

RISØ-M-2740

**PATFIT-88: A DATA-PROCESSING SYSTEM FOR
POSITRON ANNIHILATION SPECTRA
ON MAINFRAME AND PERSONAL COMPUTERS**

Peter Kirkegaard, Niels Jørgen Pedersen, and Morten Eldrup.

Abstract. A data processing system has been developed for analyzing positron annihilation lifetime and angular correlation spectra on mainframe and Personal Computers (PCs). The system is based on the PATFIT programs previously developed for use on mainframe computers. It consists of the three fitting programs POSITRONFIT, RESOLUTION and ACARFIT and three associated programs for easy editing of the input data to the fitting programs, as well as a graphics program for the display of measured and fitted spectra. They can be used directly on any IBM-compatible PC. The PATFIT-88 software is available from Risø National Laboratory.

February 1989 — revised August 1997 — reprinted November 2003

Risø National Laboratory, DK-4000 Roskilde, Denmark

ISBN 87-550-1491-7

ISSN 0418-6435

Grafisk Service, Risø, 1989–2003

Contents

1	Introduction	<i>5</i>
2	The Fitting Programs	<i>7</i>
2.1	POSITRONFIT	<i>9</i>
2.2	RESOLUTION	<i>13</i>
2.3	ACARFIT	<i>15</i>
3	The Input Editors	<i>21</i>
3.1	Common Features of the Input Editors	<i>21</i>
3.2	POSEDIT	<i>28</i>
3.3	RESEDIT	<i>36</i>
3.4	ACAREDIT	<i>43</i>
4	Experience with PATFIT	<i>54</i>
4.1	POSITRONFIT	<i>54</i>
4.2	RESOLUTION	<i>55</i>
4.3	ACARFIT	<i>58</i>
5	Installation and Operation of PATFIT-88	<i>61</i>
5.1	General Remarks	<i>61</i>
5.2	PC-versions	<i>63</i>
5.3	Mainframe versions	<i>66</i>
6	PATGRAPH: Program for Graphical Output	<i>67</i>
6.1	Details of PATGRAPH menus	<i>68</i>
6.2	Installation of PATGRAPH	<i>77</i>
	Acknowledgements	<i>80</i>
	Appendices	<i>81</i>

A	Least-Squares Technique with Statistics	<i>81</i>
B	Details of Mathematical Models	<i>93</i>
C	Input data descriptions for the fitting programs	<i>98</i>
D	Subroutines in PATFIT-88	<i>109</i>
	References	<i>116</i>

1 Introduction

An important aspect of doing experiments by conventional positron annihilation techniques (PAT) is carrying out an analysis of the measured spectra to extract physically meaningful parameters. A number of computer programs have been developed over the last 20 years or so by various authors for this purpose. At our laboratory we have concentrated on developing programs for least squares fitting of such spectra on mainframe computers. First, a program for lifetime analysis, POSITRONFIT [1], was developed, followed by an extended version [2] and two programs for analyzing angular correlation curves, PAACFIT [3] and PARAFIT [4]. Together with an additional lifetime program RESOLUTION, these programs were described in detail — under the name PATFIT — by Kirkegaard *et al.* [5]. Other authors have also developed programs, based on POSITRONFIT [6, 7, 8].

The rapid increase in the use of Personal Computers (PCs), also for data collection, has stimulated the need for computer programs which can carry out analyses of PAT data on a PC. The present data-processing system, PATFIT-88, was developed to meet this need (although the programs can also run on mainframe computers). A brief account of the system was published in [9].

PATFIT-88 contains three fitting programs: 1) POSITRONFIT for the extraction of lifetimes and intensities from lifetime spectra. 2) RESOLUTION for determining the time resolution function from measured lifetime spectra. 3) ACARFIT for fitting angular correlation curves. The two lifetime programs are very similar to the previous mainframe versions, while ACARFIT combines the features of the previous two programs PAACFIT and PARAFIT with the addition of some new features.

In order to make the programs more “user-friendly”, three separate interactive programs have been included as a new feature in PATFIT-88. With each of these programs one can easily generate and edit the input data to one of the fitting programs. The programs are called POSEDIT, RESEDIT and ACAREEDIT.

Finally, the package includes a graphics program, PATGRAPH, which makes possible the display of measured and/or fitted spectra on the screen. (This program will not run on a mainframe computer.)

The main purpose of the present report is to give a user-oriented description of the program system. Additional details on the mathematical models used for fitting the data and on the least-squares technique have been included in appendices.

The aim has been to write a comprehensive report in order to avoid too frequent references to previous publications. Therefore, this report at some

places contains some repetitions from previous publications.

In the following we therefore first, in Section 2, give brief descriptions of the model functions used in the three fitting programs along with other characteristics of the programs, in particular the main printout (while Appendices A and B contain additional mathematical details). In Section 3 the functioning of the input editor programs is thoroughly discussed (while Appendix C contains independent prescriptions for direct preparation of the input files to the fitting programs). In Section 4 we briefly discuss the experiences we and others have had with PATFIT-88. In order to further facilitate the use of the program package, Section 5 contains some considerations on the installation and operation of PATFIT-88. Finally, Section 6 gives an account of the features of the graphics program PATGRAPH. PATFIT-88 is available from Risø National Laboratory.

2 The Fitting Programs

The common feature of all the three fitting programs POSITRONFIT, RESOLUTION, and ACARFIT, is that they fit a parametrized model function to a distribution (a “spectrum”) of experimental data points y_i , typically counts recorded in “channels” and obeying Poisson statistics. We use the least-squares criterion, i.e. we seek values of the k model parameters b_1, \dots, b_k that minimizes

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

where n is the number of data points, $f_i(b_1, \dots, b_k)$ the model prediction for point no. i , and w_i a fixed weight attached to this point; in this work we use “statistical weighting”,

$$w_i = 1/\sigma_i^2, \quad (2)$$

where σ_i^2 is the estimated variance of y_i ($= y_i$ itself for unnormalized counts). As some of our parameters enter the models nonlinearly, we must use an iterative fitting technique. In Appendix A we describe the so-called semilinear method applied in all our fitting programs. As a result of the calculations, a number of fitting parameters are estimated which characterize the fitted model function and hence the measured spectrum (e.g. lifetimes and intensities in lifetime spectra and component widths and intensities in angular correlation spectra). 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 intensities is put equal to zero (this latter constraint can be used to fix the ratio of intensities).

As mentioned in the Introduction, the fitting programs are rather similar to the mainframe versions published previously [5]. However, the input and output formats have been changed, the latter to be compatible with the width of an ordinary PC screen and printer. Furthermore, larger flexibility has been introduced for the maximum number of components to be fitted as well as for the number of channels in the spectrum and the spectrum format. This has been obtained as a result of a rationalization of the program structures which also reduced the memory consumption, as required by the PC memory size. Specific information on use and installation of PATFIT-88 on PCs as well as mainframe computers is given in Section 5.

The peripheral data handling has been standardized as much as possible. The normal procedure will be to generate the input data (Control Data) by using the appropriate input editor program (Section 3), but any other editor may also be used to set up Control Data that conform with the input recipes given in Appendix C.

Input

A detailed discussion of the input facilities, including the various types of constraints, is given in connection with the description of the –EDIT programs, Section 3.

Output

The printed output for each of the three programs comprise:

- a) An edited result sheet, called the “Main Output”. 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, “variance of the fit”, “chi-square”, and “significance of imperfect model” are printed. They inform about the agreement between the measured spectrum and the model function (Appendix A.2). 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. The Main Output includes various supplementary results. These are different for the individual programs, and so are the lay-outs of the Main Outputs (see examples below).
- b) An “Input Echo”. This is a raw copy of all the entered input data (optional).
- c) Fitting parameters after each iteration (optional). The printed parameters are internal. After convergence they may need a transformation prior to presentation in the Main Output.
- d) A deviation table and a residual plot (optional). The residuals are displayed in units of their standard deviations.
- e) An estimated Correlation Matrix for the parameters (optional). This matrix and its interpretation is discussed in Appendix A.2.

As indicated above, the printed outputs b)-e) are optional, while the Main Output is always printed (assuming that the output from a program is directed to the printer; see Section 5).

An additional new feature is the optional generation of a so-called Communication File (called POSCOM, RESCOM, and ACARCOM for the three programs, respectively), in addition to the printed output. A Communication File consists of Communication Data Sets, each of which contains the measured spectrum (optional), all the output data given in the printed Main Output and the Correlation Matrix (see below) from a single analysis. The Communication Data Sets are meant to be used as input to further data processing, e.g. trapping model calculations or graphical displays. Each time a new Communication Data Set is generated it will be appended to the Communication File.

The fitting programs may produce many Communication Data Sets in one single run. Should an unforeseen error occur, or you discontinue the execution, the Communication File will not be updated. However, you have not lost those results that were completed prior to the fault: the corre-

sponding Data Sets, as well as the old Data Sets from the Communication File, are saved in an “Error Communication File” with the common name ERRCOM for the three programs. It is formatted exactly as the Communication File, and may be used as such, only you should rename it to, say, POSCOM. (In case of a spectrum reading error, there will be an incomplete Data Set at the end, which you must remove “manually” before you rename.) An already existing ERRCOM file will be overwritten.

In the following three subsections, we briefly describe the characteristics of the individual programs.

2.1 POSITRONFIT

In the POSITRONFIT program (as well as in RESOLUTION), the model function consists in principle of a sum of decaying exponentials convoluted with the resolution function of the lifetime spectrometer, plus a constant background. Let k_0 be 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; 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

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

In (4) τ_j is the mean lifetime of the j th component, and A_j a pre-exponential factor ($A_j \tau_j$ is the area of the component and is printed after each iteration in the optional printout of the iteration process.) We assume, furthermore, that R is given by a sum of k_g Gaussians which may be displaced with respect to each other:

$$R(t) = \sum_{p=1}^{k_g} \omega_p G_p(t), \quad (5)$$

where

$$G_p(t) = \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left(-\frac{(t - T_0 - \Delta t_p)^2}{2\sigma_p^2}\right) \quad (6)$$

and

$$\sum_{p=1}^{k_g} \omega_p = 1. \quad (7)$$

The Gaussian in (6) is centered around channel number $T_0 + \Delta t_p$, where T_0 is a reference channel number (called “time-zero”) and Δt_p a displacement. The standard deviation σ_p of the Gaussian is related to its Full Width at Half Maximum by

$$\text{FWHM}_p = 2(2 \ln 2)^{1/2} \sigma_p. \quad (8)$$

**Model
function**

The curve given by (3) is a continuous curve, while the spectra normally are recorded in channels of a multichannel analyser. For proper comparison, the curve (3) shall therefore be transformed into a histogram by integration over intervals each of which is one channel wide. The model count distribution which is actually fitted to the measured spectrum is therefore given by:

$$f_i = \int_{t_i}^{t_{i+1}} f(t) dt, \quad (9)$$

where t_i is the value of t at the common limit of channel Nos. $i - 1$ and i . Here we assume equidistant channel widths and measure the time in units of channels such that $t_{i+1} - t_i = 1$; in this way the integrated count turns to a channel average.

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

$$f_i = \sum_{j=1}^{k_0} F_{ij} + B, \quad (10)$$

where F_{ij} is the contribution from lifetime component j in spectral channel i . (We relegate the full write-up of F_{ij} to Appendix B.1.)

Fitting parameters

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

$$I_j = A_j \tau_j / \sum_{k=1}^{k_0} A_k \tau_k, \quad (11)$$

Constraints

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.

$$\sum_{j=1}^{k_0} h_{\ell j} I_j = 0. \quad (12)$$

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,

$$\sum_{j=1}^{k_0} A_j \tau_j = \text{constant}. \quad (13)$$

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

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 data are fitted in a first iteration cycle. Then, the spectrum of the source correction is subtracted and the corrected spectrum fitted again in a second iteration cycle. In this second cycle it is optional to choose another number of lifetime components as well as type and number of constraints than are used for the first iteration cycle. The source correction spectrum

f_i^s itself is composed of k_s lifetime components and expressed in analogy with (10) (with $B = 0$) as follows:

$$f_i^s = \sum_{j=1}^{k_s} F_{ij}^s. \quad (14)$$

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

$$\sum_{j=1}^{k_s} A_j^s \tau_j^s = \alpha \sum_{j=1}^{k_0} A_j \tau_j. \quad (15)$$

The necessary mathematical treatment of the POSITRONFIT model for the least-squares analysis is given in Appendix B.1.

The detailed discussion of POSEDIT in Section 3.2, the input description in Appendix C.1, and the Main Output shown below, illustrate the use of the source correction, the various types of constraints and other input options.

In the following we give an example of a Main Output from POSITRONFIT, which we shall now briefly discuss:

Main Output

```

P O S I T R O N F I T .. VERSION SEP 92 .. JOB TIME 17:13:43 21-SEP-92
*****
34800 CU,RTNS II,61.4 DPA,T30, K=153          07:18:16 29-JUN-87
*****
                                L T I B Z A G
                                4 2 -2 1 0 1 3

TIME SCALE  NS/CHANNEL      :      0.0311
AREA RANGE  STARTS IN CH.   3 AND ENDS IN CH. 511
FIT RANGE   STARTS IN CH. 100 AND ENDS IN CH. 300

RESOLUTION  FWHM (NS)       :      0.2010   0.3050   0.2050
FUNCTION     INTENSITIES (%) :      14.0000  14.0000  72.0000
             SHIFTS (NS)    :      0.0960  -0.0340   0.0000

INITIAL      TIME-ZERO (CH.NO):  105.7000G
PARAMETERS   LIFETIMES (NS)  :      0.1500G  0.2500G  0.1800F  2.0000F
PARAMETERS   LIN.COMB.COEFF. :      0.5000  -0.5000  0.0000  0.0000
             0.0000   0.0000   1.0000 -100.0000

BACKGROUND  FIXED TO MEAN FROM CH. 400 TO CH. 500 =      7.6733
AREA        FIXED SPECTRUM VALUE FROM CH. 10 TO CH. 510 = 9.12394E+05

----- R E S U L T S   B E F O R E   S O U R C E   C O R R E C T I O N -----
CONVERGENCE OBTAINED AFTER 6 ITERATIONS
CHI-SQUARE = 277.53 WITH 197 DEGREES OF FREEDOM
             LIFETIMES (NS)  :      0.1474   0.2271   0.1800F  2.0000F
             INTENSITIES (%) :      42.5439  42.5439  14.7646  0.1476
TIME-ZERO   CHANNEL NUMBER  :      105.6676
TOTAL-AREA  FROM FIT       : 9.12455E+05          FROM TABLE : 9.12456E+05

----- S O U R C E   C O R R E C T I O N -----
SOURCE      LIFETIMES (NS)  :      0.1800   0.3560
CORRECTION  INTENSITIES (%) :      60.2400  39.7600
             TOTAL (%)      :      8.3000

```

```

2ND CYCLE LIFETIMES (NS) : 0.1500G 0.2500G 2.0000F
PARAMETERS

AREA FIXED SPECTRUM VALUE FROM CH. 10 TO CH. 510 = 8.36984E+05

##### F I N A L R E S U L T S #####
                                           L T I B Z A G
                                           3 1 0 1 0 1 3

CONVERGENCE OBTAINED AFTER 9 ITERATIONS
VARIANCE OF THE FIT = 0.971 WITH STANDARD DEVIATION 0.101
CHI-SQUARE = 190.23 WITH 196 DEGREES OF FREEDOM
SIGNIFICANCE OF IMPERFECT MODEL = 39.72 %

LIFETIMES (NS) : 0.1756 0.2835 2.0000
STD DEVIATIONS : 0.0026 0.0398 FIXED

INTENSITIES (%) : 93.8735 5.9926 0.1339
STD DEVIATIONS : 4.1835 4.1751 0.0168

BACKGROUND COUNTS/CHANNEL : 7.6733
STD DEVIATIONS : MEAN

TIME-ZERO CHANNEL NUMBER : 105.6314
STD DEVIATIONS : 0.0102
TOTAL-AREA FROM FIT : 8.37046E+05 FROM TABLE : 8.37046E+05

##### P O S I T R O N F I T #####

```

This output is the result of the input Control Data Set generated by POSEDIT in the example discussed in Section 3.2 (the listing of which is displayed in Appendix C.1). It is not a typical analysis of a spectrum, but rather meant to illustrate a number of features of the program.

After a heading which contains the spectrum headline the key numbers are displayed in the upper right hand corner. “L” indicates the number of lifetime components (k_0), “T” the number of fixed lifetimes, “I” the number and type of intensity constraints (a positive number for fixed intensities, a negative number for linear combinations of intensities, i.e. the number M, Appendix C.1), “B” the type of background constraint (KB, Appendix C.1), “Z” whether or not time-zero is fixed (= 0 when T_0 is free, = 1 when T_0 is fixed) , “A” the type of area constraint (KAR, Appendix C.1), and “G” the number of Gaussians used to describe the resolution curve (k_g). The rest of the upper part of the output reproduces various input parameters, in particular those for the resolution function, the shape of which is fixed, and the initial values (“G” for guessed and “F” for fixed) of the fitting parameters.

The next part contains the outcome of the first iteration cycle. If no convergence has been obtained after 25 iterations, a message will be printed and the iteration procedure discontinued, but still the obtained results are presented.

The next three lines contain information about the goodness of the fit (for details, see Appendix A.2.2).

The next part of the output gives the parameters of the chosen source correction as well as optional initial values of the fitting parameters for the second iteration cycle.

The “Final Results” part prints the number of iterations in the final cycle, followed by three lines with information about the goodness of the fit (for details, see Appendix A.2.2). Then follows a survey of the final estimates of the fitted (and fixed) parameters and their standard deviations. The “area from fit” is calculated as $\sum_j A_j \tau_j$ plus the background inside the “area range” specified at the top of the output. The “area from table” is the total number of counts in the (source corrected) measured spectrum inside the “area range”.

2.2 RESOLUTION

In the RESOLUTION program, the model function is the same as in POSITRONFIT, Eqs. (3) – (10). A few additional formulas relevant to this program are given in Appendix B.2. The purpose of RESOLUTION is to extract the shape of the resolution function. The widths and displacements (Eqs. (8) and (6)) of the Gaussians in the resolution function are therefore included as fitting parameters in this program. However, in order not to have too many fitting parameters (which may lead to ill-defined estimates of the parameters) the intensities of the Gaussians are fixed parameters. For the same reason we normally determine resolution functions by fitting only simple lifetime spectra, i.e. spectra containing only one major lifetime component. The extracted resolution curve may then be used in POSITRONFIT to analyse more complicated spectra. Along the same line, RESOLUTION does not include as many features as does POSITRONFIT, e.g. there is no source correction and hence only one iteration cycle, there is no constraints possible on time-zero and on the area, and the background is always fixed to a mean value between certain channel limits.

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

The detailed discussion of RESEDIT in Section 3.3, the input description in Appendix C.2, and the Main Output shown below illustrate the various input options, including the constraints.

In the following we give an example of a Main Output from RESOLUTION, which we shall now briefly discuss:

**Fitting
parameters**

**Main
Output**

```
R E S O L U T I O N ... VERSION SEP 92 ... JOB TIME 17:16:56 21-SEP-92
```

```

*****
35090 CU-ANN.#6, CU-ANN.#5, CU-ANN.#3, K=156          13:01:35 09-JUL-88
*****
TIME SCALE NS/CHANNEL      :      0.0311
AREA RANGE STARTS IN CH.   3 AND ENDS IN CH. 511
FIT RANGE  STARTS IN CH.  95 AND ENDS IN CH. 250

INITIAL  FWHM (NS)        :      0.2000G  0.3000G  0.2200G
RESOLUTION INTENSITIES (%) :      12.0000  13.0000  75.0000
FUNCTION  SHIFTS (NS)     :      0.1000G -0.0500G  0.0000F

OTHER INIT. TIME-ZERO (CH.NO):  107.0000G
PARAMETERS LIFETIMES (NS)   :      0.1100G  0.4500G  0.1800F  2.0000F
          INTENSITIES (%)   :                          5.0000F

BACKGROUND FIXED TO MEAN FROM CH. 411 TO CH. 512 =      49.3235

##### F I N A L   R E S U L T S #####

CONVERGENCE OBTAINED AFTER 7 ITERATIONS
VARIANCE OF THE FIT = 0.910 WITH STANDARD DEVIATION 0.117
CHI-SQUARE = 132.01 WITH 145 DEGREES OF FREEDOM
SIGNIFICANCE OF IMPERFECT MODEL = 22.74 %

-----
RESOLUTION FUNCTION:
                                                    G W S
          FWHM (NS)      :      0.2799  0.3353  0.2260      3 0 1
          STD DEVIATIONS :      0.0207  0.0138  0.0017
          INTENSITIES (%) :      12.0000  13.0000  75.0000
          SHIFTS (NS)    :      0.0890 -0.0300  0.0000
          STD DEVIATIONS :      0.0156  0.0119  FIXED

-----
LIFETIME COMPONENTS:
                                                    L T I
          LIFETIMES (NS) :      0.1116  0.4005  0.1800  2.0000
          STD DEVIATIONS :      0.0015  0.0095  FIXED  FIXED
          INTENSITIES (%) :      90.3466  4.6330  5.0000  0.0204
          STD DEVIATIONS :      0.2169  0.2072  FIXED  0.0165

-----
TIME-ZERO CHANNEL NUMBER :  107.0305
          STD DEVIATIONS :      0.0454

TOTAL-AREA FROM FIT      :  2.37722E+06          FROM TABLE :  2.37778E+06

SHAPE PARAMETERS FOR RESOLUTION CURVE (NSEC):
          N      2      5      10      30      100      300      1000
FW AT  1/N  0.2427  0.3779  0.4612  0.5821  0.7079  0.8161  0.9239
MIDP AT 1/N  0.0013  0.0035  0.0053  0.0075  0.0064  0.0024 -0.0027

PEAK POSITION IS IN CHANNEL # 107.1528
##### R E S O L U T I O N #####

```

This output is the result of the input Control Data set generated by RES-EDIT in the example discussed in Section 3.3 (the listing of which is displayed in Appendix C.2).

After a heading which includes the spectrum headline, the upper part

of the output reproduces various input parameters in a way that is very similar to the POSITRONFIT output. The important difference is that in RESOLUTION all the FWHMs and all of the displacements (called “shifts” for brevity) except one, may be fitting parameters. In addition, the background is displayed as well as the channels between which it is calculated.

In the “Final Results” part the number of iterations used to obtain convergence is given first. If no convergence has been obtained after 25 iterations, a message will be printed and the iteration procedure discontinued, but still the results found are presented. The next three lines contain information about how good the fit is (for definition of the terms see Appendix A.2.2).

