT. The result is a (multidimensional) sample of n individual outputs containing parameter estimates and estimated standard deviations. After this, the auxiliary program PANALYSE collects all these sample estimates and makes a simple statistical overview (“tally”) analysis. In the POSITRONFIT input we use off-central parameter guesses. We give here an illustrative example of applying POSCHECK where we use the following input data:

- Total number of channels = 512, fit range = [35,512]
- Background = 680, total area = $9 \cdot 10^6$ (without Bg); free lifetimes, Bg, and total area
- $C = 0.0773$ ns/ch, cf. (11) in Section 2.2
- Instrumental resolution function with 1 gaussian, FWHM = 0.42 ns
- 2 lifetime components with $\tau = 0.30$ ns and $\tau = 2.00$ ns, respectively
- No log-normal smearing of lifetime components
- Lifetime intensities: 60% and 40%, respectively
- $T_0 = 136$
- Sample size $n = 100$

We show below (part of) the output produced by PANALYSE:

ANALYSIS REPORT OBTAINED FROM 100 SPECTRA
INITIAL RANDOM NUMBER: 54711441

TALLY STATISTICS FOR REDUCED CHI-SQUARE

SAMPLE MEAN (SM): 1.014088
SAMPLE DEVI: 0.063280
MEAN PREDICTED DEVI 0.065094
SEM: 0.006328
U=(SM-TARGET)/SEM: 2.226363
MEAN SIGNIFICANCE % 57.246279

TALLY STATISTICS FOR LIFETIMES

TARGET:	0.300000	2.000000
SAMPLE MEAN (SM):	0.299974	1.999719
SAMPLE DEVI:	0.000448	0.001898
MEAN PREDICTED DEVI	0.000400	0.002000
SEM:	0.000045	0.000190
U=(SM-TARGET)/SEM:	-0.579734	-1.480639

TALLY STATISTICS FOR INTENSITIES

TARGET:	60.000000	40.000000
SAMPLE MEAN (SM):	59.999830	40.000170
SAMPLE DEVI:	0.046572	0.046572
MEAN PREDICTED DEVI	0.042172	0.042172
SEM:	0.004657	0.004657
U=(SM-TARGET)/SEM:	-0.036503	0.036503

TALLY STATISTICS FOR BACKGROUND

TARGET: 680.000000
SAMPLE MEAN (SM): 680.081674
SAMPLE DEVI: 1.002812
MEAN PREDICTED DEVI 1.500642
SEM: 0.100281
U=(SM-TARGET)/SEM: 0.814450

TALLY STATISTICS FOR T0

TARGET: 136.000000
SAMPLE MEAN (SM): 136.000038
SAMPLE DEVI: 0.002181
MEAN PREDICTED DEVI 0.002300
SEM: 0.000218
U=(SM-TARGET)/SEM: 0.174251

CORRELATION MATRIX AVERAGED OVER 100 PREDICTIONS

	INTEN	INTEN	BACKG	LIFET	LIFET	TZERO
INTEN	1.000	-1.000	-0.112	0.666	0.706	-0.249
INTEN	-1.000	1.000	0.112	-0.666	-0.706	0.249
BACKG	-0.112	0.112	1.000	-0.107	-0.273	0.056

LIFET	0.666	-0.666	-0.107	1.000	0.540	-0.682
LIFET	0.706	-0.706	-0.273	0.540	1.000	-0.233
TZERO	-0.249	0.249	0.056	-0.682	-0.233	1.000

PEARSON CORRELATION MATRIX BY 100 PARAMETER ESTIMATES

	INTEN	INTEN	BACKG	LIFET	LIFET	TZERO
INTEN	1.000	-1.000	-0.052	0.742	0.686	-0.273
INTEN	-1.000	1.000	0.052	-0.742	-0.686	0.273
BACKG	-0.052	0.052	1.000	-0.123	-0.322	0.100
LIFET	0.742	-0.742	-0.123	1.000	0.575	-0.688
LIFET	0.686	-0.686	-0.322	0.575	1.000	-0.266
TZERO	-0.273	0.273	0.100	-0.688	-0.266	1.000

To explain the meaning of the tally statistics, note that the TARGET values are input values to PALGEN (except for the reduced χ^2 where TARGET = 1). The SAMPLE MEAN (SM) and SAMPLE DEVI are for any parameter x given by the usual formulas for sample mean and sample standard deviation:

$$\text{SM} = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (322)$$

$$\text{SAMPLE DEVI} = s = \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 / (n - 1)} \quad (323)$$

The MEAN PREDICTED DEVI is the average over the n predictions of the standard deviation made by POSITRONFIT. SEM is the Standard Error of the Mean:

$$\text{SEM} = (\text{SAMPLE DEVI}) / \sqrt{n} \quad (324)$$

The quantity (SM-TARGET)/SEM is called the u -value, and when n is not too small, we expect this to be distributed approximately in a standardized normal distribution, $u \in N(0, 1)$.

Our results show good agreement between TARGET and SAMPLE MEAN. Moreover, except for the Bg, there is a fair agreement between SAMPLE DEVI and MEAN PREDICTED DEVI. For the u -values we see that $|u|$ is of the order of magnitude 1, which should be expected.

We have used POSCHECK successfully in many other kinds of problems, including some with log-normal lifetime broadenings.

If we increase the sample size n to very large values, there is a tendency to get $|u|$ -values substantially larger than 1. This is because round-off and imperfect numerical algorithms begin to dominate over the statistical errors.

Concerning the correlation matrix we have made another kind of comparison between predicted values and sample values. The first of the two matrices, M_1 is formed by a simple averaging over the n matrices predicted by POSITRONFIT. In the second one, M_2 , the elements are estimated from the sample itself. Indeed, each element is computed as a Pearson Product Moment Correlation (PPMC). The general formula for this correlation coefficient between x and y is:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (325)$$

In our example we see that there is a rough equivalence between M_1 and M_2 , $M_1 \approx M_2$, again apart from the Bg entries. The bad news are that the convergence of $M_2 = M_2(n)$ is extremely slow, as $n \rightarrow \infty$. The good news are that we have the opposite situation for M_1 where the convergence $M_1(n) \rightarrow M_1(\infty)$ is fast. Already $M_1(1)$ (single sample) is a reasonable approximation to $M_1(\infty)$ (infinite sample).

8.3 Comparison of PALS*fit*3 with LT10

The model function we have assumed in PALS*fit*3 to provide a realistic broadening (standard deviation) of each of the decaying exponentials is a log-normal distribution. This is the same as in the LT10 program by Giebel and Kansy [15]. We have therefore carried out a comparison of PALS*fit*3 with LT10 by simulating – with PALGEN – a series of spectra and analysed them with both programs and for both carried out a statistical analysis of the results like the one described in Section 8.2

The input data to the simulation were the following:

- Total number of channels = 2000
- Area without background = $4 \cdot 10^6$
- Background = 800
- Time-zero = 259
- Time per channel = 0.015 ns
-
- *Resolution Function:*
- Number of Gaussians = 2
- FWHM of Gaussians = 0.25, 0.35 (ns)
- Intensities of Gaussians = 80, 20 (%)
- Shifts of Gaussians = 0, 0.075 (ns)
-
- *Lifetime components:*
- Number of lifetimes = 3
- Lifetimes = 0.15, 0.4, 1.8 (ns)
- Log-normal broadenings = 0, 0.1, 0.4 (ns)
- Lifetime intensities = 15, 40, 45 (%)
-
- Number of simulated spectra: 20

The input data to the analysis of the simulated spectra were the following:

- Start- and stop-channel for the analyses: 240-1994 (PALS*fit*), 241-1995 (LT10); the channel number deviates by 1 between the two programs.
- Time-zero, FIXED at 259 (PALS*fit*), 260 (LT10).
- Time per channel = 0.015 ns
-
- *Resolution Function:*
- Number of Gaussians = 2
- FWHM of Gaussians = 0.25, 0.35 (ns)
- Intensities of Gaussians = 80, 20 (%)
- Shifts of Gaussians = 0, 0.075 (ns)
-
- *Lifetime components:*
- Number of lifetimes = 3
- Lifetimes = 0.15, 0.4, 1.8, all GUESSED
- Log-normal broadenings = 0 FIXED, 0.1 FIXED, 0.4 GUESSED