The main part of the output, i.e. the estimated values of the fitted (and fixed) parameters and their standard deviations, follows next (for fixed parameters “FIXED” is printed instead of the standard deviation). This part is divided into two, one giving the parameters for the resolution curve and one the lifetimes and their intensities. Both parts have three key numbers displayed in the upper right hand corner. For the resolution function the “G” indicates the number of Gaussians (k_g), “W” the number of fixed widths, and “S” the number of fixed displacements (shifts). For the lifetime components the “L” indicates the number of these (k_0), “T” the number of fixed lifetimes, and “I” the number and type of intensity constraints. As in POSITRONFIT, a positive value means fixed intensities, while a negative value indicates constraints on linear combinations of intensities, the absolute value giving the number of constraints (i.e. the number M, Appendix C.2). After the estimated time-zero the “area from fit” and “area from table” are given, both calculated as in POSITRONFIT. Finally, for easy comparison of the extracted resolution curve with other such curves, a table of the full width of this curve at different fractions of its peak value is displayed, as well as of the midpoints of the curve compared to the peak position. The latter number clearly shows possible asymmetries in the resolution curve.

2.3 ACARFIT

For ACAR curves (ACAR = Angular Correlation of Annihilation Radiation) the functional form of the components is normally not known a priori. In many cases, however, especially for non-metallic systems, they are adequately described by Gaussians or sums of Gaussians. For many metals ACAR curves contain components which are described more accurately by truncated parabolas than by Gaussians. In some cases, off-centre peaks may appear (“side peaks”). The model function used in ACARFIT is therefore a sum of Gaussians and truncated parabolas, each convoluted with a Gaussian to allow for some broadening of the parabola. (The

**Model
function**

program subtracts a constant background from the spectrum before the fitting.) Each component may be considered to be composed of two peaks, symmetrically displaced around a reference angle θ_0 (“angle-zero”). Only for “side peaks” will this displacement differ from zero:

$$F(\theta) = \frac{1}{2} \sum_{j=1}^{k_g} J_j^g (g(\theta - \theta_0 + \theta_j^g, \sigma_j) + g(\theta - \theta_0 - \theta_j^g, \sigma_j)) + \frac{1}{2} \sum_{j=1}^{k_p} J_j^p (p(\theta - \theta_0 + \theta_j^p, \theta_j^c, \omega_j) + p(\theta - \theta_0 - \theta_j^p, \theta_j^c, \omega_j)), \quad (16)$$

where k_g is the number of Gaussians and k_p of folded-parabola terms; either of k_g and k_p may be zero. Moreover g stands for a normalized Gaussian

$$g(\theta, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\theta^2}{2\sigma^2}\right), \quad (17)$$

and $p = P * g$ for a normalized folded parabola,

$$p(\theta, \theta^c, \omega) = \int_{-\infty}^{\infty} P(\theta - \theta', \theta^c) g(\theta', \omega) d\theta', \quad (18)$$

P being the truncated parabola

$$P(\theta, \theta^c) = \begin{cases} 3/(4\theta^c) (1 - (\theta/\theta^c)^2), & |\theta| \leq \theta^c \\ 0, & |\theta| > \theta^c. \end{cases} \quad (19)$$

Thus, in (16) θ_j^g is the displacement of the j th Gaussian and σ_j its standard deviation; for the j th folded-parabola term, θ_j^p is its displacement, θ_j^c the angle of truncation (sometimes called the cut-off angle or Fermi angle), and ω_j is the broadening of the parabola. Finally, J_j^g is the absolute intensity of the j th Gaussian pair, and similarly, J_j^p the absolute intensity of the j th parabolic term. The independent angular variable in (16) is θ .

Fitting parameters

The fitting parameters in ACARFIT are the widths of the Gaussian components (FWHM_j^g), of the parabolas (θ_j^c), and of the Gaussians that are folded with the parabolas (FWHM_j^p). The width FWHM_j^g corresponds to the standard deviation σ_j in (16), and similarly FWHM_j^p corresponds to the parabola broadening ω_j (in the in- and output to the program we prefer FWHM, but internally we use σ or ω , cf. the relation (8)). Furthermore, the displacements ¹ (θ_j^g and θ_j^p) of the Gaussian and parabolic components, and their relative intensities

$$I_j^n = J_j^n / \left(\sum_{k=1}^{k_g} J_k^g + \sum_{k=1}^{k_p} J_k^p \right) \quad (n = g \text{ or } p), \quad (20)$$

as well as angle-zero (θ_0), are fitting parameters. Each of these parameters may be constrained to a fixed value. In another type of constraint you

¹As a central peak is a confluent and degenerate form of a side-peak pair, the fitting procedure is unable to separate an initial central peak into side peaks.

Constraints

may put one or more linear combinations of intensities equal to zero in the fitting:

$$\sum_{j=1}^{k_g} h'_{\ell_j} I_j^g + \sum_{j=1}^{k_p} h''_{\ell_j} I_j^p = 0. \quad (21)$$

This type of constraint is particularly useful for the analysis of measured ACAR curves in terms of components of known shapes (see Section 4.3).

Appendix B.3 contains an elaboration of the mathematical model required for the least-squares analysis.

The detailed discussion of ACAREFIT in Section 3.4, the input description in Appendix C.3, and the Main Output shown below illustrate the various input options, including the constraints.

The measured ACAR curve to be analysed by ACARFIT may be incorporated in the Control Data Set in different ways. The standard input to the program will be the data set $\{\theta_i, y_i, s_i\}$ ($i = 1, \dots, n$), where the angles θ_i are in mrad; in general, they will not be equidistant. The y_i are counts which may be normalized, and s_i are the estimated standard deviations of y_i . However, if the points are equidistant, the θ_i -table may be omitted, and only their separation entered. Likewise, if y_i represent the raw data (i.e. not normalized) the s_i -table may be omitted and the program will take the s_i -values as $\sqrt{y_i}$ (like in the lifetime programs).

In the following we give an example of a Main Output from ACARFIT, which we shall now briefly discuss:

Main Output

```

A C A R F I T      . . . . . VERSION SEP 92      . . . . . JOB TIME 17:17:43 21-SEP-92
*****
01591 COPPER,5N,CW 70%                                06-AUG-86
*****

NUMBER OF DATA POINTS :      55
RANGE OF DATA POINTS  :  -17.24 MRAD  TO  +17.77 MRAD

REQUESTED FIT RANGE    :  -17.00 MRAD  TO  +17.00 MRAD
TRUNCATED TO          :  -15.64 MRAD  TO  +16.18 MRAD

SUBTRACTED BACKGROUND :      0.0000

INITIAL PARAMETERS:

          GAUSSIAN      PARABOLA      PARABOLA
TERM TYPE      POSITION      FWHM      CUT-OFF      BROADENING      INTENSITY
#              (+-MRAD):    (MRAD):    (MRAD):    (MRAD):    (%) :
1 PARABOLA:    0.0000F      16.3685F      5.5010F      2.5191F      LIN.COM.
2 GAUSSIAN:    0.0000F      4.7605F      2.0000G      3.0000G      LIN.COM.
3 GAUSSIAN:    0.0000F      8.0000G      2.0000G      3.0000G      LIN.COM.
4 GAUSSIAN:    0.0000F      8.0000G      2.0000G      3.0000G      LIN.COM.
5 PARABOLA:    6.0000G      2.0000G      3.0000G      3.0000G      LIN.COM.

ANGLE-ZERO (MRAD)      :      0.0000G
NORMALIZATION RANGE (+-MRAD):    13.0000

NONZERO COEFFICIENTS IN THE 2 LINEAR-COMBINATION CONSTRAINTS:
CONSTRAINT # 1:      TERM # 1      2.7286      TERM # 3      27.4036

```

CONSTRAINT # 2: TERM # 1 -75.3250 TERM # 2 27.4036

F I N A L R E S U L T S

CONVERGENCE OBTAINED AFTER 22 ITERATIONS
VARIANCE OF THE FIT = 1.234 WITH STANDARD DEVIATION 0.211
CHI-SQUARE = 55.54 WITH 45 DEGREES OF FREEDOM
SIGNIFICANCE OF IMPERFECT MODEL = 86.51 %

ACAR COMPONENTS: GAU P W PAR P C B INT
3 3 2 2 1 1 1 -2

TERM #	TYPE	POSITION (+-MRAD)	GAUSSIAN FWHM (MRAD)	PARABOLA CUT-OFF (MRAD)	PARABOLA BROADENING (MRAD)	INTENSITY (%)
1	PARABOLA:	0.0000		5.5010	2.5191	23.2797L
1	ST.DEV.:	FIXED		FIXED	FIXED	0.0986
2	GAUSSIAN:	0.0000	16.3685			63.9896L
2	ST.DEV.:	FIXED	FIXED			0.2710
3	GAUSSIAN:	0.0000	4.7605			-2.3180L
3	ST.DEV.:	FIXED	FIXED			0.0098
4	GAUSSIAN:	0.0000	8.1048			14.6631
4	ST.DEV.:	FIXED	0.1455			0.3984
5	PARABOLA:	6.6425		0.0237	2.7604	0.3856
5	ST.DEV.:	0.2604		118.5141	1.3873	0.1618

ANGLE-ZERO (MRAD) : -0.0127
ST.DEV. : 0.0051

AREA TYPE	AREA RANGE (MRAD)	MRAD * COUNTS
FIT (TOTAL)	(-INFINITY, +INFINITY)	1.92487E+05
FIT (NORMALIZATION)	(PEAK - 13.00, PEAK + 13.00)	1.84913E+05
FIT (RANGE OF FIT)	(-15.64, +16.18)	1.89752E+05
TABLE	(-15.64, +16.18)	1.89861E+05

NORMALIZED PEAK COUNT (+- 13.00 MRAD) (1/MRAD): 0.082950

SHAPE PARAMETERS FOR ACAR CURVE (MRAD):

N	2	5	10
FW AT 1/N	10.2921	18.3399	24.3354

A C A R F I T #####

01591 COPPER,5N,CW 70% 06-AUG-86

This output is the result of the input Control Data Set generated by ACAREEDIT in the example discussed in Section 3.4 (the listing of which is displayed in Appendix C.3). This example is not a typical analysis of this type of an ACAR curve, but rather meant to illustrate several features of the program in a single analysis.

After the top heading which includes the spectrum headline, the number of data points and the mrad-range they cover are displayed. The next line

shows the range of angles requested by the input that are to be included in the analysis. The actually included angular range may be somewhat smaller, since it has to begin at a data point and end at one, as shown in the next line. After the background information, a table of the initial parameter values follows (“F” for fixed and “G” for guessed) with the parameters for the Gaussians and the parabolas clearly separated (and for each component listed horizontally; in the lifetime programs the listing is vertically). Notice that in ACARFIT the sequence of the components is completely unrestricted with respect to their type, i.e. we need not let the Gaussians precede the parabolas as in the model formulation (16). The intensity column either shows the values of fixed intensities or (as in the present example) that one or more linear combinations of intensities are constrained to zero as in Eq. (21). Next follows the initial angle-zero value and the normalization range used for determining the normalized peak count. When “linear combination” constraints are used, the next lines give the non-zero coefficients for the various constraints. All the other coefficients in (21) equal zero, (but are not shown to save space and make the information easier to overlook. This is particularly important in the ACARFIT program since many linear combinations and many terms may often be used to analyse ACAR curves (see Section 4.3)). A similar convention is followed when preparing the input (see Section 3.4 and Appendix C.3).

The “Final Results” part first shows the number of iterations used to obtain convergence. If 25 iterations do not lead to convergence, a message will be printed and the iteration procedure discontinued, but still the results found are presented. The next three lines contain information about the goodness of the fit (for details, see Appendix A.2.2).

The main part of the output, i.e. the estimated values of the fitted (and fixed) parameters, follows next. For fixed parameters, “FIXED” is printed instead of the standard deviation, and if “linear-combination” constraints are used, an “L” follows those intensities that have been multiplied by non-zero coefficients. In the upper right-hand corner of this part, the key numbers are given. “GAU” refers to the number of Gaussians, the first “P” to the number of these that have fixed positions, and “W” to the number of Gaussians with fixed widths. “PAR” refers to the number of parabolas, the “P” that follows to the number of parabolas with fixed positions, “C” to the number of parabolas with fixed cut-off angles, and “B” to the number of parabolas with fixed widths of the Gaussians with which they are convoluted (i.e. the “parabola broadening”). The “INT” indicates the number and type of the intensity constraints. The absolute value of the number gives the number of constraints. If the sign is positive, one or more intensities are fixed, while a negative sign means that the constraints are of the “linear-combination” type.

In the analysis shown in the Main Output, the first parabola and the three Gaussians are central components while the 5th component consists of a

“sidepeak” pair of parabolas. The total intensity of the pair is given in the “Intensity” column.

The next few lines give the values of areas of the fitted curve within three different regions and, for comparison, the area of the measured ACAR curve (calculated from a linear interpolation between the measured points).

Finally, before the repetition of the spectrum headline at the bottom, the width of the fitted curve is given at $1/2$, $1/5$, and $1/10$ of the peak value, in order to give a very simple indication of the shape of the ACAR curve. Occasionally this shape computation is abandoned because the curve is bi-modular or otherwise difficult to handle.

3 The Input Editors

Each of the three fitting programs described above requires — together with the spectrum to be analysed — a set of input data, e.g. some characteristic parameters of the lifetime or angular correlation spectrometer, guesses of the parameters to be fitted, and possible constraints on these parameters. These data, which are referred to as Control Data, must conform with the input prescriptions for the fitting programs given in Appendix C. In order to facilitate the generation of error-free Control Data Sets, and thus relieve the user from the errorprone task of correct input formatting, three programs were developed (POSEDIT, RESEDIT, and ACAREEDIT), one for each fitting program. With these –EDIT programs one can interactively generate and/or edit Control Data Sets to the fitting programs. Each –EDIT program generates an output file (a so-called Control File) that provides the necessary input data for one of the three fitting programs. Much effort was devoted to design the editors in such a way as to ensure uncorrupted output files. The interplay between the various programs and data files is illustrated in Fig. 1 using POSITRONFIT and its associated programs and files as example (the data processing is analogous in the other two fitting programs.)

Control Data

To summarize, there is a Control File with (Control) Data Sets, a Spectrum File with (Spectrum) Data Sets, and a Communication File with (Communication) Data Sets. Moreover, each –EDIT program has a Restart File with the most recently edited Control Data Set.

We shall subsequently describe each of the three –EDIT programs in detail, although to a large extent the programs are self-explanatory. However, first we discuss common general features of the three programs. In this and the following discussion printouts of the screen are frequently used to illustrate the text. For clarity, we have underlined those parts which have been entered from the keyboard. All the rest has been displayed by the program. The key “Return” (or Carriage Return or Enter) will be indicated by CR.

3.1 Common Features of the Input Editors

The purpose of running one of the –EDIT programs is to generate a Control File. This file contains one or more Control Data Sets. Each Control Data Set serves as the necessary input to a fitting program for a single fit of a measured spectrum.

Control File

The –EDIT programs are interactive and menu oriented. On starting the program, a Main Menu is displayed which lists 7 – 10 Local Menus. Each Local Menu deals with a fraction of the Control Data Set (a so-called Data Block which could be, for example, the measured spectrum, the number of

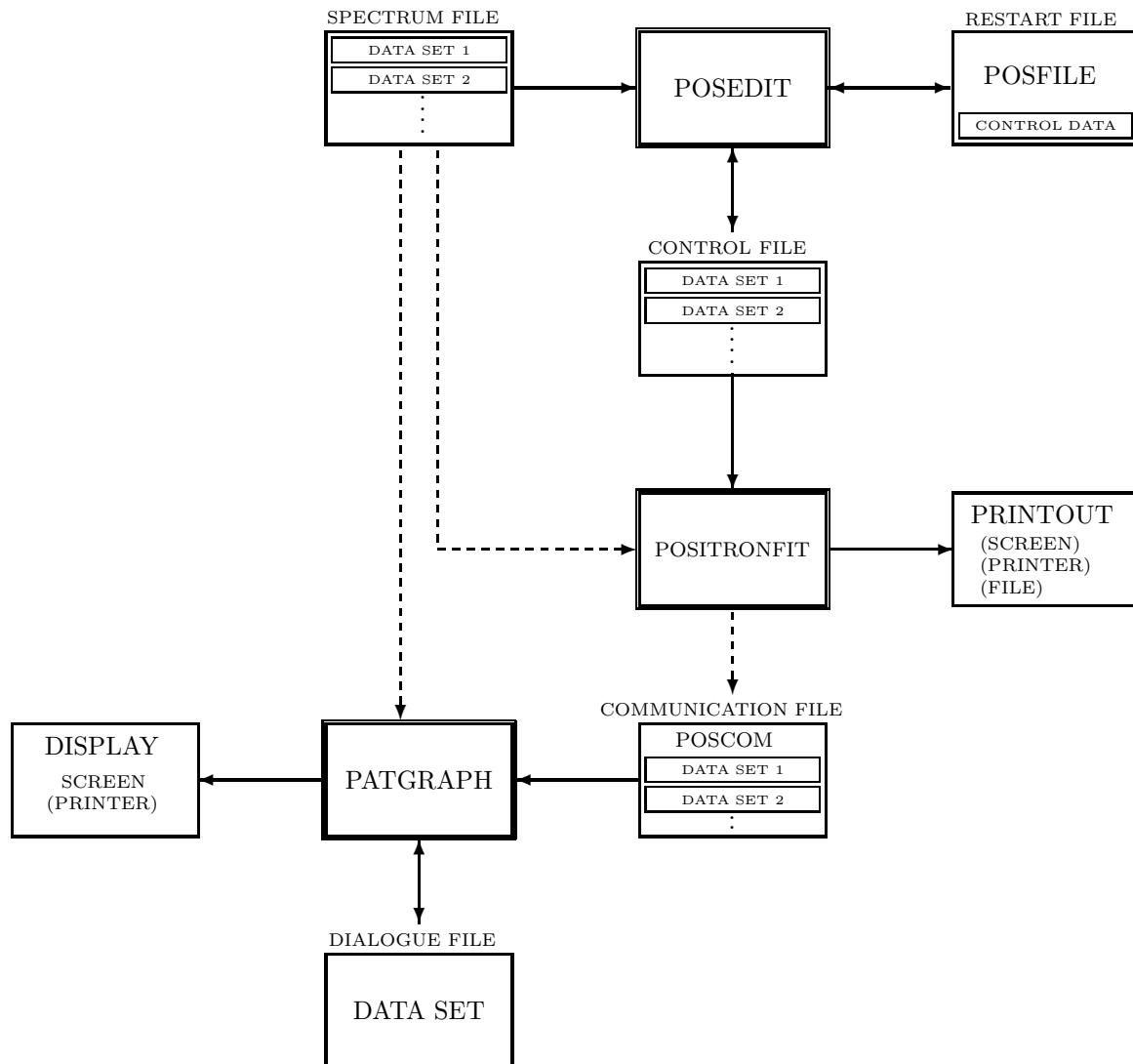


Figure 1. Schematic representation of the interaction between programs, files and output devices, using the POSITRONFIT system as example. The lines with arrows indicate data transfer between a program and a file or output device. The dashed lines represent optional data transfer. The POSFILE contains default input values for the editing program POSEDIT. These values are generated by the previous POSEDIT run. The Spectrum File contains one or more of the measured spectra, each spectrum preceded by a headline. The Control Data Sets are generated by POSEDIT and act as input to the fitting program POSITRONFIT. The POSCOM file contains Communication Data Sets as optional output from POSITRONFIT and acts as input to the graphics program PATGRAPH, which displays spectra on the screen or on the printer.

lifetimes and their initial values, the source correction, etc.). To generate the Control Data Set, one goes through the Local Menus of interest.

It is possible to diverge from the “natural” or “default” order of menu items, both in the Main Menu and in the Local Menus; as illustrated in subsequent examples, you may jump freely from one item to another.

A Local Menu consists of a number of questions. In each question a previously chosen (or default) parameter value is displayed. One may keep this value or enter a new one. It is possible to find the parameter(s) rapidly that you want to change, without dealing with the other parameters in the Main or Local Menu.

Local Menu

When you run the program, the displayed parameter values are those which were generated when the program was used the previous time. (These data were saved in one of the files POSFILE, RESFILE, or ACARFILE and serve as restart input to the –EDIT program; see Fig. 1). Alternatively, you may choose any previously generated Control File as input to the –EDIT program. In the absence of a Restart File, for example during the first start-up, or when you start from a different directory or drive, the editor uses default data, often with very little physical meaning.

Restart File

The parameter values to be typed-in can be numbers, answers (e.g. Yes or No), or text strings (e.g. file name, spectrum label). Some parameters are initial, guessed values of the parameters to be fitted. A “G” after a number signifies that the number is an initial (Guessed) value, an “F”, that the parameter is constrained (Fixed) to the value given. If neither an F nor a G is entered, the number is taken to be a guessed value. Other letters are used in other contexts, too, as specified in the description of the various menus.

Apart from the keyed-in data and the Restart File (or another Control Data Set), an –EDIT program requires input of a spectrum (or at least its headline), see Fig. 1. The measured spectra are supposed to be stored as Spectrum Data Sets in the so-called Spectrum File, each spectrum preceded by a headline. The headlines are used to identify the individual spectra.

Spectrum File

Additional information may be associated with an ACAR spectrum, viz. the number of data points, a table of angles and a table of standard deviations for the data points, in total referred to as the Spectrum Data Set. It is optional what additional information, apart from the spectrum, that is included in the Spectrum Data Set, as discussed in Section 3.4 and Appendix C.3.

In the following, we show the typical structure of a local menu. A heading is displayed at first, e.g.:

```
POSEDIT SELECTION IS # 6: LIFETIMES AND INTENSITY CONSTRAINTS
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): □
```

Local Menu: structure

The first line identifies the menu. The second makes it possible to accept

the displayed menu, to choose another menu (e.g. if you enter *3 followed by Return (CR), Local Menu number 3 will appear), or ask for Help by typing *H and CR (see further below about "Help"). The symbol □ indicates the cursor position. This will normally be at the end of the present line, but on some computer systems it is positioned at the beginning of the next line.

If you accept the present menu by striking CR, the line(s) to follow will display an overview of (all or some of) the parameters belonging to the menu. Next, the same parameters are displayed, one at a time, and the cursor stops after the parameter, e.g.:

```
LOCAL MENU # 6: LIFETIMES AND INTENSITY CONSTRAINTS
1. NUMBER OF LIFETIMES      :      4 □
```

This display shows that the current selection is 4 lifetimes. If you want to keep 4 lifetimes, strike CR and the following lines are displayed:

```
2. LIFETIMES (NS)          :   0.1300G   0.3300F   0.9500G
                           1.7000F
   LIFETIME      # 1:    0.1300G □
```

If the first lifetime shall no longer be a free fitting parameter with an initial, guessed value of 0.13, but rather a fixed lifetime of 0.125, this latter value should be typed-in followed by an F and CR. The display will now read:

```
LIFETIME      # 1:    0.1300G 0.125F CR
LIFETIME      # 2:    0.3300F □
```

In order to change the fixed value of the second lifetime to a guess of 0.330 ns, only a G need be entered, and the display shows:

```
LIFETIME      # 2:    0.3300F G CR
LIFETIME      # 3:    0.9500G □
```

Similarly, a guessed value can be changed to a fixed value by typing F. The third and fourth initial lifetime value can now be entered in the same way as above.