The resolution function parameters were fixed in both programs. The reason for this was that the two programs treat these parameters differently, i.e. in LT10 the resolution function parameters may be fitted, while time-zero (T_0) is a fixed input parameter. In PALS*fit* (POSITRONFIT) on the other hand, the shape of the resolution function (also described by a sum of Gaussians) is

fixed, while T_0 may be fitted. Because of this difference between the two programs, to make the fitting conditions identical, both T_0 , the FWHMs and the shifts of the Gaussians were chosen as fixed parameters in the analyses with both programs.

Results of statistical analyses of output from the two programs are shown below:

Comparison for		PALSfit3			LT10	
Chi-square						
Sample mean	0.9974			0.9961		
Sample std	0.0272			0.0266		
Mean predicted std	0.0338					
Lifetimes (ns)						
Sample mean	0.1491	0.3979	1.7970	0.1482	0.3953	1.7955
Sample std	0.0065	0.0166	0.0156	0.0067	0.0167	0.0156
Mean predicted std	0.0059	0.0137	0.0121	0.0051	0.0284	0.0359
Broadening (ns)						
Sample mean			0.4020			0.4147
Sample std			0.0403			0.0409
Mean predicted std			0.0386			0.0511
Intensities (%)						
Sample mean	14.7234	40.1540	45.1226	14.4494	40.2505	45.3001
Sample std	1.8242	1.0487	0.8729	1.8603	1.0638	0.8881
Mean predicted std	1.5697	1.0923	0.6467	0.0000	9.4876	12.5859
Background						
Sample mean	800.2811			799.8365		
Sample std	0.9523			1.0921		
Mean predicted std	1.0414			0.7775		

Statistics for 20 PALGEN-generated spectra, analysed with PALSfit3 and LT10. “Sample mean” is the average of the results from the 20 spectra used in the test. “Sample std” is the sample standard deviation which measures the scatter of the 20 results. “Mean predicted std” is the average of the standard deviations for the fitted parameters as predicted by the programs.

In general, there is good agreement between the “Sample mean” values obtained by the two programs for all the fitted parameters as well as for the Chi-square. Similarly, the “Sample standard deviation” also shows good agreement between the two programs, i.e. the scatter of the results are similar.

However, when it comes to the ‘Standard deviations’ of the fitted parameters, as estimated by the programs, there are quite large differences. PALSfit3 seems to predict standard deviations of the fitted parameters fairly well (“Mean predicted std” agrees fairly well with “Sample std” for all parameters) while LT10 produces larger deviations for many of the parameters (in particular for the intensities, the predicted standard deviations of which show large scatter).