When all parameters in a menu have been entered (or the displayed ones accepted), a summary of the menu will be displayed, and you can either accept the contents of the menu (the Data Block) by striking CR or modify it again by entering * <item no.> CR. The <item no.> refers to the numbers to the left in the menu summary, e.g.:

```
SUMMARY FOR THIS MENU:
1. NUMBER OF LIFETIMES      :      4
2. LIFETIMES (NS)          :   0.1250F   0.3300G   0.9500G
                           1.7000F
3. INTENSITY CONSTRAINT TYPE : F
   INTENSITIES              :  VARIABLE   VARIABLE   54.0000
                           VARIABLE
```

```
ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): *2 CR
```

The lifetimes will now be displayed again and new values may be entered as described above.

When the modified content of the menu is accepted by striking a CR after the menu summary, a line, e.g.

```
END OF POSEDIT MENU # 6: LIFETIMES AND INTENSITY CONSTRAINTS
```

signals the end of the menu. This message is then followed by the heading of the next menu.

The last menu, called "EXPORT ... CONTROL DATA", creates the Control File which contains the Control Data Set that has been generated by the -EDIT program.

**Leaving
-EDIT program**

You may jump to the Export Menu from the top of any Local Menu by typing *10 if you are in POSEDIT (*8 in RESEDIT, *7 in ACAREEDIT). For convenience, you are allowed to type merely "a large number" like *99 in all three programs.

Suppose in the following example we enter the POSEDIT Export Menu:

```
LOCAL MENU # 10: EXPORT POSITRONFIT CONTROL DATA  
1. LIST CONTROL DATA (Y,N) : Y n CR
```

Before you save the Control Data, you may want to display them on the screen to make a final check of the values. This can be accomplished by typing Yes to the question above. In the present example, though, a No was entered and no listing occurs. After a CR, the following question appears:

```
2. SAVE DATA IN FILE WITH NAME: OCURTNS.POS CR
```

The name OCURTNS.POS is the name of the Control File generated when the -EDIT program was run the previous time, or, if another Control File was chosen in Menu 1, the name of that file. If you plan to save the Control Data in a file with another name, this name must be entered.

In this example, CR was entered instead, and you are informed that the file you have chosen already exists and asked whether you wish to Append your Control Data Set at the end of the file (this is always the displayed default value in order to avoid accidental overwriting of the file) or Overwrite the file, or not save the data at all (Quit):

```
FILE EXISTS: OCURTNS.POS  
(A=APPEND, O=OVERWRITE, Q=QUIT): A q CR
```

```
SUMMARY FOR THIS MENU:  
1. LIST CONTROL DATA (Y,N) : N  
2. SAVE DATA IN FILE WITH NAME: OCURTNS.POS
```

```
** NO NEW CONTROL DATA WILL BE SAVED **
```

```
CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): *2 CR
```

The "Append" option can be used to generate a Control File that contains two or more Control Data Sets. When using such a Control File as input to

a fitting program, the fitting program will carry out two or more spectrum analyses, one for each Control Data Set.² However, here the last possibility (Q) was chosen, but instead of accepting the summary, you wish to change the file name. Therefore, *2 has been entered. After the CR, question 2 is then repeated and a new file name typed:

```
2. SAVE DATA IN FILE WITH NAME: OCURTNS.POS peter.pos CR
```

After a CR the Control Data is written into the new file, PETER.POS, and a new summary displayed:

```
SUMMARY FOR THIS MENU:
1. LIST CONTROL DATA (Y,N)      : N
2. SAVE DATA IN FILE WITH NAME: PETER.POS

** CONTROL DATA WAS WRITTEN TO A NEW FILE **

CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
```

With the CR you continue to the end of the program:

```
=====TERMINAL POINT FOR POSEDIT MAIN MENU=====
KEY <*> OR <*E> TO EXIT FROM EDITOR, <RETURN> TO RECYCLE THE MENU: * CR

END OF POSEDIT SESSION
```

By typing * as above or *E you can terminate the program. If, instead, you want to generate one more Control Data Set, strike CR and you return to the beginning of the –EDIT program.

From the above example one notices that the –EDIT programs accept both small and capital letters.

The –EDIT programs contain a number of checks on the parameter values entered by the user. If an input error has been made, in some cases you are asked to enter a new value; in other cases a warning is given, and the program automatically changes the input parameter to an acceptable value. We give some examples:

Illegal input:

```
LOCAL MENU # 1: IMPORT POSITRONFIT CONTROL DATA
CONTROL DATA ORIGINATES FROM FILE: A.POS
1. LOAD CONTROL DATA FROM A FILE (Y,N): N q CR
ILLEGAL INPUT: Q

1. LOAD CONTROL DATA FROM A FILE (Y,N): N CR
```

Warning:

```
LOCAL MENU # 10: EXPORT POSITRONFIT CONTROL DATA
1. LIST CONTROL DATA (Y,N)      : N CR

*** WARNING ***: CHANNEL LIMIT FOR FIT RANGE
```

²A Control File may contain Data Sets generated by different –EDIT programs. A particular fitting program will only read those Data Sets that are relevant for it and skip the others.

**Small/Capital
Letters**
**Warnings and
Consistency
Checks**

REDUCED FROM 1111 TO 512

2. SAVE DATA IN FILE WITH NAME: A.POS

Number out of range:

LOCAL MENU # 3: SPECTRUM

1. CHANNELS IN SPECTRUM : 512 10000 CR

LIMIT VIOLATION - VALID BOUNDS ARE:

MIN MAX

1 8192

ILLEGAL INPUT: 10000

1. CHANNELS IN SPECTRUM : 512 CR

In the last example a number was out of range. All numeric data to the editor programs are bounded in this way to prevent the production of unreasonable input to the fitting programs.

In order to generate the Control File you can of course use any editor available, but you will then be without the safety provided by the various parameter checks included in the present -EDIT programs.

At any point in a program you may ask for "help" by entering *H. The **Help** following will then be displayed:

OPTIONS FOR INPUT LINE, VALID BOTH IN MAIN AND LOCAL MENUS:

<RETURN> ACCEPT CURRENT ITEM
*EXIT (*E) JUMP TO BOTTOM OF PRESENT MENU
* SAME AS ABOVE
*HELP (*H) GIVE ON-LINE HELP (THIS LIST)
*MAIN (*M) SHOW MAIN MENU
*<NUMBER> JUMP TO ITEM NO. <NUMBER>
OTHERWISE ENTER NUMERICAL AND/OR TEXT INPUT

With *E (or just *) you jump to the bottom of the menu in which you are presently working, i.e. from a Local Menu out to the heading of the next Local Menu (i.e. to the Main Menu level). From this level a * (or *E) will take you to the terminal point of the -EDIT program. If you want to see the Main Menu, entering *M will display it (from any position in the editor). The other lines in the "Help display" have already been explained above.

After this description of some common, general features of the -EDIT programs, we give in the following a brief description of the special features of each of the three programs. (You may fail to reproduce the dialogues exactly, as the reactions sometimes depend on previous conditions.)

Since the -EDIT programs contain many features, it is not possible to discuss them all in detail. Special features which are equivalent in the three programs may therefore be illustrated in just one of them.

3.2 POSEDIT

Introduction

In this Section we describe in some detail the interactive program POSEDIT. The purpose of the program is to generate a Control File that can act as input to POSITRONFIT. The functioning of POSEDIT is therefore strongly correlated to the input options in POSITRONFIT.

Main Menu

On starting the program, after the display of a heading the Main Menu is displayed. It contains a listing of 10 Local Menus. In the following, we shall discuss each of these:

```
POSEDIT MAIN MENU:
  1. IMPORT POSITRONFIT CONTROL DATA
  2. OUTPUT OPTIONS
  3. SPECTRUM
  4. AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
  5. RESOLUTION FUNCTION
  6. LIFETIMES AND INTENSITY CONSTRAINTS
  7. BACKGROUND CONSTRAINTS
  8. AREA CONSTRAINTS
  9. SOURCE CORRECTION
 10. EXPORT POSITRONFIT CONTROL DATA
```

```
POSEDIT SELECTION IS # 1: IMPORT POSITRONFIT CONTROL DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

Menu 1: Import Control Data

After the Main Menu, Local Menu No. 1 is automatically selected. If accepted as above, the following is displayed:

```
LOCAL MENU # 1: IMPORT POSITRONFIT CONTROL DATA
CONTROL DATA ORIGINATES FROM FILE: NIELS.POS
1. LOAD CONTROL DATA FROM A FILE (Y,N): N y CR
```

In order to avoid generating an entirely new Control Data Set for each run of POSEDIT, the Data Set from the previous run has been saved by the program (in POSFILE) to be used as default values on restart. To identify the Data Set, the name of the Control File in which the user saved the Data Set is also displayed (i.e. NIELS.POS). However, it is optional to choose another input Control File by entering Y (like in the example), and the display will read:

```
NAME OF INPUT CONTROL FILE: NIELS.POS rtn2.pos CR
```

After the name has been entered, this other Control File will be read from the disk. If the file is not found, the question is repeated (the specific error code given below depends on the FORTRAN compiler):

```
ERROR: COULD NOT OPEN OLD FILE WITH NAME RTN2.POS
(FORTRAN ERROR CODE IOSTAT = 2015)
```

```
NAME OF INPUT CONTROL FILE: NIELS.POS CR
```

The Control File may contain several Control Data Sets and the wanted Data Set must be selected:

```
READ CONTROL DATA SET NUMBER:      1 CR
```

You may then inspect the content of the Data Set (but in this example we choose not to.) Finally you obtain the Menu summary. If acceptable, a CR takes you to the end of the Menu:

2. LIST CURRENT CONTROL DATA SET (Y,N): N CR

SUMMARY FOR THIS MENU:

1. LOAD CONTROL DATA FROM A FILE (Y,N): Y

2. LIST CURRENT CONTROL DATA SET (Y,N): N

CONTROL DATA WAS LOADED FROM A FILE

CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

END OF POSEDIT MENU # 1: IMPORT POSITRONFIT CONTROL DATA

Local Menu No. 2 deals with the output options (for POSITRONFIT). The standard output is the “Main Output”, which contains all input and fitted parameters, etc. (see Section 2.1). In addition one may choose to have a listing of the whole Control Data Set as read by POSITRONFIT (“Input Echo”). Further, a printout of the fitting parameters after each iteration during the fitting process (“Iteration Printout”), a graphical representation of the deviations between the fitted and the measured spectrum (“Residual Plot”), and a printout of the matrix of total correlations between the fitted parameters (“Correlation Matrix”) are possible. Finally, one may choose to save in a “Communication File” (POSCOM) a data set that contains the measured spectrum (optional), the Main Output and the Correlation Matrix for graphic display or other purposes (“Communication Data Set”). If all of these options are chosen, Menu 2 looks like this:

Menu 2: Output Options

POSEDIT SELECTION IS # 2: OUTPUT OPTIONS

ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 2: OUTPUT OPTIONS

1. INPUT ECHO ? (Y,N) : N y CR

2. ITERATION PRINTOUT ? (Y,N) : N y CR

3. RESIDUAL PLOT ? (Y,N) : N y CR

4. CORRELATION MATRIX ? (Y,N) : N y CR

5. COMMUNICATION DATA ? (Y,N) : N y CR

SUMMARY FOR THIS MENU:

1. INPUT ECHO ? (Y,N) : Y

2. ITERATION PRINTOUT ? (Y,N) : Y

3. RESIDUAL PLOT ? (Y,N) : Y

4. CORRELATION MATRIX ? (Y,N) : Y

5. COMMUNICATION DATA ? (Y,N) : Y

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

END OF POSEDIT MENU # 2: OUTPUT OPTIONS

The third Local Menu is responsible for reading the measured spectrum. First the number of channels contained in the spectrum must be entered, secondly the proper FORTRAN format for reading the spectrum (notice that I-formatting is not allowed), thirdly the name of the file which contains the spectrum, and subsequently, the label by which the wanted spectrum is identified must be entered. Finally, you can choose whether

Menu 3: Spectrum

to incorporate the spectrum in your Control Data Set and Communication Data Set or not. If the spectrum is not incorporated in the Control Data, it will also be left out of the Communication Data Set. The latter option can be chosen to save space, but requires that there is access to the spectrum when you run POSITRONFIT:

```
POSEDIT SELECTION IS # 3: SPECTRUM
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 3: SPECTRUM
1. CHANNELS IN SPECTRUM      : 512 CR
2. SPECTRUM FORMAT          : (F7.0/(10F7.0)) CR
3. SPECTRUM FILE, NAME IS   : GR34800.DAT CR
4. SPECTRUM LABEL          : 34802 34800 CR
5. SPECTRUM TO BE INCORPORATED IN CONTROL DATA ? (Y,N): Y CR
```

In the present example the label is a number. However, it may be any string of max. 32 characters. After the Menu summary, the program will try to locate the spectrum by scanning through the Spectrum File in order to find a headline, the beginning of which matches the entered spectrum label. For a successful match, the spectrum label must coincide, up to its last nonblank character, with the first part of a spectrum headline. If the spectrum is found it will be read and its headline will be displayed (if it is not found, a message will appear):

```
SUMMARY FOR THIS MENU:
1. CHANNELS IN SPECTRUM      : 512
2. SPECTRUM FORMAT          : (F7.0/(10F7.0))
3. SPECTRUM FILE, NAME IS   : GR34800.DAT
4. SPECTRUM LABEL          : 34800
5. SPECTRUM TO BE INCORPORATED IN CONTROL DATA ? (Y,N): Y
```

```
ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
READING OF SPECTRUM FROM FILE WAS REQUESTED
```

```
SPECTRUM LABEL LOCATED:
34800 CU,RTNS II,61.4 DPA,T30, K=153                07:18:16 29-JUN-87
```

```
END OF POSEDIT MENU # 3: SPECTRUM
```

Menu 4: Ranges

In Local Menu 4 the two first questions deal with the range of channels that shall be included in calculations of the spectrum area:

```
POSEDIT SELECTION IS # 4: AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 4: AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
1. AREA RANGE STARTS IN CH. # : 3 CR
2. AREA RANGE ENDS IN CH. # : 511 CR
```

This area is calculated because after the fitting, a comparison is made between the total area from the fit and from the table (Section 2.1). Area constraints may also be imposed as an option (Menu No. 8). By defining the Area Range, one can exclude channels with special contents (e.g., measuring time).

Normally, fewer channels are included in the fitting than in the area cal-

ulation. Therefore, the Fit Range is defined next. First, though, the peak channel of the spectrum is displayed, so that you have a clear indication of the position of the spectrum:

```
SPECTRUM PEAK IS IN CHANNEL: 109
3. FIT RANGE STARTS IN CH. # : 100 CR
4. FIT RANGE ENDS IN CH. # : 510 300 CR
```

Finally, the time-equivalent of one channel width and the initial (guessed or fixed) value of the channel number equivalent to zero time shall be entered. Of all the parameters entered in this Local Menu, only time-zero may be a fitting parameter:

Time scale
Time-zero

```
5. TIME SCALE (NS/CHANNEL) : 0.0311 CR
6. TIME-ZERO (CHANNEL NUMBER) : 105.7000F G CR
```

SUMMARY FOR THIS MENU:

```
1. AREA RANGE STARTS IN CH. # : 3
2. AREA RANGE ENDS IN CH. # : 511
3. FIT RANGE STARTS IN CH. # : 100
4. FIT RANGE ENDS IN CH. # : 300
5. TIME SCALE (NS/CHANNEL) : 0.0311
6. TIME-ZERO (CHANNEL NUMBER) : 105.7000G
```

```
ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF POSEDIT MENU # 4: AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
```

Local Menu number 5 defines the resolution function of the lifetime spectrometer. The resolution function is described as a sum of Gaussians, the centroids of which may be shifted with respect to each other. A maximum of six Gaussians can be used to describe the resolution function. After defining the number of Gaussians, the full width at half maximum (FWHM) of each Gaussian is entered, then their relative intensities, and finally the shift of each Gaussian:

Menu 5:
Resolution
Function

```
POSEDIT SELECTION IS # 5: RESOLUTION FUNCTION
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

LOCAL MENU # 5: RESOLUTION FUNCTION

```
1. NUMBER OF GAUSSIANS : 1 3 CR
2. FWHM (NS) : 0.2400 1.0000 1.0000
   FWHM (NS) # 1: 0.2400 .2010 CR
   FWHM (NS) # 2: 1.0000 .3050 CR
   FWHM (NS) # 3: 1.0000 .2050 CR
3. INTENSITY (%) : 100.0000 0.0000 0.0000
   INTENSITY (%) # 1: 100.0000 14 CR
   INTENSITY (%) # 2: 0.0000 14 CR
   INTENSITY (%) # 3: 0.0000 72 CR
4. SHIFT (NS) : 0.0000 0.0000 0.0000
   SHIFT (NS) # 1: 0.0000 .0960 CR
   SHIFT (NS) # 2: 0.0000 -.0340 CR
   SHIFT (NS) # 3: 0.0000 CR
```

SUMMARY FOR THIS MENU:

```
1. NUMBER OF GAUSSIANS : 3
2. FWHM (NS) : 0.2010 0.3050 0.2050
3. INTENSITY (%) : 14.0000 14.0000 72.0000
4. SHIFT (NS) : 0.0960 -0.0340 0.0000
```


In the example above, we have finally typed *3 in order to illustrate the other type of intensity constraint. In this constraint a linear combination of the intensities is put equal to zero. This possibility is chosen by entering an "L", as illustrated by the following example.

```

3. INTENSITY CONSTRAINT TYPE
(N=NONE, F=FIX%, L=LINEAR COMBINATIONS): F L CR
NUMBER OF LINEAR COMBINATIONS : 1 2 CR
LINEAR COMBINATION # 1      :  0.0000      0.0000      0.0000
                               0.0000
LINEAR COMBINATION # 2      :  0.0000      0.0000      0.0000
                               0.0000

LINEAR COMBINATION # 1:
COEFFICIENT # 1 (  0.1500 NS):  0.0000 .5 CR
COEFFICIENT # 2 (  0.2500 NS):  0.0000 -.5 CR
COEFFICIENT # 3 (  0.1800 NS):  0.0000 CR
COEFFICIENT # 4 (  2.0000 NS):  0.0000 CR

LINEAR COMBINATION # 2:
COEFFICIENT # 1 (  0.1500 NS):  0.0000 CR
COEFFICIENT # 2 (  0.2500 NS):  0.0000 CR
COEFFICIENT # 3 (  0.1800 NS):  0.0000 1 CR
COEFFICIENT # 4 (  2.0000 NS):  0.0000 -100 CR

```

Two linear combinations have been chosen. For each linear combination you have to enter the coefficients by which the intensities shall be multiplied. In the first linear combination the coefficients entered are 0.5, -0.5, 0.0 and 0.0. This means that $0.5I_1 - 0.5I_2 + 0.0I_3 + 0.0I_4 = 0$ in the fitting, i.e. $I_1 = I_2$. The second linear combination puts $I_4 = I_3/100$.

Finally, the Menu summary is displayed:

```

SUMMARY FOR THIS MENU:
1. NUMBER OF LIFETIMES      :      4
2. LIFETIMES (NS)          :  0.1500G  0.2500G  0.1800F
                               2.0000F
3. INTENSITY CONSTRAINT TYPE : L
NUMBER OF LINEAR COMBINATIONS : 2
LINEAR COMBINATION # 1      :  0.5000  -0.5000  0.0000
                               0.0000
LINEAR COMBINATION # 2      :  0.0000  0.0000  1.0000
                               -100.0000

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF POSEDIT MENU # 6: LIFETIMES AND INTENSITY CONSTRAINTS

```

Normally, the background is a free parameter in POSITRONFIT. However, two different constraints may be imposed on the background:

```

POSEDIT SELECTION IS # 7: BACKGROUND CONSTRAINTS
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

```

Menu 7: Background

```

LOCAL MENU # 7: BACKGROUND CONSTRAINTS
1. BACKGROUND CONSTRAINT (N=NONE, S=SPECTRUM MEAN, I=INPUT): N 1 CR
BACKGROUND FIXED TO INPUT VALUE : 0.00000E+00 3.59 CR

```

```

SUMMARY FOR THIS MENU:
1. BACKGROUND CONSTRAINT (N=NONE, S=SPECTRUM MEAN, I=INPUT): I
BACKGROUND FIXED TO INPUT VALUE 3.59000E+00

```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): *1 CR

If “N” is entered after the first question, no constraint is imposed on the background. But if you want to fix the background to a certain value (3.59 above) an “I” has to be entered and the fixed value typed in on the next line, as in the example above. If you would rather fix the background to the mean value over a number of channels of the measured background, an “S” should be entered; in the following two lines, you then enter the lower and upper channel number between which the averaging shall take place:

```
1. BACKGROUND CONSTRAINT (N=NONE, S=SPECTRUM MEAN, I=INPUT): I s CR
BACKGROUND FIXED TO MEAN BETWEEN TWO CHANNELS
LOWER CHANNEL LIMIT      :    1 400 CR
UPPER CHANNEL LIMIT      :   512 500 CR
```

SUMMARY FOR THIS MENU:

```
1. BACKGROUND CONSTRAINT (N=NONE, S=SPECTRUM MEAN, I=INPUT): S
BACKGROUND FIXED TO MEAN BETWEEN TWO CHANNELS
LOWER CHANNEL LIMIT :    400
UPPER CHANNEL LIMIT :    500
```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF POSEDIT MENU # 7: BACKGROUND CONSTRAINTS

Menu 8: Area

In a similar way as in Menu 7, Menu 8 is used if a constraint on the total area of the spectrum is wanted. Normally, the area is not constrained, but in some cases, where the analysis does not include the spectrum peak region, it may be useful to constrain the area to obtain a well-defined time-zero in the fitting. If no constraint is necessary an “N” is entered, but if an “S” or an “I” is typed, the area will be fixed to the area of the measured spectrum or to a value entered from the keyboard, respectively. In both cases the lower and upper channel limit between which the area is defined must also be entered. For example:

POSEDIT SELECTION IS # 8: AREA CONSTRAINTS
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 8: AREA CONSTRAINTS

```
1. AREA CONSTRAINT (N=NONE, S=SPECTRUM VALUE, I=INPUT): S CR
LOWER CHANNEL LIMIT      :    10 CR
UPPER CHANNEL LIMIT      :   510 CR
```

SUMMARY FOR THIS MENU:

```
1. AREA CONSTRAINT (N=NONE, S=SPECTRUM VALUE, I=INPUT): S
LOWER CHANNEL LIMIT :    10
UPPER CHANNEL LIMIT :   510
AREA FIXED TO SPECTRUM VALUE
```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF POSEDIT MENU # 8: AREA CONSTRAINTS

Menu 9: Source Correction

In the main part of Menu 9 the lifetime spectrum and total intensity of the so-called source correction is defined (due to positrons annihilating in the source material and at surfaces). First, the number of components in the source correction spectrum is entered:

POSEDIT SELECTION IS # 9: SOURCE CORRECTION
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 9: SOURCE CORRECTION
1. NUMBER OF CORRECTION TERMS : 2 CR

If an 0 (zero) is entered, the menu contains no more questions, and no source correction will be made. In the present case the second and third questions deal with the lifetimes and intensities of the source correction spectrum:

2. LIFETIMES (NS) : 0.1800 0.3560
LIFETIME FOR TERM # 1: 0.1800 CR
LIFETIME FOR TERM # 2: 0.3560 CR
3. RELATIVE INTENSITIES (%) : 60.2400 39.7600
INTENSITY OF TERM # 1: 60.2400 CR
INTENSITY OF TERM # 2: 39.7600 CR
4. TOTAL (%) : 8.3000 CR

The percentage in question 4 refers to the fraction of positions annihilating in the source and at surfaces. The source correction spectrum with a total area equal to this fraction of the measured spectrum will be subtracted from the measured spectrum before a second fitting cycle is carried out. (In fact, the total fraction is defined by the product of the percentage in question 4 and the sum of the relative intensities in question 3. The same correction will be made, if the intensities are entered as 5% and 3%, and the total as 100%. This may in some cases be more convenient.) It may sometimes be useful to change the number of lifetime components and the initial parameters before this 2nd cycle. Hence, a question No. 5 asks if you want to do this. If the answer is Yes, the program leads you through a submenu which is identical to Menu No. 6:

5. NEW 2ND CYCLE DATA? (Y,N) : Y CR

SUMMARY FOR THIS MENU:
1. NUMBER OF CORRECTION TERMS : 2
2. LIFETIMES (NS) : 0.1800 0.3560
3. RELATIVE INTENSITIES (%) : 60.2400 39.7600
4. TOTAL (%) : 8.3000
5. NEW 2ND CYCLE DATA? (Y,N) : Y

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

YOU NOW ENTER 2ND CYCLE SUBMENU

1. NUMBER OF LIFETIMES : 4 3 CR
2. LIFETIMES (NS) : 0.1500G 0.2500G 0.4500G
LIFETIME # 1: 0.1500G CR
LIFETIME # 2: 0.2500G CR
LIFETIME # 3: 0.4500G 2.f CR
3. INTENSITY CONSTRAINT TYPE
(N=NONE, F=FIX%, L=LINEAR COMBINATIONS): N CR

SUMMARY FOR THIS SUBMENU:
1. NUMBER OF LIFETIMES : 3
2. LIFETIMES (NS) : 0.1500G 0.2500G 2.0000F
3. INTENSITY CONSTRAINT TYPE : N

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

```
END OF 2ND CYCLE SUBMENU
END OF POSEDIT MENU # 9: SOURCE CORRECTION
```

Menu 10: Export Control Data

Finally, Menu 10 is reached. This menu was described in detail above in the general description of the -EDIT programs and shall therefore not be discussed further here:

```
POSEDIT SELECTION IS # 10: EXPORT POSITRONFIT CONTROL DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 10: EXPORT POSITRONFIT CONTROL DATA
1. LIST CONTROL DATA (Y,N)      : N CR
2. SAVE DATA IN FILE WITH NAME: NIELS.POS sample.pos CR
```

```
SUMMARY FOR THIS MENU:
1. LIST CONTROL DATA (Y,N)      : N
2. SAVE DATA IN FILE WITH NAME: SAMPLE.POS
```

```
** CONTROL DATA WAS WRITTEN TO A NEW FILE **
```

```
CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
```

```
=====TERMINAL POINT FOR POSEDIT MAIN MENU=====
KEY <*> OR <*> TO EXIT FROM EDITOR, <RETURN> TO RECYCLE THE MENU: * CR
```

Control File Listing

POSEDIT has now generated a Control File called SAMPLE.POS. For comparison with the data entered above you may wish to consult the printout of the generated Control File given in Appendix C.1. This is the type of listing you obtain if you answer Yes to the first question in Menu 10 (or 2nd question in Menu 1). Appendix C.1 also contains a detailed discussion of the structure of the Control Data set. The Main Output from a POSITRONFIT run with SAMPLE.POS as input is shown in Section 2.1.

3.3 RESEDIT

Introduction

In this Section we describe in some detail the interactive program RESEDIT. The purpose of the program is to generate a Control File that can act as input to RESOLUTION. The functioning of RESEDIT is therefore strongly correlated to the input options in RESOLUTION. Because some features are common for POSITRONFIT and RESOLUTION, some Menus in RESEDIT are identical or very similar to some in POSEDIT.

Main Menu

When you start the program, first a heading is displayed and the Main Menu follows:

```
RESEDIT MAIN MENU:
1. IMPORT RESOLUTION CONTROL DATA
2. OUTPUT OPTIONS
3. SPECTRUM
4. AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
5. RESOLUTION FUNCTION
6. LIFETIMES AND INTENSITY CONSTRAINTS
```

- 7. BACKGROUND RANGE
- 8. EXPORT RESOLUTION CONTROL DATA

Each of the 8 Local Menus will be discussed below.

The functioning of the first Menu is the same as in POSEDIT. After the Menu has been selected by typing CR, the following can be seen:

**Menu 1:
Import
Control Data**

```
RESEDIT SELECTION IS # 1: IMPORT RESOLUTION CONTROL DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 1: IMPORT RESOLUTION CONTROL DATA
CONTROL DATA ORIGINATES FROM FILE: PETER.RES
1. LOAD CONTROL DATA FROM A FILE (Y,N): N CR
```

In order that you may avoid generating an entirely new Control Data Set each time you run RESEDIT, the program has saved (in RESFILE) the Data Set from the previous run to be used as default values on restart. The name of the Control File into which the Data Set was written is also displayed (i.e. in this case PETER.RES). However, it is optional to choose another Control File as default by entering Yes to the first question above (see POSEDIT example). In the present example, though, the No reply was accepted, and no new Control Data were read.

If you want to check the contents of the imported Control Data Set, a Yes should be entered after the next question. In this example a No was chosen (for a listing of a Control Data Set; see Appendix C.2):

```
2. LIST CURRENT CONTROL DATA SET (Y,N): N CR
```

```
SUMMARY FOR THIS MENU:
```

- 1. LOAD CONTROL DATA FROM A FILE (Y,N): N
- 2. LIST CURRENT CONTROL DATA SET (Y,N): N

```
CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF RESEDIT MENU # 1: IMPORT RESOLUTION CONTROL DATA
```

The summary concludes Menu 1.

Local Menu No. 2 deals with the output options (for RESOLUTION). It is identical to Menu 2 in POSEDIT and ACAREEDIT. The standard output from RESOLUTION is the “Main Output” which contains all input and fitted parameters, etc. (see Section 2.2). Additionally, it is optional to have a listing of the whole Control Data Set as read by RESOLUTION (“Input Echo”) and a printout of the fitting parameters after each iteration during the fitting process (“Iteration Printout”). Further, a graphical representation on the lineprinter of the deviations between the fitted and the measured spectrum (“Residual Plot”), and a printout of the matrix of total correlations between the fitted parameters (“Correlation Matrix”) are possible. Finally, you can choose to save in a “Communication File” (RESCOM) a data set that contains the measured spectrum, the Main Output, and the Correlation Matrix for graphic display or other purposes (“Communication Data Set”). If all these options are chosen, Menu 2 may look like this:

**Menu 2:
Output Options**

RESEDIT SELECTION IS # 2: OUTPUT OPTIONS
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 2: OUTPUT OPTIONS
1. INPUT ECHO ? (Y,N) : N y CR
2. ITERATION PRINTOUT ? (Y,N) : N y CR
3. RESIDUAL PLOT ? (Y,N) : Y CR
4. CORRELATION MATRIX ? (Y,N) : Y CR
5. COMMUNICATION DATA ? (Y,N) : N y CR

SUMMARY FOR THIS MENU:
1. INPUT ECHO ? (Y,N) : Y
2. ITERATION PRINTOUT ? (Y,N) : Y
3. RESIDUAL PLOT ? (Y,N) : Y
4. CORRELATION MATRIX ? (Y,N) : Y
5. COMMUNICATION DATA ? (Y,N) : Y

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF RESEDIT MENU # 2: OUTPUT OPTIONS

Menu 3: Spectrum

Local Menu No. 3 defines the measured spectrum to be analysed, its number of channels and format. This Menu is identical to Menu No. 3 in POSEDIT.

First, you enter the number of channels in the spectrum, and secondly, the proper FORTRAN format for reading the spectrum (notice that I-formatting is not allowed.) Then, you must enter the name of the file that contains the spectrum and the label by which the spectrum is identified. In the present example, the spectrum file is located on the diskette in drive B. Finally, you have to decide whether or not the measured spectrum shall be included in the Control Data Set (and the Communication Data Set, if applicable) and enter this:

RESEDIT SELECTION IS # 3: SPECTRUM
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 3: SPECTRUM
1. CHANNELS IN SPECTRUM : 512 CR
2. SPECTRUM FORMAT : (F7.0/(10F7.0)) CR
3. SPECTRUM FILE, NAME IS : B:GR35050.DAT CR
4. SPECTRUM LABEL : 35090 CR
5. SPECTRUM TO BE INCORPORATED IN CONTROL DATA ? (Y,N): Y CR

The label can be any text string of at most 32 characters (including a number, as above). In the present example the spectrum with No. 35090 shall be analysed again. If another spectrum had been chosen, the program would have been tried to locate that spectrum by scanning through the Spectrum File in order to find a headline, the beginning of which matches the entered spectrum label. For a successful match, the spectrum label must coincide, up to its last nonblank character, with the first part of a spectrum headline (see POSEDIT for an example). In the present example, no spectrum was read:

SUMMARY FOR THIS MENU:
1. CHANNELS IN SPECTRUM : 512
2. SPECTRUM READING FORMAT : (F7.0/(10F7.0))

```
3. SPECTRUM FILE, NAME IS      : B:GR35050.DAT
4. SPECTRUM LABEL              : 35090
5. SPECTRUM TO BE INCORPORATED IN CONTROL DATA ? (Y,N): Y
```

```
ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
NO READING OF SPECTRUM FROM FILE
END OF RESEDIT MENU # 3: SPECTRUM
```

Local Menu 4 is very similar to Menu 4 in POSEDIT. The two first questions concern the range of channels to be included in calculations of the spectrum area:

Menu 4: Ranges

```
RESEDIT SELECTION IS # 4: AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 4: AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
1. AREA RANGE STARTS IN CH. # :      3 CR
2. AREA RANGE ENDS   IN CH. # :    511 CR
```

By defining the area range you can exclude channels with special contents (e.g. identification number or measuring time).

Normally, fewer channels are included in the fitting than in the area calculation. Hence, a fit range is defined next. First, though, the peak channel of the spectrum is found and displayed to give a clear indication of the position of the spectrum:

```
SPECTRUM PEAK IS IN CHANNEL:    109
3. FIT RANGE STARTS IN CH. # :   98 95 CR
4. FIT RANGE ENDS   IN CH. # :  250 CR
```

Finally, you shall enter the time equivalent to the width of one channel and the guessed channel number corresponding to zero time:

Time scale Time-zero

```
5. TIME SCALE (NS/CHANNEL)      :    0.0311 CR
6. TIME-ZERO (CHANNEL NUMBER)   :  107.0000 CR
```

```
SUMMARY FOR THIS MENU:
1. AREA RANGE STARTS IN CH. # :      3
2. AREA RANGE ENDS   IN CH. # :    511
3. FIT RANGE STARTS IN CH. # :     95
4. FIT RANGE ENDS   IN CH. # :    250
5. TIME SCALE (NS/CHANNEL)      :    0.0311
6. TIME-ZERO (CHANNEL NUMBER)   :  107.0000
```

```
ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF RESEDIT MENU # 4: AREA AND FIT RANGES. TIME SCALE. TIME-ZERO.
```

Of the parameters entered in Menu 4, only time-zero is a fitting parameter (and cannot be fixed).

The main purpose of the RESOLUTION program is to determine the resolution function of the lifetime spectrometer. This function is described as a sum of (maximum six) Gaussians, the centroids of which may be shifted with respect to each other. The widths (defined by the Full Width at Half Maximum) and the shifts are fitting parameters, while the intensities of the Gaussians are always fixed.

Menu 5: Resolution Function

The structure of this Menu is the same as in POSEDIT. After defining the number of Gaussians, you enter the FWHM of each Gaussian (guessed or fixed), their relative intensities, and finally the shift of each Gaussian (guessed or fixed). Since the shifts are relative to time-zero (which is also a fitting parameter), not all the shifts can be free during the fitting. At least one shift has to be fixed. (If all three shifts end with a G, a warning will be displayed and you are returned to question 4 as shown below).

RESEDIT SELECTION IS # 5: RESOLUTION FUNCTION
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

```
LOCAL MENU # 5: RESOLUTION FUNCTION
1. NUMBER OF GAUSSIANS      :      3 CR
2. FWHM (NS)                :      0.2000G      0.2000G      0.2200G
   FWHM (NS) # 1:          0.2000G CR
   FWHM (NS) # 2:          0.2000G .3 CR
   FWHM (NS) # 3:          0.2200G CR
3. INTENSITY (%)            :      12.0000      13.0000      75.0000
   INTENSITY (%) # 1:       12.0000 CR
   INTENSITY (%) # 2:       13.0000 CR
   INTENSITY (%) # 3:       75.0000 CR
4. SHIFT (NS)               :      0.1000G     -0.0500F      0.0000F
   SHIFT (NS) # 1:          0.1000G CR
   SHIFT (NS) # 2:         -0.0500F G CR
   SHIFT (NS) # 3:          0.0000F G CR
```

```
SUMMARY FOR THIS MENU:
1. NUMBER OF GAUSSIANS      :      3
2. FWHM (NS)                :      0.2000G      0.3000G      0.2200G
3. INTENSITY (%)            :      12.0000      13.0000      75.0000
4. SHIFT (NS)               :      0.1000G     -0.0500G      0.0000G
```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

AT LEAST ONE SHIFT MUST BE FIXED

```
4. SHIFT (NS)               :      0.1000G     -0.0500G      0.0000G
   SHIFT (NS) # 1:          0.1000G CR
   SHIFT (NS) # 2:         -0.0500G CR
   SHIFT (NS) # 3:          0.0000G f CR
```

```
SUMMARY FOR THIS MENU:
1. NUMBER OF GAUSSIANS      :      3
2. FWHM (NS)                :      0.2000G      0.3000G      0.2200G
3. INTENSITY (%)            :      12.0000      13.0000      75.0000
4. SHIFT (NS)               :      0.1000G     -0.0500G      0.0000F
```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF RESEDIT MENU # 5: RESOLUTION FUNCTION

**Menu 6:
Lifetime
Components.
Constraints.**

The 6th Local Menu is identical to the one in POSEDIT. It defines the number of lifetime components in the lifetime spectrum and their initial (guessed or fixed) values, as well as possible constraints on the relative intensities. The maximum number of lifetime components is six:

RESEDIT SELECTION IS # 6: LIFETIMES AND INTENSITY CONSTRAINTS
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

```
LOCAL MENU # 6: LIFETIMES AND INTENSITY CONSTRAINTS
1. NUMBER OF LIFETIMES      :      3 4 CR
2. LIFETIMES (NS)          :      0.1100G      0.3000G      0.4500G
                           :      1.0000F
```

```

LIFETIME # 1: 0.1100G CR
LIFETIME # 2: 0.3000G .45 CR
LIFETIME # 3: 0.4500G .18f CR
LIFETIME # 4: 1.0000F 2f CR

```

The 3rd question deals with possible intensity constraints. In this example the default value of no constraints is changed to an F and fixed values are imposed on all the intensities:

```

3. INTENSITY CONSTRAINT TYPE
(N=NONE, F=FIX%, L=LINEAR COMBINATIONS): N f CR
INTENSITIES : VARIABLE VARIABLE VARIABLE
              VARIABLE
INTENSITY # 1 ( 0.1100 NS): VARIABLE 34 CR
INTENSITY # 2 ( 0.4500 NS): VARIABLE 3.5 CR
INTENSITY # 3 ( 0.1800 NS): VARIABLE 5 CR
INTENSITY # 4 ( 2.0000 NS): VARIABLE .1 CR

```

These are too many constraints and if you accept the Menu summary, you are returned to this question:

```

SUMMARY FOR THIS MENU:
1. NUMBER OF LIFETIMES : 4
2. LIFETIMES (NS) : 0.1100G 0.4500G 0.1800F
                   2.0000F
3. INTENSITY CONSTRAINT TYPE : F
   INTENSITIES : 34.0000 3.5000 5.0000
                0.1000

```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

NOT ALL INTENSITIES CAN BE FIXED

```

3. INTENSITY CONSTRAINT TYPE
(N=NONE, F=FIX%, L=LINEAR COMBINATIONS): F CR
INTENSITIES : 34.0000 3.5000 5.0000
              0.1000
INTENSITY # 1 ( 0.1100 NS): 34.0000 v CR
INTENSITY # 2 ( 0.4500 NS): 3.5000 v CR
INTENSITY # 3 ( 0.1800 NS): 5.0000 CR
INTENSITY # 4 ( 2.0000 NS): 0.1000 v CR

```

```

SUMMARY FOR THIS MENU:
1. NUMBER OF LIFETIMES : 4
2. LIFETIMES (NS) : 0.1100G 0.4500G 0.1800F
                   2.0000F
3. INTENSITY CONSTRAINT TYPE : F
   INTENSITIES : VARIABLE VARIABLE 5.0000
                VARIABLE

```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

END OF RESEDIT MENU # 6: LIFETIMES AND INTENSITY CONSTRAINTS

In RESOLUTION the background is always fixed to the mean value between two channels. Hence, in the Menu only these limits shall be specified. We use this also to illustrate one of the check features built into the -EDIT programs:

Menu 7: Background

```

RESEDIT SELECTION IS # 7: BACKGROUND RANGE
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

```

```
LOCAL MENU # 7: BACKGROUND RANGE
1. LOWER CHANNEL LIMIT      : 411 CR
2. UPPER CHANNEL LIMIT      : 511 311 CR
```

```
SUMMARY FOR THIS MENU:
1. LOWER CHANNEL LIMIT      : 411
2. UPPER CHANNEL LIMIT      : 311
```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

If you accept the “upper limit” to be below the “lower limit” you will be returned to the questions again:

```
INVALID CHANNEL LIMITS FOR BACKGROUND
1. LOWER CHANNEL LIMIT      : 411 CR
2. UPPER CHANNEL LIMIT      : 311 513 CR
```

```
SUMMARY FOR THIS MENU:
1. LOWER CHANNEL LIMIT      : 411
2. UPPER CHANNEL LIMIT      : 513
```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF RESEDIT MENU # 7: BACKGROUND RANGE

Notice that the upper limit now exceeds the number of channels specified in Menu 3. However, checks of inconsistencies between the data from the various Menus are not carried out until in the last Menu where the Control Data are “exported” (since the data may be entered in any sequence of menus):

RESEDIT SELECTION IS # 8: EXPORT RESOLUTION CONTROL DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

```
LOCAL MENU # 8: EXPORT RESOLUTION CONTROL DATA
1. LIST CONTROL DATA (Y,N) : N CR
```

```
*** WARNING ***: CHANNEL LIMIT FOR BACKGROUND CONSTRAINT
REDUCED FROM 513 TO 512
```

```
2. SAVE DATA IN FILE WITH NAME: PETER.RES sample.res CR
FILE EXISTS: SAMPLE.RES
(A=APPEND, O=OVERWRITE, Q=QUIT): A CR
```

```
SUMMARY FOR THIS MENU:
1. LIST CONTROL DATA (Y,N) : N
2. SAVE DATA IN FILE WITH NAME: SAMPLE.RES
```

** CONTROL DATA WAS APPENDED TO A FILE **

CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

```
=====TERMINAL POINT FOR RESEDIT MAIN MENU=====
KEY <*> OR <E> TO EXIT FROM EDITOR, <RETURN> TO RECYCLE THE MENU: * CR
```

Since this Menu was discussed in detail in Section 3.1, we shall not discuss it further here.

RESEDIT has now generated a Control Data Set and appended it to a Control File, called SAMPLE.RES. For comparison with the data entered above you may wish to consult the printout given in Appendix C.2. This is the type of listing you obtain if you answer Yes to the first question

Menu 8: Export Control Data

Control File Listing

in Menu 8 (or 2nd question in Menu 1). Appendix C.2 also contains a detailed discussion of the structure of the Control Data Set. The Main Output from a RESOLUTION run with the Control Data Set generated above as input is shown in Section 2.2.

3.4 ACAREEDIT

In this Section we describe in some detail the interactive program ACAREEDIT. The purpose of the program is to generate a Control File that can act as input data to ACARFIT. The functioning of ACAREEDIT is therefore strongly correlated to the input options in ACARFIT.

The structure of ACAREEDIT is analogous to POSEDIT and RESEDIT, but several of the Menus are very different in detail.

When you start the program, a heading is displayed first and the Main Menu follows:

```
ACAREEDIT MAIN MENU:
  1. IMPORT ACARFIT CONTROL DATA
  2. OUTPUT OPTIONS
  3. SPECTRUM, ANGLE-ZERO, ANGLE SCALE.
  4. FIT AND NORMALIZATION RANGES
  5. COMPONENTS AND CONSTRAINTS
  6. BACKGROUND
  7. EXPORT ACARFIT CONTROL DATA
```

Each of the 7 Local Menus will be discussed below.

After the Main Menu, Local Menu No. 1 is automatically selected. If accepted, by a CR, the following is displayed:

```
ACAREEDIT SELECTION IS # 1: IMPORT ACARFIT CONTROL DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 1: IMPORT ACARFIT CONTROL DATA
CONTROL DATA ORIGINATES FROM FILE: CUME1.ACA
1. LOAD CONTROL DATA FROM A FILE (Y,N): N CR
```

In order to enable you to avoid generating an entirely new Control Data Set each time you run ACAREEDIT, the Data Set from the previous run has been saved (in ACARFILE) to be used as default values on restart. The name of the Control File into which this previous Data Set was written is displayed (in this case CUME1.ACA). It is optional to choose another Control File as default by answering Yes to the first question above. In the present example, though, the No was accepted, and no new Control Data were read.

If you want a display of the content of the “imported” Control Data Set, a Yes should be entered after the next question. In this example the No was accepted (for a listing of a Control Data Set, see Appendix C.3):

```
2. LIST CURRENT CONTROL DATA SET (Y,N): N CR
```

Introduction

Main Menu

Menu 1: Import Control Data

```

SUMMARY FOR THIS MENU:
1. LOAD CONTROL DATA FROM A FILE (Y,N): N
2. LIST CURRENT CONTROL DATA SET (Y,N): N

CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): *1 CR

```

However, the Menu summary was not accepted in this example, in order to illustrate how to change the input to a Menu. By entering *1 the first question is repeated, and a Yes is entered:

```

1. LOAD CONTROL DATA FROM A FILE (Y,N): N y CR
NAME OF INPUT CONTROL DATA FILE: CUME1.ACA MORTEN.ACA CR
READ DATA SET NUMBER:      2 1 CR

```

A new Control File (MORTEN.ACA) has now been chosen; the next line asks which one of the Control Data Sets in the file you want to import. Having chosen the first one, you continue with the rest of the Menu:

```

2. LIST CURRENT CONTROL DATA SET (Y,N): N CR

```

```

SUMMARY FOR THIS MENU:
1. LOAD CONTROL DATA FROM A FILE (Y,N): Y
2. LIST CURRENT CONTROL DATA SET (Y,N): N

```

```

CONTROL DATA WAS LOADED FROM A FILE

```

```

CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF ACAREREDIT MENU # 1: IMPORT ACARFIT CONTROL DATA

```

Menu 2: Output Options

Local Menu No. 2 deals with the output options (for ACARFIT). It is identical with Menu 2 in POSEDIT and RESEDIT.