References

- [1] S. J. Tao, “Methods of data reduction in analyzing positron annihilation lifetime spectra,” *IEEE Transactions on Nuclear Science*, vol. NS15, pp. 175–187, 1968.
- [2] P. C. Lichtenberger, J. R. Stevens, and T. D. Newton, “Analysis of counting distributions with a complex character,” *Canadian Journal of Physics*, vol. 50, pp. 345–351, 1972.
- [3] P. Kirkegaard and M. Eldrup, “POSITRONFIT: A versatile program for analysing positron lifetime spectra,” *Computer Physics Communications*, vol. 3, pp. 240–255, 1972.
- [4] P. Kirkegaard and M. Eldrup, “POSITRONFIT EXTENDED: A new version of a program for analysing positron lifetime spectra,” *Computer Physics Communications*, vol. 7, pp. 401–409, 1974.
- [5] J. K. Stevenson, “Graphical method for the analysis of multicomponent exponential decay data,” *International Journal of Applied Radiation and Isotopes*, vol. 28, pp. 900–918, 1977.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] G. H. Dai, J. Fu, and Q. S. Liu, “A program for the interactive analysis of positron lifetime spectra on personal computers with the aid of screen graphics,” *Applied Physics A*, vol. 53, pp. 303–309, 1991.
- [11] J. Kansy, “Microcomputer program for analysis of positron annihilation lifetime spectra,” *Nuclear Instruments and Methods in Physics Research A*, vol. 374, pp. 235–244, 1996.
- [12] 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.
- [13] A. Karbowski, J. J. Fisz, G. P. Karwasz, J. Kansy, and R. S. Brusa, “Genetic algorithms for positron lifetime data,” *ACTA PHYSICA POLONICA A*, vol. 113, pp. 1365–1372, 2008.
- [14] C. Pascual-Izarra, A. W. Dong, S. J. Pas, A. J. Hill, B. J. Boyd, and C. J. Drummond, “Advanced fitting algorithms for analyzing positron annihilation lifetime spectra,” *Nuclear Instruments and Methods in Physics Research A*, vol. 603, pp. 456–466, 2009.
- [15] D. Giebel and J. Kansy, “A new version of LT program for positron lifetime spectra analysis,” *Materials Science Forum*, vol. 666, pp. 138–141, 2011.
- [16] D. M. Schrader and S. G. Usmar, “Nonexponential decay: The problems of deconvolution and lifetime extraction,” in *Positron annihilation studies of fluids* (S. C. Sharma, ed.), pp. 215–238, World Scientific, 1988.
- [17] Y. Zhu and R. B. Gregory, “Analysis of positron annihilation lifetime data presented as a sum of convoluted exponentials with the program SPLMOD,” *Nuclear Instruments and Methods in Physics Research A*, vol. 284, pp. 443–451, 1989.

- [18] R. B. Gregory and Y. Zhu, "Analysis of positron annihilation lifetime data by numerical Laplace inversion with the program CONTIN," *Nuclear Instruments and Methods in Physics Research A*, vol. 290, pp. 172–182, 1990.
- [19] R. B. Gregory, "Analysis of positron annihilation lifetime data by numerical Laplace inversion: Corrections for source terms and zero-time shift errors," *Nuclear Instruments and Methods in Physics Research A*, vol. 302, pp. 496–507, 1991.
- [20] A. Shukla and M. Peter, "Maximum entropy analysis of positron annihilation lifetime spectra," *Materials Science Forum*, vol. 105-110, pp. 1981–1984, 1992.
- [21] L. Hoffmann, A. Shukla, M. Peter, B. Barbiellini, and A. A. Manuel, "Linear and non-linear approaches to solve the inverse problem: applications to positron annihilation experiments," *Nuclear Instruments and Methods in Physics Research A*, vol. 335, pp. 276–287, 1993.
- [22] A. Shukla, M. Peter, and L. Hoffmann, "Analysis of positron lifetime spectra using quantified maximum entropy and a general linear filter," *Nuclear Instruments and Methods in Physics Research A*, vol. 335, pp. 310–317, 1993.
- [23] A. H. Deng, B. K. Panda, S. Fung, C. D. Beling, and D. M. Schrader, "Positron lifetime analysis using the matrix inverse Laplace transformation method," *Nuclear Instruments and Methods in Physics Research B*, vol. 140, pp. 439–448, 1998.
- [24] 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. Risoe-M; No.2740, Technical University of Denmark, DTU Risoe Campus, DK-4000 Roskilde, Denmark, February 1989. Available as pdf-file from www.palsfit.dk.
- [25] M. Metcalf and J. Reid, *FORTRAN 90/95 explained*. Oxford: Oxford University Press, 1996.
- [26] 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.
- [27] M. Eldrup, "Positron lifetimes in water and ice, and in frozen aqueous solutions," Tech. Rep. Risoe-R; No.254, Technical University of Denmark, DTU Risoe Campus, DK-4000 Roskilde, Denmark, 1971. PhD Thesis; available in pdf from www.palsfit.dk.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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), pp. 645–647, 1988. World Scientific 1989.
- [33] 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), pp. 216–226, 1988. World Scientific 1989.
- [34] 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.