The standard output from ACARFIT is the “Main Output” which contains all input and fitted parameters etc. (see Section 2.3).

Additionally, it is optional to have a listing of the whole Control Data Set as read by ACARFIT (“Input Echo”) together with a printout of the fitting parameters after each iteration during the fitting process (“Iteration Printout”). Further, a graphical representation on the lineprinter of the deviations between the fitted and the measured spectrum (“Residual Plot”), and a printout of the matrix of total correlations between the fitted parameters (“Correlation Matrix”) are possible. Finally, you can choose to save in a “Communication File” (ACARCOM) a data set that contains the measured spectrum (optional), the Main Output, and the Correlation Matrix for graphic display or other purposes (“Communication Data Set”). If all the options are chosen, Menu 2 may look like this:

```

ACAREREDIT SELECTION IS # 2: OUTPUT OPTIONS
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 2: OUTPUT OPTIONS
1. INPUT ECHO ? (Y,N)      : N y CR
2. ITERATION PRINTOUT ? (Y,N) : Y CR
3. RESIDUAL PLOT ? (Y,N)   : N y CR
4. CORRELATION MATRIX ? (Y,N) : Y CR
5. COMMUNICATION DATA ? (Y,N) : N y CR

```

SUMMARY FOR THIS MENU:

1. INPUT ECHO ? (Y,N) : Y
2. ITERATION PRINTOUT ? (Y,N) : Y
3. RESIDUAL PLOT ? (Y,N) : Y
4. CORRELATION MATRIX ? (Y,N) : Y
5. COMMUNICATION DATA ? (Y,N) : Y

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF ACAREEDIT MENU # 2: OUTPUT OPTIONS

The main function of the third Local Menu is to define the measured spectrum to be analysed. For the analysis of a spectrum, the number of data-points must be specified. Furthermore, the angle and the standard deviation on the number of counts must be specified for each data point. To admit a high degree of flexibility, it is optional whether this information shall be entered from the Spectrum File as part of the Spectrum Data Set (see Section 3.1) or be defined in a different way. This is described further below.

Menu 3: Spectrum

The first item of Menu 3 deals with the angles associated with the data points:

ACAREEDIT SELECTION IS # 3: SPECTRUM, ANGLE-ZERO, ANGLE SCALE.
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 3: SPECTRUM, ANGLE-ZERO, ANGLE SCALE.

1. LOAD ANGLE TABLE FROM SPECTRUM FILE ? (Y,N) : Y CR
FORMAT FOR ANGLES : (12F6.2) CR

If the Spectrum Data Set to be loaded from the Spectrum File contains a table of the angles (which is necessary if the data points do not have equidistant separation), a Yes shall be entered (or accepted as above) after the first question, and a format for the table defined. Similarly, if a table of the uncertainties shall be loaded (necessary if the ACAR spectrum has been normalized) the answer to the next question shall be a Yes and a format for the table entered.

2. LOAD UNCERTAINTY TABLE FROM SPECTRUM FILE ? (Y,N) : Y CR
FORMAT FOR UNCERTAINTIES : (12F6.0) CR

The next four questions concern the spectrum itself. First, is the number of data points in the spectrum part of the Spectrum Data Set, or shall it be entered from the keyboard? In the present example, the latter is the case:

3. LOAD NUMBER OF DATA POINTS FROM SPECTRUM FILE ? (Y,N) : N CR
NUMBER OF DATA POINTS : 65 55 CR

The FORTRAN format for properly reading the spectrum table follows next (notice that I-formatting is not allowed), and afterward the name of the file that contains the spectrum. The present example illustrates a case where the file is to be found on the diskette in drive A:

4. FORMAT FOR SPECTRUM : (12F6.0) CR

5. SPECTRUM FILE, NAME IS : AC32200.DAT A:01591.dat CR

In most cases, of course, when the first 4 questions have been answered once, the input shall not be changed again (except possibly on occasion the number of data points).

The next question deals with the label by which the spectrum is identified. For a successful match the spectrum label must coincide, up to its last nonblank character, with the first part of a spectrum headline. In this example the label is a number, but it may be any string of max. 32 characters:

6. SPECTRUM LABEL : 32291 01591 CR

Angle-zero

In the following line the (guessed or fixed) initial value of angle-zero shall be entered. If the Spectrum Data Set contains an angle table the angle-zero is entered in mrad and with reference to the angular axis defined by this table (as in the present example). If the Spectrum Data Set contains no angle table (and the data points therefore are considered to be in equidistant channels), the unit for angle-zero is "channels". In that case an additional number must be entered, viz. the separation of channels in mrad units (not shown here). Finally, you have to choose whether to incorporate the Spectrum Data Set (including, if applicable, angle- and uncertainty table) in your Control Data Set, or not. If you do not incorporate it (them), it (they) will not appear in the Communication File either. This latter option can be chosen to save space, but requires that there is access to the Spectrum Data Set, when you run ACARFIT:

Angle scale

7. ANGLE-ZERO (MRAD) : 0.0000G CR
8. SPECTRUM TO BE INCORPORATED IN CONTROL DATA ? (Y,N) : Y CR

After the menu summary, the program will try to locate the Spectrum Data Set by scanning through the Spectrum File in order to find a headline, the beginning of which matches the spectrum label. If the spectrum is found it will be read and its headline displayed. (If it is not found, a message will appear):

SUMMARY FOR THIS MENU:
1. LOAD ANGLE TABLE FROM SPECTRUM FILE ? (Y,N) : Y
FORMAT FOR ANGLES : (12F6.2)
2. LOAD UNCERTAINTY TABLE FROM SPECTRUM FILE ? (Y,N) : Y
FORMAT FOR UNCERTAINTIES : (12F6.0)
3. LOAD NUMBER OF DATA POINTS FROM SPECTRUM FILE ? (Y,N) : N
NUMBER OF DATA POINTS : 55
4. FORMAT FOR SPECTRUM : (12F6.2)
5. SPECTRUM FILE, NAME IS : A:01591.DAT
6. SPECTRUM LABEL : 01591
7. ANGLE-ZERO (MRAD) : 0.0000G
8. SPECTRUM TO BE INCORPORATED IN CONTROL DATA ? (Y,N) : Y

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
READING OF SPECTRUM FROM FILE WAS REQUESTED

SPECTRUM LABEL LOCATED:
01591 COPPER,5N,CW 70%

06-AUG-86

END OF ACARETIT MENU # 3: SPECTRUM, ANGLE-ZERO, ANGLE SCALE.

In Local Menu 4, two ranges of angles are defined. First, you enter the range within which you want the fitting to take place. This range need not be symmetrical. (Since the start and end point of the fitting range have to coincide with data points, the requested range may be truncated to a smaller range. This is done by ACARFIT). Secondly, you have to enter a symmetrical normalization range. A peak count will be normalized to the area within this range:

Menu 4: Ranges

ACARETIT SELECTION IS # 4: FIT AND NORMALIZATION RANGES
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 4: FIT AND NORMALIZATION RANGES
1. REQUESTED FIT RANGE STARTS IN (MRAD): -15.0000 -17 CR
2. REQUESTED FIT RANGE ENDS IN (MRAD) : 15.0000 17 CR
3. NORMALIZATION RANGE (+-MRAD) : 13.0000 CR

SUMMARY FOR THIS MENU:
1. REQUESTED FIT RANGE STARTS IN (MRAD): -17.0000
2. REQUESTED FIT RANGE ENDS IN (MRAD) : 17.0000
3. NORMALIZATION RANGE (+-MRAD) : 13.0000

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF ACARETIT MENU # 4: FIT AND NORMALIZATION RANGES

Local Menu 5 defines the initial (fixed, "F", or guessed, "G") values of non-linear fitting parameters, i.e. the positions and widths of the Gaussians and parabolas. The constraints on the relative intensities are also defined in this menu. First, you enter the total number of components (i.e. Gaussians + parabolas):

Menu 5: Components and Constraints

ACARETIT SELECTION IS # 5: COMPONENTS AND CONSTRAINTS
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 5: COMPONENTS AND CONSTRAINTS

1. TOTAL NUMBER OF COMPONENTS: 4 6 CR

The maximum number of components is 15. The next question deals with the positions and widths of the components. The numbering of components is completely free, irrespective of their type. After a summary of the present values, the parameters for the first component are repeated, and a line offering seven different actions is displayed:

2. COMPONENTS:

#	TYPE	POSITION	FWHM	CUT-OFF	BROADENING	INTENSITY
1	PARA	0.0000F		5.5010F	2.5191F	25.0000
2	GAUS	0.0000F	16.0000G			VARIABLE
3	GAUS	0.0000F	4.7605F			10.0000
4	GAUS	0.0000F	8.0000G			VARIABLE
5	GAUS	0.0000F	1.0000F			VARIABLE
6	GAUS	0.0000F	1.0000F			VARIABLE

CURRENT COMPONENT IS # 1: PARA 0.0000F 5.5010F 2.5191F
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N CR

In the following we illustrate the functioning of the various actions. On accepting the “N” you jump to the next component:

```
CURRENT COMPONENT IS # 2: GAUS  0.0000F  16.0000G
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N e CR

COMPONENT READY FOR EDITING:

TYPE (G=GAUSS, P=PARABOLA) :    G CR
POSITION (+-MRAD)           :    0.0000F CR
FWHM (MRAD)                 :   16.0000G 16.3685f CR
```

If you want to change any of the parameters in the current component you strike “E” (as above) and carry out the editing. Subsequently, the parameters of the next component are displayed.

```
CURRENT COMPONENT IS # 3: GAUS  0.0000F  4.7605F
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): E A CR
```

If you wish to change parameters in another component than the current one, you enter “A” (as above) and the number of the component to be changed:

```
ENTER # OF THE COMPONENT TO BE EDITED:    3 6 CR

COMPONENT READY FOR EDITING:

TYPE (G=GAUSS, P=PARABOLA) :    G P CR
POSITION (+-MRAD)           :    0.0000F 6 CR
CUT-OFF (MRAD)              :    1.0000F 2 CR
BROADENING (MRAD)           :    1.0000F 3 CR
```

```
3. INTENSITY CONSTRAINT TYPE
(N=NONE, F=FIX%, L=LINEAR COMBINATIONS): F *2 CR
```

After editing component 6, the program continues to the third question (intensity constraints). In order to illustrate the other possible actions on the components, we return, however, to the first component by entering *2 as shown above. The display then shows:

```
2. COMPONENTS:

# TYPE POSITION FWHM CUT-OFF BROADENING INTENSITY
1 PARA 0.0000F 5.5010F 2.5191F 25.0000
2 GAUS 0.0000F 16.3685F VARIABLE
3 GAUS 0.0000F 4.7605F 10.0000
4 GAUS 0.0000F 8.0000G VARIABLE
5 GAUS 0.0000F 1.0000F VARIABLE
6 PARA 6.0000G 2.0000G 3.0000G VARIABLE
```

```
CURRENT COMPONENT IS # 1: PARA  0.0000F  5.5010F  2.5191F
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N d CR
```

If you enter “D”, as above, you may delete one of the components; in the following example we choose component number 5. A renumbering takes place, and the parameters of the new 5th component are shown:

```
ENTER # OF THE COMPONENT TO BE DELETED:    1 5 CR
COMPONENT # 5 WAS DELETED
```

CURRENT COMPONENT IS # 5: PARA 6.0000G 2.0000G 3.0000G
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N m CR

On entering "M" a map (= summary) of the parameter values is displayed (which confirms that a component has been deleted):

#	TYPE	POSITION	FWHM	CUT-OFF	BROADENING	INTENSITY
1	PARA	0.0000F		5.5010F	2.5191F	25.0000
2	GAUS	0.0000F	16.3685F			VARIABLE
3	GAUS	0.0000F	4.7605F			10.0000
4	GAUS	0.0000F	8.0000G			VARIABLE
5	PARA	6.0000G		2.0000G	3.0000G	VARIABLE

CURRENT COMPONENT IS # 5: PARA 6.0000G 2.0000G 3.0000G
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N i CR

Finally, the "I" (insert) action that makes it possible to insert new components is shown. After entering "I" (above) the next question deals with the number of the new component in the sequence of components:

ENTER # OF THE NEW COMPONENT AFTER INSERTION: 5 2 CR

Subsequently, the parameter values for the new component shall be entered:

INSERTED COMPONENT WITH DEFAULT VALUES READY FOR EDITING:

TYPE (G=GAUSS, P=PARABOLA) : G CR
POSITION (+-MRAD) : 0.0000F CR
FWHM (MRAD) : 1.0000F 2 CR

A map ("M") shows that the new component has in fact been inserted as number 2 and the other components renumbered:

CURRENT COMPONENT IS # 3: GAUS 0.0000F 16.3685F
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): I m CR

#	TYPE	POSITION	FWHM	CUT-OFF	BROADENING	INTENSITY
1	PARA	0.0000F		5.5010F	2.5191F	25.0000
2	GAUS	0.0000F	2.0000G			VARIABLE
3	GAUS	0.0000F	16.3685F			VARIABLE
4	GAUS	0.0000F	4.7605F			10.0000
5	GAUS	0.0000F	8.0000G			VARIABLE
6	PARA	6.0000G		2.0000G	3.0000G	VARIABLE

CURRENT COMPONENT IS # 3: GAUS 0.0000F 16.3685F
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): I q CR

The effect of the "Q" (quit) action is to jump out of part "2.COMPO- NENTS" to the next part:

3. INTENSITY CONSTRAINT TYPE
(N=NONE, F=FIX%, L=LINEAR COMBINATIONS): F CR

If you choose "N", no constraints are imposed on the intensities. By accepting an "F", you indicate that one or more of the intensities shall have a fixed value, and after a summary you have to enter this (these) value(s). For intensities that shall no longer be fixed a "V" (for variable) shall be

entered. An example would be:

#	TYPE	POSITION	FWHM	CUT-OFF	BROADENING	INTENSITY
1	PARA	0.0000F		5.5010F	2.5191F	25.0000
2	GAUS	0.0000F	2.0000G			VARIABLE
3	GAUS	0.0000F	16.3685F			VARIABLE
4	GAUS	0.0000F	4.7605F			10.0000
5	GAUS	0.0000F	8.0000G			VARIABLE
6	PARA	6.0000G		2.0000G	3.0000G	VARIABLE

INTENSITY # 1 (PARA): 25.0000 CR
 INTENSITY # 2 (GAUS): VARIABLE CR
 INTENSITY # 3 (GAUS): VARIABLE CR
 INTENSITY # 4 (GAUS): 10.0000 v CR
 INTENSITY # 5 (GAUS): VARIABLE 10 CR
 INTENSITY # 6 (PARA): VARIABLE CR

After the last CR a summary of the Menu is displayed. However, instead of accepting it we return to part 3 (by “*3”) in order to illustrate also the other type of intensity constraint. In this constraint a linear combination of the intensities is put equal to zero. This type of constraint is often particularly useful in the analysis of ACAR curves:

SUMMARY FOR THIS MENU:

1. TOTAL NUMBER OF COMPONENTS : 6
 2. COMPONENTS :

#	TYPE	POSITION	FWHM	CUT-OFF	BROADENING	INTENSITY
1	PARA	0.0000F		5.5010F	2.5191F	25.0000
2	GAUS	0.0000F	2.0000G			VARIABLE
3	GAUS	0.0000F	16.3685F			VARIABLE
4	GAUS	0.0000F	4.7605F			VARIABLE
5	GAUS	0.0000F	8.0000G			10.0000
6	PARA	6.0000G		2.0000G	3.0000G	VARIABLE

3. INTENSITY CONSTRAINT TYPE : F

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): *3 CR

3. INTENSITY CONSTRAINT TYPE
 (N=NONE, F=FIX%, L=LINEAR COMBINATIONS): F L CR

NUMBER OF LINEAR COMBINATIONS : 1 2 CR

LC#	COMP.#	COEFF.	COMP.#	COEFF.	COMP.#	COEFF.
1	1(PARA)	1.0000				
2	1(PARA)	1.0000				

LINEAR COMBINATION # 1:
 1(PARA) 1.0000

(N=NEXT, E=EDIT, A=ALTER LC#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N e CR

After a summary of the (in this case default) coefficients with which the intensities in the linear combination shall be multiplied, information is to be entered on which intensities are to be included in the constraints, and on the values of the coefficients:

LC CONSTRAINT READY FOR EDITING:

```

NUMBER OF NONZERO COEFFICIENTS : 1 2 CR
COEFFICIENT # 1
ATTACHED TO COMPONENT (PARA) # : 1 CR
VALUE : 1.0000 2.7286 CR
COEFFICIENT # 2
ATTACHED TO COMPONENT (GAUS) # : 2 4 CR
VALUE : 2.7286 27.4036 CR

LINEAR COMBINATION # 2:
1(PARA) 1.0000
(N=NEXT, E=EDIT, A=ALTER LC#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): E CR

```

LC CONSTRAINT READY FOR EDITING:

```

NUMBER OF NONZERO COEFFICIENTS : 1 2 CR
COEFFICIENT # 1
ATTACHED TO COMPONENT (PARA) # : 1 CR
VALUE : 1.0000 -75.3250 CR
COEFFICIENT # 2
ATTACHED TO COMPONENT (GAUS) # : 2 3 CR
VALUE : -75.3250 27.4036 CR

```

The first linear combination has the effect that $2.7286I_1 + 27.4036I_4 = 0$, i.e. that the intensity I_1 will be about 10 times higher than I_4 and of opposite sign. Similarly the second linear combination leads to an I_3 which is roughly three times higher than I_1 .

In order to facilitate the set-up of the (sometimes many) linear combinations with the sometimes many coefficients, the same seven actions are available as in part 2 (components).

Instead of accepting the summary, we return to part 2 to show what happens to the linear combinations if you delete a component:

SUMMARY FOR THIS MENU:

```

1. TOTAL NUMBER OF COMPONENTS : 6
2. COMPONENTS :
# TYPE POSITION FWHM CUT-OFF BROADENING INTENSITY
1 PARA 0.0000F 2.0000G 5.5010F 2.5191F LIN.COMB.
2 GAUS 0.0000F 16.3685F 27.4036 LIN.COMB.
3 GAUS 0.0000F 4.7605F 27.4036 LIN.COMB.
4 GAUS 0.0000F 8.0000G 2.0000G 3.0000G
5 GAUS 0.0000F 2.0000G 3.0000G
6 PARA 6.0000G

3. INTENSITY CONSTRAINT TYPE : L
LC# COMP.# COEFF. COMP.# COEFF. COMP.# COEFF.
1 1(PARA) 2.7286 4(GAUS) 27.4036
2 1(PARA) -75.3250 3(GAUS) 27.4036

```

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): *2 CR

```

2. COMPONENTS:
# TYPE POSITION FWHM CUT-OFF BROADENING INTENSITY
1 PARA 0.0000F 2.0000G 5.5010F 2.5191F LIN.COMB.
2 GAUS 0.0000F 16.3685F 27.4036 LIN.COMB.

```

```

3 GAUS 0.0000F 16.3685F          LIN.COMB.
4 GAUS 0.0000F 4.7605F          LIN.COMB.
5 GAUS 0.0000F 8.0000G
6 PARA 6.0000G          2.0000G  3.0000G

```

```

CURRENT COMPONENT IS # 1: PARA 0.0000F 5.5010F 2.5191F
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N d CR

```

If you want to delete component number 2, you see the following display:

```

ENTER # OF THE COMPONENT TO BE DELETED: 1 2 CR
COMPONENT # 2 WAS DELETED

```

```

CURRENT COMPONENT IS # 2: GAUS 0.0000F 16.3685F
(N=NEXT, E=EDIT, A=ALTER#, D=DELETE, I=INSERT, M=MAP, Q=QUIT): N * CR

```

The “*” produces a summary of the menu:

```

SUMMARY FOR THIS MENU:

1. TOTAL NUMBER OF COMPONENTS      :      5

2. COMPONENTS                       :

# TYPE POSITION      FWHM      CUT-OFF  BROADENING  INTENSITY
1 PARA 0.0000F      5.5010F  2.5191F     LIN.COMB.
2 GAUS 0.0000F      16.3685F          LIN.COMB.
3 GAUS 0.0000F      4.7605F          LIN.COMB.
4 GAUS 0.0000F      8.0000G
5 PARA 6.0000G      2.0000G  3.0000G

3. INTENSITY CONSTRAINT TYPE      : L
LC#  COMP.#  COEFF.  COMP.#  COEFF.  COMP.#  COEFF.
  1    1(PARA)  2.7286  3(GAUS)  27.4036
  2    1(PARA) -75.3250  2(GAUS)  27.4036

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF ACAREREDIT MENU # 5: COMPONENTS AND CONSTRAINTS

```

By comparing this summary with the one above, you can see that not only has one component been deleted and those which follow been renumbered, but in the linear combinations the component numbers have been adjusted to the new sequence of components. If instead components are inserted, an equivalent adjustment of component numbers takes place. These features greatly facilitate the set-up of linear combinations when many coefficients are involved (as may easily be the case, see Section 4.3). To further ease the set-up, two of the actions above, viz. “EDIT” and “INSERT” are automatically maintained from one component to the next, until another action is entered. In the other cases “NEXT” will always be the default value. Likewise, an entered value of a coefficient in a linear combination will be maintained as the default value until another value is entered.

You might have noticed a pronounced difference in design and organization of the component and constraint input in ACAREREDIT compared to the corresponding menus in POSEDIT and RESEDIT. This is particularly the case for the “linear combination” constraints. The reason lies again in the often large number of components and constraints in a typical ACARFIT

analysis.

After the summary has been accepted, the next menu which deals with the background appears:

Menu 6: Background

```
ACAREFIT SELECTION IS # 6: BACKGROUND
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 6: BACKGROUND
1. BACKGROUND TO BE SUBTRACTED      : 0.00000E+00 CR
```

```
SUMMARY FOR THIS MENU:
1. BACKGROUND TO BE SUBTRACTED      : 0.00000E+00
```

```
ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
END OF ACAREFIT MENU # 6: BACKGROUND
```

Menu 6 contains only one question, viz. the value of a constant background. In the present case it is assumed to be zero.

Finally, the “Export Control Data” menu is reached. This menu was described in detail above, in the general description of the –EDIT programs, and shall therefore not be discussed further here:

Menu 7: Export Control Data

```
ACAREFIT SELECTION IS # 7: EXPORT ACAREFIT CONTROL DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 7: EXPORT ACAREFIT CONTROL DATA
1. LIST CONTROL DATA (Y,N)         : N CR
2. SAVE DATA IN FILE WITH NAME: MORTEN.ACA sample.aca CR
   FILE EXISTS: SAMPLE.ACA
   (A=APPEND, O=OVERWRITE, Q=QUIT): A o CR
```

```
SUMMARY FOR THIS MENU:
1. LIST CONTROL DATA (Y,N)         : N
2. SAVE DATA IN FILE WITH NAME: SAMPLE.ACA
```

```
** CONTROL DATA WAS OVERWRITTEN IN A FILE **
```

```
CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
```

```
=====TERMINAL POINT FOR ACAREFIT MAIN MENU=====
KEY <*> OR <*E> TO EXIT FROM EDITOR, <RETURN> TO RECYCLE THE MENU: * CR
```

The Control File, called SAMPLE.ACA, generated in the example above, is listed in Appendix C.3. This is the type of listing you obtain if you answer Yes to the first question in Menu 7 (or 2nd question in Menu 1). Appendix C.3 also contains a detailed discussion of the structure of the Control Data Set. The Main Output from an ACAREFIT run with SAMPLE.ACA as input is shown in Section 2.3.

4 Experience with PATFIT

In this Section we shall briefly discuss some of the experiences we (and others) have had with the programs in present and earlier versions. Further discussion can be found in [5] parts of which are repeated below.

Editor programs

The three –EDIT programs have been used a large number of times to set up Control Data for the fitting programs, both for testing the programs and for ordinary use. All through this, the –EDIT programs have been proved to be very easy and safe to use. One drawback, though, will be apparent if many data sets are incorporated into one or both of the data files with which an –EDIT program communicates (Spectrum and Control Files, see Fig. 1, Section 3.1). In such cases, the time required to access these files may become rather long on a standard PC, and one might prefer to split the files into two or more.

Fitting programs

The fitting programs, both in the present and earlier versions, have also been proved to be very reliable and easy to use, in particular in connection with the new –EDIT programs.

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 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 to the fitting programs, 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, for the case of lifetime spectra in e.g. [10, 11, 12, 13, 14, 15, 16]. In this connection, an advantage of the programs is the possibility of various types of constraints which makes it possible to select meaningful numbers and types of fitting parameters.

4.1 POSITRONFIT

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 [17, 18, 19] while in some insulators where positronium is formed, up to four components may be extracted e.g. [20, 21, 22, 23]. (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., refs. [10, 16]). 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).

Contrary to RESOLUTION (see below), in our experience POSITRONFIT always produces the same estimates of the fitted parameters after convergence, irrespective of the initial guesses (except in some extreme cases). However, others have informed us that for spectra containing very many counts (of the order of 10^7) one may obtain different results, depending upon the initial guesses on the fitting parameters, i.e. local minima exist in the χ^2 as function of the fitting parameters; these minima are often quite shallow. When this happens, POSITRONFIT as well as most other least-squares fitting codes are in trouble, because they just find some local minimum. From a single fitting you cannot know whether the absolute minimum in the parameter space has been found. The problem of “global minimization” is very much harder to solve (though not impossible), 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.

4.2 RESOLUTION

When using a program such as RESOLUTION an important question of course is whether it is possible in practice to separate the resolution curve reliably from the lifetime components. Our experience and those of others [6, 7, 8, 16, 23] suggest that this separation is possible, although in general great care is necessary to obtain well-defined results [5, 16]. 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 some of the parameters defining the resolution curve couple too strongly to the lifetime parameters, in particular when three Gaussians (or more) are used to describe the resolution curve. As in the example used in this report (Section 2.2), we have often measured annealed copper in order to deduce the resolution curve. Even with different settings of the lifetime spectrometer, the copper lifetime normally comes out from a RESOLUTION fit within a couple of ps (statistical scatter) from the 112 ps obtained here (in agreement with recent results of others, e.g. [24]). 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 curve, one frequently experiences that two (or more) different sets of resolution curve parameters may be obtained from the same spectrum in different analy-

ses, if different initial guesses are applied. The lifetimes and intensities come out essentially the same in the different analyses, the fits are almost equally good, and a comparison of the widths at the various heights of the resolution curves obtained in the analyses show that they are essentially identical. Thus, in spite of the many fitting parameters (i.e. so many that the same resolution curve may be described by more than one set of parameters), it still seems possible to separate the lifetimes and resolution function reliably, at least when the lifetime spectrum contains a short-lived component of about 90 % intensity or more.

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

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, 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 resolution curve, or if some other functional form, e.g., a Gaussian convoluted with two exponentials [6, 8], is better. Of course, it will depend on the detailed shape of the instrumental resolution curve, but practical experience seems to show that the two descriptions give only small differences in the extracted shape of the curve [8, 16], and the better the resolution is, the less does a small difference influence the extracted lifetime parameters [16]. The sum-of-Gaussians used in the PATFIT programs 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 programs, 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 these are will depend on energy window settings and source-sample-detector arrangement of the lifetime spectrometer [16, 25, 26]. (Apart from the distortions, these spectrometer characteristics will, of course, also influence the width and shape of the correct resolution function.)

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 practical to always measure the annealed sample in exactly the same physical arrangement as the defect containing sample (e.g. 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 curve is then determined from measurements on the annealed sample outside the cryostat, but will be 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 uncertainties and 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 closely as possible to a reliable resolution function. We are reluctant to determine the resolution function directly from the spectra for the defected metal crystals, 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 RESOLUTION analysis apart from the extraction of the resolution curve, viz. the determination of the “source correction” (Section 3.2). If the sample gives rise to only one lifetime component, any remaining components must be due to positrons annihilating elsewhere and is therefore normally considered as a source correction. In the RESOLUTION Main Output (Section 2.2) the 0.112 ns is the annealed Cu lifetime, while the 0.18 ns, 5% 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 0.401 ns, 4.633% 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). Finally, the very weak longlived component (if at all present) may be due to Ps or positrons escaping the sample sandwich. We consider the three latter components as corrections to the measured spectrum in any subsequent POSITRONFIT analysis (when the same source and similar sample material have been used).

4.3 ACARFIT

Over a number of years very many ACAR curves have been analysed with ACARFIT and — in particular — the earlier versions (PAACFIT and PARAFIT). The programs have been shown to be very reliable in the sense that rapid convergence is always obtained (except in extreme cases). Often the “Variance of the Fit” exceeds significantly the value expected for a good fit. One reason for this can be that the model (a sum of a certain number of Gaussians and parabolas) is not ideal, i.e. the model function cannot fit the data in all details, and more, maybe sidepeak-, components should be added to the model. Another reason may be that the spectra contain small errors caused by various experimental effects. An important error is the asymmetry which may arise, partly because of the penetration of the positrons into the sample, and partly because of the absorption of the annihilation quanta in the sample [27]. A study of the deviation plot (in the ACARFIT output or by PATGRAPH) is useful for determining the nature of the misfit. For many analyses though, a “Variance of the Fit” below 1.5 – 2.0 will be satisfactory.

In simple metals (without defects) with an almost spherical Fermi surface

the conduction electrons will — to a good approximation — give rise to a parabolic component in 1D-ACAR curves, while a broad Gaussian contains contributions from core electrons and other “high-momentum components”. In such cases, fitting of the ACAR curves results in components, each of which may have a direct physical meaning (e.g. the parabola cut-off determines the Fermi momentum.) A few other examples exist where components derived from the fitting of ACAR curves clearly have a direct physical meaning (e.g. narrow para-Pos peaks in certain insulators [28]). However, normally that is not the case and no à priori functional form is assumed known for a measured ACAR spectrum (contrary to lifetime spectra). Hence, the fitting of a spectrum in terms of Gaussians and parabolas often produces only a parametrisation of the spectrum at first, with no direct physical meaning attached to the individual components necessarily. However, the shape of the spectrum is determined by the fitting. An analysis principle that we have often used is to fit a measured curve by a superposition of “basis-curves” of known shapes, thus determining the weights of each “basis-curve” by the fitting procedure. This has been discussed, e.g., for (a) defects in metals [17, 29] where the “basis-curves” are those for the bulk metal (measured independently) and for one or two types of defects (vacancies and voids in Mo, He-bubbles in Al), for (b) positron trapping on halide ions (the “basis-curves” are those for positrons bound to Cl^- , Br^- , and I^- ions) [30], and for (c) positrons and positronium in a molecular crystal (where the “basis-curves” are the shapes of the curves for positrons, para-positronium and ortho-positronium in solid pivalic acid) [31]. Discussions of these analyses are also included in [5, 16].

When you want to analyse in terms of superpositions of shapes, it is mandatory to use constraints of the “linear combination of intensities” type. A simple example, using the analysis in the Main Output given in Section 2.3, will illustrate this. An ACAR curve measured for annealed polycrystalline copper is well fitted by three central peaks, two Gaussians and one parabola. The fitted parameters are: Gaussians: $\text{FWHM}_1^g = 16.3685$ mrad, $I_1^g = 75.3250\%$, $\text{FWHM}_2^g = 4.7605$ mrad, $I_2^g = -2.7286\%$; Parabola: $\theta_1^c = 5.5010$ mrad, $\text{FWHM}_1^p = 2.5191$ mrad, $I_1^p = 27.4036\%$. These parameters then represent the shape of the annealed-Cu-curve. (The negative intensity of one component means only that the ACAR curve is more flat in the top than can be described by one Gaussian and one folded parabola). We then wanted to fit the ACAR spectrum of Section 2.3 (deformed Cu) with one basis-curve of the shape of annealed Cu and one more curve arising from positrons trapped in defects. Hence, the three first components are chosen to represent the Cu shape by fixing all their width and (by using the “linear-combination” constraint) the ratios of the intensities to $I_1^p/I_2^g = -27.4036/2.7286$ and $I_1^p/I_1^g = 27.4036/75.3250$. The sum of the two components with numbers 4 and 5 then arise from trapped positrons.

It must be added though, that defect components determined in this way

do not necessarily represent the total defect curve (and in this example do not). This is so, because any fraction of the total defect curve which has the shape of an annealed-Cu curve will be incorporated in the first three components (Section 2.3). However, if the shape of the total defect curve were known, this shape could be constrained too in the analysis and the fraction of trapped positrons could then be determined. This approach has been applied in positron chemistry [16, 30]. Furthermore, if a good fit cannot be obtained this way, the analysis testifies either that one or more of the chosen shapes are wrong, or that additional components are present [16].

Let us conclude this Section on experiences with the programs by quoting one of the last sentences in [7]: “The extraction of reasonably complete yet meaningful information on the decay components present in a complicated positron lifetime spectrum is a very subtle problem. Experience seems to be essential”. This also holds true for the analysis of ACAR curves.

5 Installation and Operation of PATFIT-88

This Section gives some information on the installation and operation of the FORTRAN programs in PATFIT-88 (POSITRONFIT, RESOLUTION, ACARFIT, POSEDIT, RESEDIT, and ACAREEDIT), on microcomputers as well as on larger multi-user installations. PATGRAPH is described separately in Section 6.

5.1 General Remarks

If you want to install PATFIT-88 as it is, you should have just a little knowledge about your operating system (DOS, VMS, UNIX, or whatever). You need not be a FORTRAN expert. (Some understanding of FORTRAN formats will, however, be needed if you read external spectra.) Of course, if you want to modify the software, you must edit the FORTRAN source codes, and subsequently compile and link them with the FORTRAN Compiler and Linker available to you. This may also be necessary in case you should discover a bug in PATFIT-88.

Easy portability was one of our principle design criteria for PATFIT-88. For this reason we decided to use standard-conforming FORTRAN 77 as our programming language, adhering strictly to the ANSI 1978 Standard [32]. After testing with many different FORTRAN compilers we believe that this objective is indeed met.

Only four small subroutines in PATFIT-88 could not be written in Standard FORTRAN. These are, the routine DATJOB for returning the current date, the timing routine MCLOCK, the routine OPENCL to open new files without FORTRAN carriage control (to accommodate for a DEC peculiarity), and the routine STONUM, which tells how many bytes correspond to one numerical storage unit when using unformatted disk I/O under direct access. These four subroutines are collected in separate "EXTRA" files, one for each compiler where PATFIT-88 is implemented.

The PATFIT-88 programs are written in double-precision arithmetic; on most computers this is equivalent to 8-byte precision (REAL*8).

The three fitting programs utilize a common software package which we call MATHPACK. It is a collection of mathematical and other general-purpose subroutines, and it originates partly from "Risø Computer Library", and partly from the linear-algebra software package LINPACK [33]. PATFIT-88 has only capacity to handle problem sizes up to certain limits, expressed in terms of maximum values of some key input parameters. Many brands of microcomputers have a fixed addressable memory

**Limits to
problem size**

size of 640 kbyte. The PATFIT-88 programs keep well below this limit (in fact we have aimed at a size of about 360 kbyte for each executable program, in order to be able to store it on a single diskette). To cope with the restricted storage, the fitting programs are structured in a flexible way: In each of the fitting programs there is a “giant array” (WA), in which all the actual problem arrays are packed and compressed (the giant array size varies from problem to problem, and it can be increased by a recompilation, see below); the array portions are then associated with individual formal array parameters in various subroutines. This may be considered a substitute for the dynamic array-allocation capability, which the FORTRAN language is unfortunately missing. At present, there are the following limits on problem-size dependent parameters:

Lifetime programs POSITRONFIT and RESOLUTION:

Maximum number of channels: $MN = 8192$.

Maximum number of lifetime components: $ML = 6$.

(also valid for source-correction spectrum)

Maximum number of Gaussians in resolution function: $MG = 6$.

Maximum size of giant array WA (8-byte words): $LMAX = 19400$ in POSITRONFIT, $LMAX = 20500$ in RESOLUTION.

ACARFIT program:

Maximum number of angles (or channels): $MN = 8192$.

Maximum number of total components

(Gaussians + folded parabolas): $MT = 15$.

Maximum size of giant array WA (8-byte words): $LMAX = 18000$.

For big problems the giant-array capacity $LMAX$ may sometimes be taxed, even when none of the other limits are exceeded. When this happens you get an error message from the fitting programs: THIS PROBLEM IS TOO BIG FOR MACHINE SIZE, IT REQUIRES A RECOMPILATION WITH $LMAX$ AT LEAST If you use the -EDIT programs for preparing the input, already here you will get a warning.

If you have access to a computer system with a larger memory capacity you may wish to increase one or more of the stated limits. In the FORTRAN source codes they are mentioned in PARAMETER statements, for example PARAMETER ($LMAX=19400$). With your text editor you may easily modify these numbers throughout (recall that the scope of PARAMETER is a program unit, not the complete executable program). No such limit changes will affect MATHPACK. Other FORTRAN parameters are found in the BLOCK DATA subprograms at the end of the -EDIT programs. Here you will find the symbolic unit numbers for the various FORTRAN files in use. These might need changes to meet the requirements of your system. You will also here find the “escape character” ESC, defined to be the character “*”. You may redefine it if another symbol is more convenient. Also, in BLOCK DATA the FORTRAN carriage control

BLOCK DATA parameters

character CTR is defined to be “\$”. This was used because then the cursor stays at the end of the current line on prompting, when using a DEC-VAX computer; on non-DEC systems the \$ will normally be synonymous with a blank and therefore without effect (With most PC FORTRAN systems the cursor is by default at the line end; an exception is MicroSoft FORTRAN, see the next subsection).

When you try to open a file from FORTRAN, and this file is not resident, the FORTRAN error indicator IOSTAT for I/O handling takes on a positive value. This indicator is monitored and printed by the PATFIT programs, but the specific value is not standardized by FORTRAN and hence depends on the compiler used to compile PATFIT-88. In the following table we have collected IOSTAT values for no-file conditions for a number of compilers:

Ryan-McFarland Fortran: IOSTAT = 2015.

IBM Professional Fortran: IOSTAT = 2015.

MicroSoft Fortran: IOSTAT = 6416.

Lahey F77L Fortran: IOSTAT = 16455.

VAX-Fortran: IOSTAT = 29.

On still other systems, the missing file is not considered as an error condition, but a NO FILE message is given, and the computations suspended until the file is presented (example UNISYS B7800 and A-series).

PATFIT-88 may be delivered from Risø either on magnetic tape or on DOS-formatted diskettes in one of several possible standard formats and densities.

Open errors

Shipment

5.2 PC-versions

A main objective in the construction of PATFIT-88 was that the resulting system could run on small Personal Computers. The PC must be equipped with the MS/PC-DOS operating system. It must have an addressable RAM memory of minimum 640 kbyte. Another prerequisite is a math coprocessor 8087, 80287 or 80387. However, if the PC does not have one, the coprocessor will be emulated by the program (but then the computing time will be increased by about 10 times according to our measurements). Preferably the PC should have a hard disk, though this is not absolutely necessary. The PC-version of PATFIT-88 consists of a number of files. When you obtain the system from Risø, you will receive a README file which we strongly recommend that you study in detail. It gives installation prescriptions and contains auxiliary information, which may prove helpful. The executable programs are given the extension .EXE, for example POEDIT.EXE. Typically you will have PATFIT-88 installed on your hard disk (C: drive), where you might organize it within a special directory of name PATFIT say. Assuming you are already in this directory

Hardware demand

Files

you simply type POSEDIT when you want to run that program. From this point you may follow the guidance for POSEDIT given in Section 3.2, and similarly for the other programs. The README file also shows some files with extension .BAT. They act as “batch” or “command” files invoking the set of commands within the .BAT file itself.

FORTRAN compilation

The FORTRAN source text of PATFIT-88 is stored in files with extension .FOR, for example POSEDIT.FOR. These .FOR files are not needed for daily work with the system. However, if you plan to modify the software, you must recompile them.

The standard PC export version of the PATFIT-88 software was compiled with the Ryan-McFarland Fortran Compiler (“RMFort”). PATGRAPH is an exception, see Section 6. We used RMFort because this compiler compared favorably with others with respect to the size and efficiency of the resulting .EXE code files. In the developing phase we preferred the Lahey F77L Fortran Compiler (“F77L”) owing to its compilation speed and its fine facilities for check and debugging. Furthermore we have tested PATFIT-88 on the IBM Professional Fortran Compiler (“ProFort”), which is very similar to RMFort (and in fact also made by Ryan-McFarland), and on the MicroSoft Fortran Compiler (“MSFort”).

Warnings

Warning: We have experienced bugs in earlier versions of some compilers. If you want to recompile any of the programs, make sure to use the compiler version mentioned in the README file, or more recent versions.

Another warning: Beware of resident application programs, if you recompile PATFIT-88 on your PC. Experience shows that such programs may induce errors in the executable code file.

Link

The output from a FORTRAN compiler is normally a so-called object file with extension .OBJ. To obtain the .EXE code, a subsequent linkage is necessary (Some compilers, such as MSFort, may combine Compile and Link into one step). To link FORTRAN modules you may use your standard DOS LINK facility. At Risø we found it more efficient to use the MS-Overlay Linker; moreover we compressed the resulting .EXE files with the EXEPACK program (from the MS-EXE File Compression Utility). Details on how the PATFIT-88 modules are organized for the linkage procedure are given in the README file. You will need this information if you are going to modify the programs.

DOS limitations

File names in DOS are limited to 8 characters; longer names will be truncated but otherwise will do no harm. For instance, you may run POSITRONFIT by typing POSITRON or POSITRONFIT, as you like.

Other limitations in DOS are concerned with the number of files and buffers you can have open at the same time. The default limit set by DOS may be insufficient for some of the PATFIT-88 programs. We recommend that you increase them by inserting the lines FILES = 20 and BUFFERS = 20 in your CONFIG.SYS file.

As mentioned in Section 5.1 the nonstandard FORTRAN utilities in PATFIT-88 occur in four subroutines collected in a so-called “EXTRA” file. If you recompile PATFIT-88 you will need precisely the EXTRA-file that is tailored to your FORTRAN compiler. In the shipment from Risø there are four such EXTRA files, namely RMFEXTRA (also valid for ProFort), F77LEXTRA, MSEXTRA, and VAXEXTRA.

EXTRA-files

The three fitting programs POSITRONFIT, RESOLUTION, and ACARFIT use a temporary, unformatted scratch file for storing partial derivative matrices in packed form; these can for large problems not be held in memory. If you execute the RMFort- or ProFort-compiled version of one of the fitting programs, you will get an error message INTERNAL FILE-READING ERROR (with an explanation), if your problem is too big. The reason is that we, for greater efficiency, have organized the scratch file with a larger record length than allowed by the compiler default. The remedy is to increase the permitted maximum length of I/O data transfer. To do this you need a runtime option. If you type

Run-time option

```
POSITRONFIT /R 32760
```

you will never get this error stop. Notice that this problem only exists with RMFort and ProFort.

With DOS you have the possibility of easy redirection of input and output files to and from disk. Suppose you have an executable code file named XXX.EXE. You may then just type XXX, in which case the XXX program is executed with standard input entered from the keyboard and standard output going to the screen (and to a printer connected as a slave). If you have already prepared an input file say SAMPLE.XXX on the disk, you tell the system to use this as input by typing XXX < SAMPLE.XXX. Similarly XXX > OUTPUT.XXX will cause the standard output to be redirected from screen to the diskfile OUTPUT.XXX. You may combine < and > by typing XXX < SAMPLE.XXX > OUTPUT.XXX. This file direction capability is very practical when you run the three fitting programs POSITRONFIT, RESOLUTION, and ACARFIT. An example would be:

Redirection of files

```
POSITRONFIT /R 32760 < SAMPLE.POS > OUTPUT.POS
```

If you use the standard version of the -EDIT programs compiled with MS-Fort you will notice that the cursor following a prompt does not stay at the end of the present line, but rather at the beginning of the next. In case you wish to alter this, you must seek out all the appropriate FORMAT statements in the editor program and end them with a backslash character (\). This nonstandard FORTRAN extension is described in the MS-Fort Manual.

Carriage control

5.3 Mainframe versions

PATFIT-88 runs perfectly well on larger computers like super-minis or mainframe systems.

VAX

At Risø most experience on such computers has been gathered with our DEC VAX-8700 super-mini. The operating system of VAX (VMS) is not too far away from the DOS for PC. In particular the standard file-name extensions .FOR, .OBJ, and .EXE, are also recognized by VAX, and a similar LINK command (only simpler) is also available.

VAX FORTRAN supports two different types of double precision (REAL*8), called D-floating and G-floating, respectively. We strongly recommend that the fitting programs POSITRONFIT, RESOLUTION, and ACARFIT be compiled with G-floating. D-floating is the default, so you must type say FORTRAN/G POSITRONFIT. The result is a very large increase in the dynamical floating-point number range at the expense of a modest decrease in precision. This will prevent over- and underflows in the fitting programs. In fact VAX G-floating gives exactly the same floating-point representation as does the double precision on Personal Computers (both conform with the so-called IEEE standard for floating-point representation).

Like other major computer systems the VAX has virtual memory allocation. Therefore, if you need to increase the limits for indata mentioned in Section 5.1, this can be carried out easily, merely by recompiling with the larger limits.