- [35] 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.
- [36] 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.
- [37] 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.
- [38] F. M. Jacobsen, M. Eldrup, and O. E. Mogensen, "The temperature dependence of the two positronium bubble states in liquid SF₆," *Chem. Phys.*, vol. 50, pp. 393–403, 1980.
- [39] 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.
- [40] F. M. Jacobsen, O. E. Mogensen, and N. J. Pedersen, "High resolution positron lifetime measurements in non-polar liquids," in *Proc. 8th Int. Conf. on Positron Annihilation*, (Gent, Belgium), pp. 651–653, 1988. World Scientific 1989.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] 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.
- [46] 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.
- [47] L. Dorikens-Vanpraet, D. Segers, and M. Dorikens, "The influence of geometry on the resolution of a positron annihilation lifetime spectrometer," *Appl. Phys.*, vol. 23, pp. 149–152, 1980.
- [48] 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.
- [49] H. Saito, Y. Nagashima, T. Kuvihara, and T. Hyodo, "A new positron lifetime spectrometer using a fast digital oscilloscope and BaF₂ scintillators," *Nucl. Instr. and Meth. in Phys. Res. A*, vol. 487, pp. 612–617, 2002.
- [50] 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.
- [51] 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.

- [52] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, pp. 431–441, 1963.
- [53] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quart. Appl. Math.*, vol. 2, pp. 164–168, 1944.
- [54] 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.
- [55] J. J. Moré, B. S. Garbow, and K. E. Hillstom, "User guide for MINPACK-1," Tech. Rep. ANL-80-74, Argonne National Laboratory, Applied Mathematics Division, 9700 South Cass Avenue, Argonne, Illinois 60439, 1980.
- [56] P. Kirkegaard, "A FORTRAN IV version of the sum-of-exponential least-squares code EX-POSUM," Tech. Rep. Risoe-M; No.1279, Technical University of Denmark, DTU Risoe Campus, DK-4000 Roskilde, Denmark, September 1970. Available in pdf from www.palsfit.dk.
- [57] P. Kirkegaard, "Some aspects of the general least-squares problem for data fitting," Tech. Rep. Risoe-M; No.1399, Technical University of Denmark, DTU Risoe Campus, DK-4000 Roskilde, Denmark, August 1971. Available in pdf from www.palsfit.dk.
- [58] P. Kirkegaard and M. Eldrup, "The least-squares fitting programme POSITRONFIT: principles and formulas," Tech. Rep. Risoe-M; No.1400, Technical University of Denmark, DTU Risoe Campus, DK-4000 Roskilde, Denmark, September 1971. Available in pdf from www.palsfit.dk.
- [59] L. Kaufman, "A variable projection method for solving separable nonlinear least squares problems," *BIT*, vol. 15, pp. 49–57, 1975.
- [60] 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.
- [61] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 90*. Cambridge: Cambridge University Press, 1996.
- [62] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPACK, User's guide*. Philadelphia: SIAM, 1979.
- [63] E. T. Whittaker and G. N. Watson, *A Course of Modern Analysis*. Cambridge: University Press, 4th ed., 1952.
- [64] 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.
- [65] G. Box and M. Muller, "A note on the generation of random normal deviates," *Annals Math. Statistics*, vol. 29, pp. 610–611, 1958.

The Technical University of Denmark (DTU) is one of Europe's leading technical universities. DTU develops and creates value using the natural sciences and the technical sciences to benefit society. The university has 5800 employees, including 1200 PhD students, and more than 10,000 BEng, BSc and MSc students.

The Department of Energy Conversion and Storage focuses on education, research and development within functional materials and their application in sustainable energy technologies. The Department has 230 employees.

Technical University of Denmark
Department of Energy Conversion and Storage
Risø Campus
Frederiksborgvej 399
DK-Roskilde
Denmark

info@energy.dtu.dk
www.energy.dtu.dk

+45 4677 5800