The FORTRAN source text of PATFIT-88 for VAX is exactly the same as that for PC, save for the "EXTRA" file. A file called VAXEXTRA.FOR is available from Risø, cf. the README file mentioned in Section 5.2.

UNISYS

You will get a dramatical reduction of execution times with PATFIT-88 on say a VAX-8700 compared to an ordinary PC. We have also tried to run PATFIT-88 on the UNISYS B7800 Mainframe computer at Risø. This machine differs somewhat from both PC and VAX systems in its file handling. Therefore some additional file-declaration statements were needed. B7800 has no separate Linker. Furthermore, the standard precision is 48 bit rather than 32, so we may do without double precision here (by a compiler directive we could tell the computer to interpret all declared double-precision quantities as single precision.) The subroutines in the "EXTRA" file are also different. Accommodation of these differences allows the system to run on the B7800.

Notice that PATGRAPH is only available in a PC version.

6 PATGRAPH: Program for Graphical Output

The purpose of the PATGRAPH program is to display measured and/or fitted model spectra in graphical form on the screen. Furthermore, a normalized plot of the deviations between measured and fitted spectra can be displayed. The data to be displayed are read by the program from a Communication File as shown in Fig. 2.

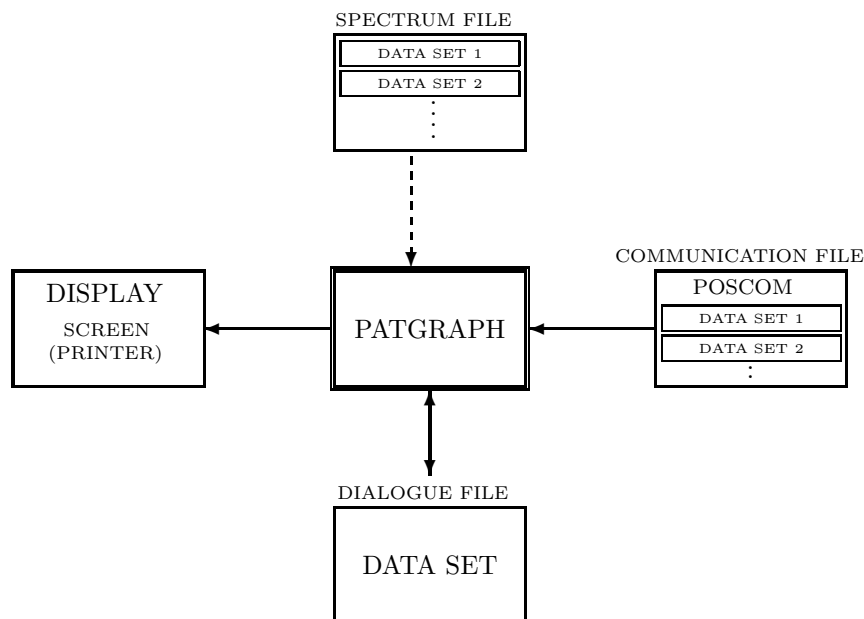


Figure 2. Schematic representation of the interaction between the PATGRAPH program, files and output device. The arrows indicate data transfer between the program and a file or output device. The dashed line represents an optional data transfer. Compare also Fig. 1 in Section 3, of which this figure is an excerpt.

A Communication File contains one or several Communication Data Sets. The main contents of each of these Data Sets are a measured spectrum (optional), the Main Output and the Correlation Matrix. The Communication Data Set has been generated by the fitting program as a (optional) part of the output (see Section 2 and Fig. 1). Thus PATGRAPH can display spectra only after they have been fitted.

The functioning of PATGRAPH is interactive in a way very similar to the -EDIT programs (Section 3). Many of the general features described in

Section 3.1 therefore also apply to PATGRAPH. As will be described in more detail below, the program first searches out the desired spectrum and analysis from the Communication File, then displays the measured spectrum (optional). Next, it calculates the fitted model function from the parameters in the Main Output, using the proper mathematical model (Section 2), and displays this function (optional). Finally, the deviations between model function and measured spectrum can be displayed (optional). The program detects the type (Lifetime or ACAR) of the selected spectrum, and automatically chooses the proper functional form of the model as well as the proper parameters for the graphical display.

The display routines used in PATGRAPH are taken from the standard MetaWINDOW program package (copyright of Metagraphics) [34].

6.1 Details of PATGRAPH menus

In the following we shall discuss PATGRAPH in some detail by going through two runs, one showing the display of a lifetime spectrum (the Main Output of which is shown in Section 2.1) and the other one the display of an ACAR spectrum (the Main Output of which is shown in Section 2.3).

Main Menu

On starting the program, after the display of a heading, the Main Menu is displayed. It contains a listing of four local Menus. We shall discuss each of these:

```
PATGRAPH MAIN MENU:
  1. IMPORT PATGRAPH DIALOGUE DATA
  2. GRAPHICS CARD AND COLOURS
  3. SPECTRUM SELECTION AND DISPLAY
  4. EXPORT PATGRAPH DIALOGUE DATA
```

```
CURRENT PATGRAPH SELECTION IS # 1: IMPORT PATGRAPH DIALOGUE DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

Menu 1: Import Dialogue Data

After the Main Menu, Local Menu No. 1 is automatically selected. If accepted, as above, the following is displayed:

```
LOCAL MENU # 1: IMPORT PATGRAPH DIALOGUE DATA
  1. LOAD DIALOGUE DATA FROM A FILE (Y,N): N y CR
     NAME OF INPUT DIALOGUE FILE: PGFILE. CR
```

The so-called Dialogue Data are the data that are entered in Menus 2 and 3. The Dialogue Data may be saved in a Dialogue File (in Menu 4) so that the data can be used as default values in a future run of PATGRAPH. The import of the Dialogue Data takes place in the present Menu #1. If a “Y” is chosen for the first equation, the PGFILE acts as a default. In the example above, this file was accepted. However, if no Dialogue File is loaded (by accepting the “N” in the first question above), standard default values will be generated by the program.

A summary of the Menu follows next. If acceptable, a CR takes you to the end of the Menu:

SUMMARY FOR THIS MENU:

1. LOAD DIALOGUE DATA FROM A FILE (Y,N): Y

DIALOGUE DATA WAS LOADED FROM A FILE

CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

END OF PATGRAPH MENU # 1: IMPORT PATGRAPH DIALOGUE DATA

Local Menu No. 2 deals with identification of the graphics card installed in the PC and with a selection of colours to be used for the display on the screen.

Graphics Card and Colours

MetaWINDOW defines a list of code numbers for a range of graphics devices (cards) and display modes. According to Metagraphics [34], the software is compatible with any of these options:

128 = AT&T Graphics Adaptor	640x400 monochrome
129 = AT&T Display Enhancement Board (DEB)	640x200 16-color
130 = AT&T Display Enhancement Board (DEB)	640x400 16-color
8192 = Conographics Adaptor	640x400 16-color
8193 = Conographics Adaptor	512x512 16-color
1024 = Everex Edge Adaptor	640x200 16-color
1025 = Everex Edge Adaptor	640x400 4-color
8 = Hercules/AST Monochrome Graphics Adaptor	720x348 monochrome
4 = IBM Color Graphics Adaptor (CGA)	320x200 4-color
6 = IBM Color Graphics Adaptor (CGA)	640x200 2-color
512 = IBM Enhanced Graphics Adaptor (EGA)	640x350 monochrome
528 = IBM Enhanced Graphics Adaptor (EGA)	640x200 16-color
544 = IBM Enhanced Graphics Adaptor (EGA)	320x200 16-color
560 = IBM Enhanced Graphics Adaptor (EGA 128Kb)	640x350 16-color
561 = IBM Enhanced Graphics Adaptor (EGA)	640x350 2-color
562 = IBM PS/2 Video Graphics Array (VGA)	640x480 16-color
563 = IBM PS/2 MultiColor Graphics Array (MCGA)	640x480 2-color
2048 = IBM 3270 PC	720x350 monochrome
2049 = IBM 3270 PC	360x350 4-color
2 = MDS Genius Display Adaptor	736x1008 monochrome
32 = Sigma Design Color-400 Adaptor	640x400 16-color
256 = STB GraphicsPlus-II Adaptor	640x352 monochrome
257 = STB GraphicsPlus-II Adaptor	320x200 16-color
258 = STB GraphicsPlus-II Adaptor	640x200 4-color
384 = STB GraphicsPlus-III Adaptor	640x400 monochrome
516 = Tseng Labs EVA/480 or NEC GB-1 Adaptor	640x480 16-color
16384 = Toshiba 3100	640x400 monochrome
16 = Tecmar Graphics-Master Adaptor	720x352 monochrome
17 = Tecmar Graphics-Master Adaptor	720x704 monochrome
19 = Tecmar Graphics-Master Adaptor	640x200 16-color
18 = Tecmar Graphics-Master Adaptor	320x200 16-color
20 = Tecmar Graphics-Master Adaptor	640x400 16-color
520 = Video-7 Vega Deluxe or Quadram ProSync	640x480 16-color
521 = Video-7 Vega Deluxe or Quadram ProSync	752x410 16-color
66 = Wyse WY-700 Graphics Display	640x400 monochrome
65 = Wyse WY-700 Graphics Display	1280x400 monochrome
64 = Wyse WY-700 Graphics Display	1280x800 monochrome

The proper code number is defined in the first question of Menu 2.

CURRENT PATGRAPH SELECTION IS # 2: GRAPHICS CARD AND COLOURS

ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR

LOCAL MENU # 2: GRAPHICS CARD AND COLOURS

1. CODE FOR GRAPHICS CARD

MetaWINDOW'S PREDICTION (MAYBE WRONG) IS: 128

ACTUAL CHOICE OF GRAPHICS CARD CODE : 560 CR

The MetaWINDOW software is in principle able to find out which graphics card has been installed. However, experience shows that the prediction may be wrong, as shown in the above example. Hence, you have to enter the code of the actually installed card, in the example above the commonly used IBM E.G.A., 16 colour, 640 × 350 pixel resolution card. If a wrong code is chosen the running of PATGRAPH will break down.

The next questions deal with the selection of colours for the graphical display. The colour codes to be entered depend on the graphics card in the PC. It will therefore take us into too much detail to discuss these codes here; each user should define her/his own codes. The present example is briefly discussed below:

2. COLOUR CODE FOR FRAME : 5 CR
3. COLOUR CODE FOR FRAMETEXT : 7 CR
4. COLOUR CODE FOR POINT PIXEL : 14 CR
5. COLOUR CODE FOR POINT MARKER : 5 CR
6. COLOUR CODE FOR FITTED CURVE : 1 CR
7. COLOUR CODE FOR DIALOGUE TEXT : 7 CR
8. CODE FOR ANTI-COLOUR TO DIALOGUE TEXT : 0 CR

As will be seen below, the display of a spectrum contains a frame as well as numbers and text associated with the frame. The colours for these are chosen in questions 2 and 3. Each data point is shown by one pixel which is surrounded by a "marker". The colour for these can be chosen independently (questions 4 and 5), as can the colour of the fitted curve (question 6). A dialogue is taken place while a spectrum is being displayed (see Local Menu 3). The colour for this dialogue text is selected in question 7. It is necessary to remove this dialogue text by overwriting it with the "opposite" colour. This "anti-colour" is defined in question 8.

Of course, the data in Menu 2 need normally only be entered once. The summary ends the Menu:

SUMMARY FOR THIS MENU:

1. CODE FOR GRAPHICS CARD

MetaWINDOW'S PREDICTION (MAYBE WRONG) IS: 128

ACTUAL CHOICE OF GRAPHICS CARD CODE : 560

2. COLOUR CODE FOR FRAME : 5

3. COLOUR CODE FOR FRAMETEXT : 7

4. COLOUR CODE FOR POINT PIXEL : 14

5. COLOUR CODE FOR POINT MARKER : 5

6. COLOUR CODE FOR FITTED CURVE : 1

7. COLOUR CODE FOR DIALOGUE TEXT : 7

8. CODE FOR ANTI-COLOUR TO DIALOGUE TEXT : 0

ACCEPT (<RETURN>) OR MODIFY (*<ITEM NO.>): CR

END OF PATGRAPH MENU # 2: GRAPHICS CARD AND COLOURS

Local menu No. 3 is responsible for the main function of the PATGRAPH program, viz. selection of the spectrum and its display.

Menu 3: Spectrum Selection and Display

The Menu is divided into two parts. The first two questions define the spectrum to be displayed. Secondly you enter into a Submenu, in which parameters governing the actual display are entered and the display shown. There are two types of Submenus; one for display of lifetime spectra and one for display of ACAR curves. The program automatically selects the proper one in accordance with the type of fitting program used to generate the Communication Data Set.

The first question in Menu 3 defines the Communication File from which a Data Set is to be read:

```
CURRENT PATGRAPH SELECTION IS # 3: SPECTRUM SELECTION AND DISPLAY
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 3: SPECTRUM SELECTION AND DISPLAY
1. COMMUNICATION FILE, NAME IS: POSCOM CR
```

The file name POSCOM is accepted in order to display a spectrum analysed by POSITRONFIT (Section 2.1). To answer the second question you have to enter the search string by which the spectrum to be displayed is identified:

```
2. SPECTRUM SEARCH STRING : P RTNS CR
```

This search string can be any text string of max. 32 characters that is a part of the spectrum headline. (This search mechanism is somewhat different from the one used in the other programs.)

Once the search string has been entered, a summary is displayed and the program starts a search from the top through the Communication Data Sets in the file POSCOM. It stops the first time the search string is encountered. (More specifically, the search string and the line are stripped for trailing blanks, and if either is contained in the other, we have a match.) The spectrum headline is displayed as well as the time and date of the particular analysis (the result of which has been saved in the encountered Communication Data Set):

```
SPECTRUM IN COMMUNICATION FILE:      POSCOM
IS IDENTIFIED BY SEARCH STRING:      RTNS
```

```
SEARCH FROM TOP OF COMMUNICATION FILE
```

```
ANALYSIS TIME STAMP AND SPECTRUM HEDALINE:
```

```
P O S I T R O N F I T . . . . . ANALYSIS TIME AND DATE: 09:46:31 23-JAN-89
34800 CU,RTNS II,61.4 DPA,T30, K=153                07:18:16 29-JUN-87
```

```
ACCEPT (A), CONTINUE SEARCH (C), SEARCH FROM TOP (T),
NEW COMMUNICATION FILE (*1), NEW SEARCH STRING (*2),
OR QUIT (Q OR *): A CR
```

The time and date of analysis are used to identify different analyses of the same spectrum. You are then asked whether you accept the encoun-

tered Data Set for display, or you wish to continue the search down the Communication File to try to find another Data Set which originates from another analysis. You may also choose to start from the top of the file (in some situations this may be useful, although not in the present one). If a new Communication File or spectrum is wanted you may return to one of the first questions in the normal way (* <item No.>). Finally, you may quit the Menu. In the present case the encountered Communication Data Set was accepted, and you automatically enter a Submenu:

==== YOU NOW ENTER LIFETIME DISPLAY SUBMENU =====

```
PEAK VALUE OF SPECTRUM (-BACKGR)= 6.48542E+04
TOTAL FIR AREA (-BACKGR)      = 8.33140E+05
SPECTRUM PEAK IS IN CHANNEL:   109
FIT RANGE IS FROM CHANNEL      100 TO 300
```

1. NORMALIZATION VALUE AND TYPE(P,A): 6.48542E+04P 1e5 CR

Lifetime Submenu

The Lifetime Display Submenu was automatically chosen, because the Communication Data Set originates from a POSITRONFIT analysis.

The display of a lifetime spectrum will take place in a semilogarithmic plot, the “count axis” ranging from 1 to 10^6 .

It is often useful to be able to normalize a spectrum to a certain peak height or a certain total area, for example if plots of two or more spectra shall be compared. Therefore, the first question in the Submenus deals with the normalization of the spectrum which will be characterized either by the peak height or the area (both with the background subtracted). A “P” indicates that the displayed or entered number is a peak value, while an “A” indicates an area value. If no letter is entered, a “P” is assumed. The area and peak values of the measured spectrum are displayed first, for information (maybe after subtraction of the “source correction” in a POSITRONFIT analysis). By entering another value, as above, the counts in all channels will be multiplied by the same factor.

The next two questions deal with the range of channels to be shown in the display. The default values are the lower and upper limit of the fit range, unless other values have been read from the Dialogue File:

```
SPECTRUM PEAK IS IN CHANNEL:   109
FIT RANGE IS FROM CHANNEL      100 TO 300
2. DISPLAY RANGE STARTS IN CH. # : 100 50 CR
3. DISPLAY RANGE ENDS   IN CH. # : 100 350 CR
```

The 4th question determines whether the spectrum is to be shown with or without the background subtracted while the last three questions deal with the data to be displayed. If a Yes is selected in all cases, we may have:

```
4. SUBTRACT BACKGROUND ? (Y,N): Y CR
5. SHOW MEASURED SPECTRUM ? (Y,N): Y CR
6. SHOW FITTED SPECTRUM ? (Y,N): Y CR
7. SHOW RESIDUAL PLOT ? (Y,N): Y CR
```

The measured spectrum will be shown between the channel limits chosen in questions 2 and 3, while the fitted spectrum will only be displayed in the fit range (specified in the line above question 2). The residual plot shows the deviations between the data points and the model function, normalized for each point to the expected standard deviation ($= (y_i)^{1/2}$, where y_i is the number of counts in channel i in the measured spectrum). A summary of the Submenu is finally displayed:

SUMMARY FOR LIFETIME DISPLAY SUBMENU:

ANALYSIS TIME STAMP AND SPECTRUM HEADLINE:

P O S I T R O N F I T ANALYSIS TIME AND DATE: 09:46:31 23-JAN-89
 34800 CU,RTNS II,61.4 DPA,T30, K=153 07:18:16 29-JUN-87

1. NORMALIZATION VALUE AND TYPE(P,A): 1.00000E+05P
2. DISPLAY RANGE STARTS IN CH. # : 50
3. DISPLAY RANGE ENDS IN CH. # : 350
4. SUBTRACT BACKGROUND ? (Y,N) : Y
5. SHOW MEASURED SPECTRUM ? (Y,N) : Y
6. SHOW FITTED SPECTRUM ? (Y,N) : Y
7. SHOW RESIDUAL PLOT ? (Y,N) : Y

QUIT SUBMENU (Q OR *), DISPLAY (D), OR MODIFY (*<ITEM NO.>): D CR

After the summary you may choose to quit (out of the Submenu), to display (default choice) or modify some of the parameters No. 1-7. If the display mode is accepted, as above, the spectrum will be displayed. A hardcopy of the display is shown in Fig. 3.

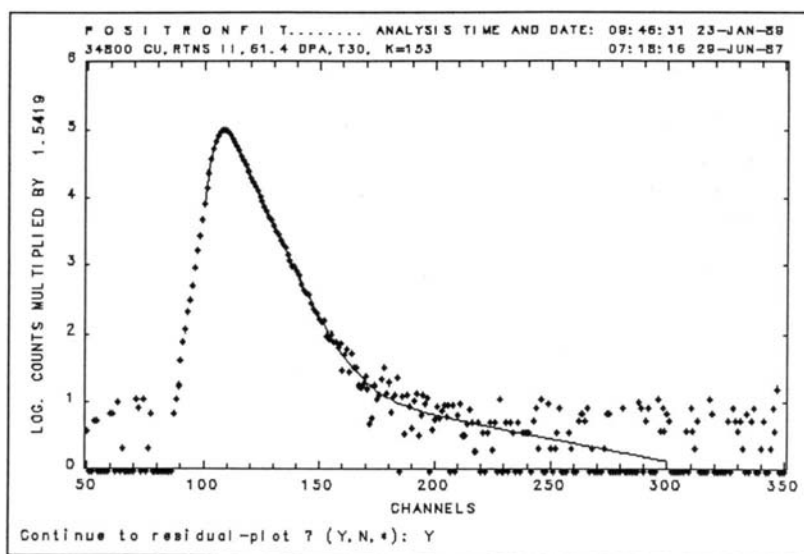


Figure 3: PATGRAPH display: Measured and fitted lifetime spectrum.

If the spectrum has been multiplied by a normalization factor different from 1 (as above), this factor will be shown along the vertical axis.

At the bottom of the screen a question is posed as whether to continue

to the residual plot (if a "Y" was entered to question 7 above). If "N" is entered now (will appear in the upper-left hand corner of the screen), no residual plot will be shown (in spite of the "Y" to question 7). If the "Y" is accepted a second question is posed, i.e. whether to clear the screen:

Clear the screen ? (Y,N,*): Y CR

If an "N" is entered, the residual plot will be superimposed on the spectrum; on accepting the "Y" only the residual plot will be displayed, as shown in Fig. 4, where the unit on the vertical scale for each point is the expected standard deviation ($= y_i^{1/2}$).

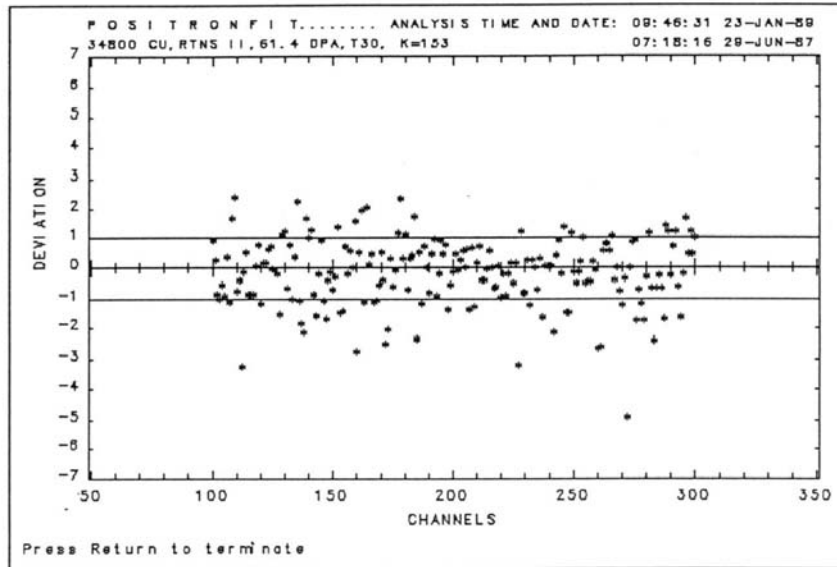


Figure 4: PATGRAPH display: Residual plot for lifetime spectrum.

Finally, you may return to the summary of the display Submenu by a CR:

Press Return to terminate CR

SUMMARY FOR LIFETIME DISPLAY SUBMENU:

ANALYSIS TIME STAMP AND SPECTRUM HEDALINE:

P O S I T R O N F I T ANALYSIS TIME AND DATE: 09:46:31 23-JAN-89
34800 CU,RTNS II,61.4 DPA,T30, K=153 07:18:16 29-JUN-87

1. NORMALIZATION VALUE AND TYPE(P,A): 1.00000E+05P
2. DISPLAY RANGE STARTS IN CH. # : 50
3. DISPLAY RANGE ENDS IN CH. # : 350
4. SUBTRACT BACKGROUND ? (Y,N) : Y
5. SHOW MEASURED SPECTRUM ? (Y,N) : Y
6. SHOW FITTED SPECTRUM ? (Y,N) : Y
7. SHOW RESIDUAL PLOT ? (Y,N) : Y

QUIT SUBMENU (Q OR *), DISPLAY (D), OR MODIFY (*<ITEM NO.>): D Q CR

You may now choose other display parameters or (as above) leave the

Submenu by entering a "Q":

DISPLAY SUBMENU FINISHED, RETURN TO SPECTRUM SELECTION.

ANALYSIS TIME STAMP AND SPECTRUM HEDALINE:

P O S I T R O N F I T..... ANALYSIS TIME AND DATE: 09:46:31 23-JAN-89
34800 CU,RTNS II,61.4 DPA,T30, K=153 07:18:16 29-JUN-87

ACCEPT (A), CONTINUE SEARCH (C), SEARCH FROM TOP (T),
NEW COMMUNICATION FILE (*1), NEW SEARCH STRING (*2),
OR QUIT (Q OR *): C *1 CR

To illustrate the display of an ACAR spectrum, we return to the first question in the Menu:

1. COMMUNICATION FILE, NAME IS: POSCOM CR
2. SPECTRUM SEARCH STRING : RTNS COPPER CR

SPECTRUM IN COMMUNICATION FILE: ACARCOM
IS IDENTIFIED BY SEARCH STRING: COPPER

The program now searches for a headline with the text string "COPPER":

SEARCH FROM TOP OF COMMUNICATION FILE

FORWARD SEARCH IN PROGRESS

ANALYSIS TIME STAMP AND SPECTRUM HEDALINE:

A C A R F I T ANALYSIS TIME AND DATE: 10:04:51 23-JAN-89
01591 COPPER,5N,CW 70% 06-AUG-86

ACCEPT (A), CONTINUE SEARCH (C), SEARCH FROM TOP (T),
NEW COMMUNICATION FILE (*1), NEW SEARCH STRING (*2),
OR QUIT (Q OR *): A CR

By accepting the encountered ACAR Communication Data Set you automatically enter the ACAR Display Submenu:

ACAR Submenu

===== YOU NOW ENTER ACAR DISPLAY SUBMENU =====

PEAK VALUE OF SPECTRUM (-BACKGR) = 1.53220E+04
FIT AREA (-BACKGR) (PEAK +- 13.00 MRAD)= 1.84901E+05
FIT RANGE IS FROM -15.64 MRAD TO 16.18 MRAD

1. NORMALIZATION VALUE AND TYPE(P,A): 1.53220E+04P 5e5a CR

As in the Lifetime Display Submenu, the peak value or the area (inside the normalization range; see Section 2.3) of the spectrum (with the background subtracted) may be normalized to a desired value, and the whole spectrum will be multiplied by the same factor. The standard ACAR display is linear, the maximum of the vertical scale being 5×10^4 .

The next question deals with the angular display range. It is given in mrad and is symmetrical around angle-zero (as defined by the Main Output). The default value is determined by the fit range (used in ACARFIT), unless a Dialogue File was imported in Local Menu 1. If the entered display range is not an integer, the actual range used for the display will be between - and + the next higher integer value (in the example below the

entered 20.4 will give a range of ± 21 mrad, see below):

2. DISPLAY RANGE (+-) IN MRAD : 16.1800 20.4 CR

The next four questions are identical to questions in the Lifetime Display Submenu:

- 3. SUBTRACT BACKGROUND ? (Y,N) : Y CR
- 4. SHOW MEASURED SPECTRUM ? (Y,N) : Y CR
- 5. SHOW FITTED SPECTRUM ? (Y,N) : Y CR
- 6. SHOW RESIDUAL PLOT ? (Y,N) : Y CR

The measured spectrum will be shown between the limits of the display range, while the fitted spectrum will only be displayed in the fit range. The residual plot is for each data point normalized to its standard deviation (taken from standard deviation table, if this is included in Spectrum Data Set (Section 2.3), or calculated from measured spectrum as $y_i^{1/2}$). Finally, a summary of the Submenu is displayed:

SUMMARY FOR ACAR DISPLAY SUBMENU:

ANALYSIS TIME STAMP AND SPECTRUM HEDALINE:

A C A R F I T ANALYSIS TIME AND DATE: 10:04:51 23-JAN-89
01591 COPPER,5N,CW 70% 06-AUG-86

- 1. NORMALIZATION VALUE AND TYPE(P,A): 5.00000E+05A
- 2. DISPLAY RANGE (+-) IN MRAD : 20.4000
- 3. SUBTRACT BACKGROUND ? (Y,N) : Y
- 4. SHOW MEASURED SPECTRUM ? (Y,N) : Y
- 5. SHOW FITTED SPECTRUM ? (Y,N) : Y
- 6. SHOW RESIDUAL PLOT ? (Y,N) : Y

QUIT SUBMENU (Q OR *), DISPLAY (D), OR MODIFY (*<ITEM NO.>): D CR

By accepting the display mode, as above, the ACAR spectrum will be displayed. A hardcopy of the display is shown in Fig. 5 below:

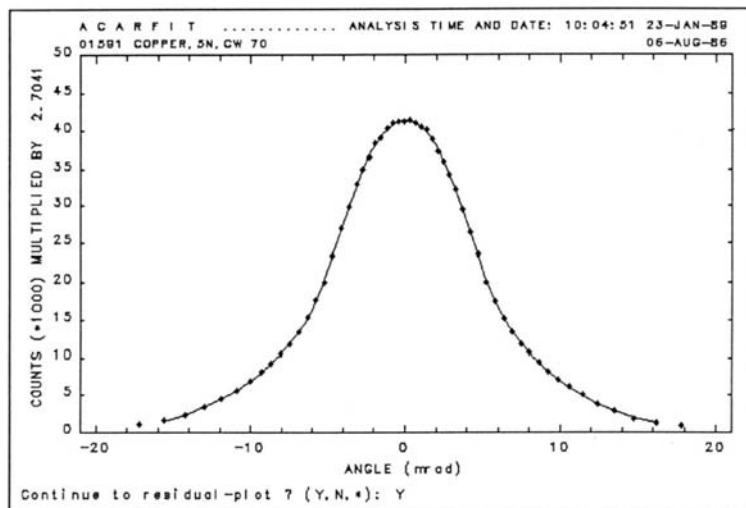


Figure 5: PATGRAPH display: Measured and fitted ACAR spectrum.

As in the lifetime display, a normalization constant different from 1 will be shown along the vertical axis.

The following questions relating to the display of the residual plot are identical to those in the Lifetime Display Submenu.

Likewise, after the displays you may return to the Summary for the display Submenu. By “quitting” twice, you leave Menu 3.

The last Menu, number 4, makes it possible to save the Dialogue Data in a Dialogue File. The values may then be used as defaults in another run of PATGRAPH. As mentioned earlier, the Dialogue File contains all the last data entered in Menus 2 and 3. If the last display was that of e.g. an ACAR curve, the Dialogue File also contains the data entered in the last Lifetime Submenu under the questions 1 and 2 (and vice versa). At first, a name for the Dialogue File must be chosen:

**Menu 4:
Export
Dialogue Data**

```
CURRENT PATGRAPH SELECTION IS # 4: EXPORT PATGRAPH DIALOGUE DATA
ACCEPT (<RETURN>), MAKE ANOTHER (*<NUMBER>), OR ASK HELP (*H): CR
```

```
LOCAL MENU # 4: EXPORT PATGRAPH DIALOGUE DATA
1. SAVE DIALOGUE DATA IN FILE: PGFILE. ACAFILE CR
```

```
SUMMARY FOR THIS MENU:
1. SAVE DIALOGUE DATA in FILE : ACAFILE
```

```
** DIALOGUE DATA WAS WRITTEN TO A NEW FILE **
```

```
CONTINUE (<RETURN>) OR MODIFY (*<ITEM NO.>): CR
```

(If the chosen file had already existed, you would have had to choose either to overwrite it with the new Dialogue Data or not to save the data, by quitting.) Next, a summary is displayed. On accepting the summary, you arrive at the terminal point of the program. You may leave the program by typing a * (or *E). A CR would take you to the beginning of the program again.

```
=====TERMINAL POINT FOR PATGRAPH MAIN MENU=====
KEY <*> OR <*E> TO EXIT FROM EDITOR, <RETURN> TO RECYCLE THE MENU: * CR
```

```
END OF PATGRAPH SESSION
```

6.2 Installation of PATGRAPH

This Section gives some information on the installation and operation of the FORTRAN program PATGRAPH on Personal Computers. (PATGRAPH will only run on PCs, and it has been tested under DOS operating systems.)

PATGRAPH contains licensed program materials of Metagraphics Software Corporation. Copyright ©1986, 1987 Metagraphics Software Corporation, Scotts Valley, CA 95066 USA (i.e. the routines METAWNDO.EXE, PRTSCN.EXE and ROMANSIM.FNT).

You are entitled to use the program PATGRAPH freely on your own installation without violating the copyright of Metagraphics.

PATGRAPH can be delivered from Risø on DOS-formatted diskettes in one of several possible standard formats and densities.

If you want to install PATGRAPH as it is, only a little knowledge about your DOS operation system is necessary. You need not be a FORTRAN expert. However, if you want to modify the software, you must edit the FORTRAN source code, then compile it with the Lahey F77L FORTRAN Compiler, and subsequently link it to the MetaWINDOW software (see the README file for an example).

The Personal Computer on which you want to run PATGRAPH must be equipped with the MS/PC-DOS operating system. It must have an addressable RAM memory of minimum 640 kbyte. Since PATGRAPH needs about 530 kbyte memory, rather strict limits exist on the size of other programs which are resident. It is an advantage to have a math coprocessor 8087, 80287 or 80387. However, if the PC does not have one, the coprocessor will be emulated by the program (but a drastic increase in computing time will result). Preferably the PC should have a hard disk, though this is not absolutely necessary.

PATGRAPH consists of a number of files. When you obtain the system from Risø, you will receive a README file which we strongly recommend that you study in detail. It gives installation prescriptions and contains auxiliary information, which may prove helpful. The executable program is called PATGRAPH.EXE. Typically you will have PATGRAPH installed on your hard disk (C:drive), where you might organize it within a directory of name PATFIT, say. (A special requirement is that the routine ROMAN-SIM.FNT must be found in a directory named PATFIT.) Assuming you are already in this directory, or you have made a path to the PATFIT directory (see the README file), you type

```
METAWNDO \PATFIT\PATGRAPH
```

when you want to run the program.

PATGRAPH requires the MetaWINDOW driver program (METAWNDO) to be resident in memory. Therefore, the run starts with the latter program being installed automatically. (Likewise at the end of the run the driver program is de-installed automatically. Some details are given in the README file). From this point you may follow the guidance for PATGRAPH given in Section 6.1.

If you want a hard copy on a matrix printer or a graphics display on the screen, the "Print-Screen Copy Utility" PRTSCRN may be used. It must be installed before METAWNDO (see also the README file). Use SHIFT-PRTSC to make the hardcopy. (NB: PRTSCRN cannot be de-installed).

The FORTRAN source text of PATGRAPH is stored in PATGRAPH.FOR. This .FOR file is not needed for daily work with the system. However, if you plan to modify the software, you must recompile and link it. To link FORTRAN modules you may use the standard DOS LINK facility. At Risø we found it more efficient to use the MS-Overlay Linker; moreover we compressed the resulting .EXE file with the EXEPACK program (from the MS-EXE File Compression Utility). Details on how PATGRAPH modules are organized for the linkage procedure are given in the README file. You will need this information if you are going to modify the programs.

In DOS there are limitations on the number of files and buffers you can have open at the same time. The default limit set by DOS may be insufficient for PATGRAPH. We recommend that you increase them by inserting the lines FILES = 20 and BUFFERS = 20 in your CONFIG.SYS file.

When you try to open a file from FORTRAN, and this file is not resident, the FORTRAN error indicator IOSTAT for I/O handling takes on a positive value. This indicator is monitored and printed by the program. The IOSTAT value for no-file condition is in Lahey F77L Fortran: IOSTAT = 16455.

Acknowledgements

We wish to thank Ole Mogensen for his contribution to the development of the earlier versions of PAACFIT and PARAFIT which served as the basis for the present ACARFIT program. We are also grateful to a number of colleagues who used and tested preliminary versions of some of the programs: C. Beling, S. Berko, J. Brunner, A. Dupasquier, F. M. Jacobsen, P. C. Jain, K. O. Jensen, B. Levay, and D. Segers. The present work was partly supported by the International Atomic Energy Agency.

Appendices

A Least-Squares Technique with Statistics

A.1 Semilinear Least-Squares Fitting

Throughout this work, model parameters are estimated by making use of least-squares techniques. We shall describe below the elements of the solution methods that are common to all three PATFIT-88 fitting codes in general terms. Mathematical details related to the specific models are given in Appendix B.

Unconstrained formulation

The weighted least-squares problem with n data points and k free parameters reads (cf. (1) in Section 2):

$$\min \left\{ \Phi \equiv \sum_{i=1}^n w_i (y_i - f_i(\mathbf{b}))^2 \equiv \|\mathbf{r}(\mathbf{b})\|^2 : \mathbf{b} \in \mathbb{R}^k \right\} \quad (\text{A1})$$

where y_i = observed value for data point no. i , $f_i(\mathbf{b})$ = model-predicted value for data point no. i , w_i = fixed positive weight attached to data point no. i , \mathbf{b} = k -vector of parameters to be estimated, $\mathbf{r}(\mathbf{b})$ = n -vector of scaled residuals, i.e. $\mathbf{r} = \{r_i\}$ with $r_i = w_i^{1/2}(y_i - f_i)$, $\|\cdot\|$ = Euclidean norm of a vector.

A substantial gain in computing efficiency can be obtained when some of the k components of the parameter vector \mathbf{b} enter the model $f_i(\mathbf{b})$ linearly. The least-squares problem is called separable in this case. Semilinear procedures for such problems have been studied by several authors [1, 35, 36, 37, 38, 39] and have proved successful in this work and in many other applications as well.

We assume that the actual model can be written

$$f_i(\mathbf{b}) = f_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^p \alpha_j f_{ij}(\boldsymbol{\beta}) = \{\mathbf{F}(\boldsymbol{\beta})\boldsymbol{\alpha}\}_i, \quad (\text{A2})$$

where $\mathbf{F} = \{f_{ij}\}$ is an $n \times p$ matrix. This corresponds to a partitioning of the k -vector (here and in the following, bordered indices for block matrices

and vectors denote the sizes of their partitions)

$$\mathbf{b} = \begin{matrix} p \\ q \end{matrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \quad (\text{A3})$$

into a “linear” p -vector $\boldsymbol{\alpha} = \{\alpha_j\}$ and a “nonlinear” q -vector $\boldsymbol{\beta} = \{\beta_j\}$, where $p + q = k$.

We shall scale \mathbf{F} and the data vector $\mathbf{y} = \{y_i\}$ by premultiplication by the diagonal matrix $\mathbf{W}^{1/2} = \text{diag}\{w_i^{1/2}\}$, i.e. we shall define $\mathbf{E} = \mathbf{W}^{1/2}\mathbf{F}$ and $\mathbf{z} = \mathbf{W}^{1/2}\mathbf{y}$. The residual vector entering (A1) equals $\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}$. Now, let $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\boldsymbol{\beta})$ be the solution of (A1) for $\boldsymbol{\alpha}$ conditioned on some fixed $\boldsymbol{\beta}$ i.e. the solution of the minimization problem

$$\min \{ \Phi(\boldsymbol{\alpha}) \equiv \|\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}\|^2 : \boldsymbol{\alpha} \in \mathbb{R}^p \}. \quad (\text{A4})$$

The standard least-squares analysis tells that $\boldsymbol{\alpha}$ is formally the solution of the p th-order linear system of “normal equations”

$$\mathbf{E}^T \mathbf{E} \boldsymbol{\alpha} = \mathbf{E}^T \mathbf{z}, \quad (\text{A5})$$

in which the $n \times p$ matrix \mathbf{E} plays the role of a “design matrix”.

A direct solution of the normal equations presents various numerical difficulties inherent with the ill-conditioning of the positive-definite coefficient matrix $\mathbf{E}^T \mathbf{E}$. Instead we use a procedure based on the so-called QR decomposition of \mathbf{E} , viz.

$$\mathbf{E} = \mathbf{Q}\mathbf{R}, \quad (\text{A6})$$

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 [33]). Using (A6) the system (A5) is reformulated to $\mathbf{R}\boldsymbol{\alpha} = \mathbf{Q}^T \mathbf{z}$, which can be easily solved by back-substitution.

We next turn to the determination of the nonlinear part $\boldsymbol{\beta}$ of the parameter vector \mathbf{b} and realize that an iterative method will be necessary. In fact, there will be an outer iteration loop, where each step provides a correction vector \mathbf{d} to $\boldsymbol{\beta}$, and an inner loop which contains an iterative computation of the next \mathbf{d} for fixed $\boldsymbol{\beta}$. The nonlinear least-squares problem is the following

$$\min \{ \Phi(\boldsymbol{\beta}) \equiv \|\mathbf{z} - \mathbf{E}(\boldsymbol{\beta})\boldsymbol{\alpha}(\boldsymbol{\beta})\|^2 : \boldsymbol{\beta} \in \mathbb{R}^q \}. \quad (\text{A7})$$

We solve (A7) by Gauss–Newton–Marquardt-technique [40], which involves a series of linear minimizations (A4) as subproblems. The solution of (A7) is a vector $\boldsymbol{\beta}$ for which $\Phi(\boldsymbol{\beta})$ is stationary, i.e. the gradient $\nabla\Phi(\boldsymbol{\beta})$ is $\mathbf{0}$. Let $\boldsymbol{\beta}$ be an approximate solution. In order to get a better approximation $\boldsymbol{\beta} + \mathbf{d}$, we use the quadratic Taylor expansion for Φ ,

$$\Phi(\boldsymbol{\beta} + \mathbf{d}) = \Phi(\boldsymbol{\beta}) + \nabla\Phi(\boldsymbol{\beta}) \cdot \mathbf{d} + \frac{1}{2}\mathbf{d}^T \mathbf{S} \mathbf{d} + \mathcal{O}(\|\mathbf{d}\|^3), \quad (\text{A8})$$

in which $\mathbf{S} = \{s_{jj'}\}$ is the Hessian of $\Phi(\boldsymbol{\beta})$ given by

$$s_{jj'} = \frac{\partial^2 \Phi}{\partial \beta_j \partial \beta_{j'}} = 2 \sum_{i=1}^n w_i \left(\frac{\partial f_i}{\partial \beta_j} \frac{\partial f_i}{\partial \beta_{j'}} - (y_i - f_i) \frac{\partial^2 f_i}{\partial \beta_j \partial \beta_{j'}} \right). \quad (\text{A9})$$

Equation (A8) and the condition $\nabla\Phi(\boldsymbol{\beta}+\mathbf{d}) = \mathbf{0}$ lead to the equation

$$\nabla\Phi(\boldsymbol{\beta}) + \mathbf{S}\mathbf{d} + \mathcal{O}(\|\mathbf{d}\|^2) = \mathbf{0}. \quad (\text{A10})$$

Furthermore, neglecting higher-order terms in $\|\mathbf{d}\|$ we see that \mathbf{d} satisfies the p th-order linear system

$$\mathbf{S}\mathbf{d} = -\nabla\Phi(\boldsymbol{\beta}). \quad (\text{A11})$$

We shall neglect the term $\sum_{i=1}^n w_i(y_i - f_i)\partial^2 f_i/\partial\beta_j\partial\beta_{j'}$ in (A9). 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 partials, $\partial^2 f_i/\partial\beta_j\partial\beta_{j'}$, which express the nonlinearity of the model, are not too large. Therefore, (A11) is replaced by the normal-equation system

$$\mathbf{G}^T\mathbf{G}\mathbf{d} = \mathbf{G}^T(\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}), \quad (\text{A12})$$

where

$$\mathbf{G} = \{g_{ij}\} \quad \text{with} \quad g_{ij} = w_i^{1/2}\partial f_i/\partial\beta_j. \quad (\text{A13})$$

(A12) solves the simple least-squares problem

$$\min \{\|\mathbf{z} - \mathbf{E}\boldsymbol{\alpha} - \mathbf{G}\mathbf{d}\| : \mathbf{d} \in \mathbb{R}^q\}. \quad (\text{A14})$$

We cannot be sure that the correction vector \mathbf{d} obtained by solution of (A12) leads to a residual norm which is not greater than the previous iterate. When strong nonlinearities occur it is likely that the Gauss–Newton process diverges, unless special precautions are taken. Therefore, we use the modification of Marquardt [40] who proposed to replace (A12) by

$$(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{D})\mathbf{d} = \mathbf{G}^T(\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}), \quad (\text{A15})$$

where \mathbf{D} is a diagonal matrix with the same diagonal row as $\mathbf{G}^T\mathbf{G}$ and λ is an adjustable nonnegative scalar. Small values of λ lead to a Gauss–Newton-like procedure and large values to a gradient-like procedure. Intermediate values provide an interpolation between these methods. (A15) determines \mathbf{d} as a function of λ , $\mathbf{d} = \mathbf{d}(\lambda)$.

Over the years, this method, also called the Levenberg–Marquardt method [41], has undergone a number of refinements, adding more robustness to it. In earlier versions of PATFIT we used Marquardt’s original parameter selection algorithm for (A15): If the computed \mathbf{d} gives a larger value of Φ than the current best value, then increase λ by some factor ν . Repeat the solution of (A15) until a smaller Φ eventually is reached, and then decrease λ by the same factor ν . This simple technique puts no bounds on the step vector \mathbf{d} . Modern implementations of the Levenberg–Marquardt method replace (A14) by the quadratic programming problem

$$\min \left\{ \|\mathbf{z} - \mathbf{E}\boldsymbol{\alpha} - \mathbf{G}\mathbf{d}(\lambda)\| : \|\mathbf{D}^{1/2}\mathbf{d}(\lambda)\| \leq \Delta \right\}. \quad (\text{A16})$$

The scaled bound vector Δ is adjusted each time an outer iteration step begins and is decreased if the solution of (A16) does not provide a suitable correction. We have adopted this idea from the work of Moré [42], as

implemented in the software package MINPACK-1 [43] for unconstrained least-squares minimization. A subroutine from this package, LMPAR, performs the inner iteration loop by finding a value of λ that solves (A16) approximately. The optimal λ is saved for use as an initial estimate in the next outer iteration. (A15) is solved by QR decomposition technique. Details of this and of the iterative technique for estimating λ , as well as updating of the bound Δ , are given by Moré [42].

A crucial point in the semilinear least-squares procedure is the evaluation of $w_i^{1/2} \partial f_i / \partial \beta_j$ to form \mathbf{G} entering (A12) and defined by (A13). The vector of scaled function values $\{w_i^{1/2} f_i\}$ is $\mathbf{e} = \mathbf{E}\boldsymbol{\alpha}$. The term \mathbf{e} depends on $\boldsymbol{\beta}$ directly through \mathbf{E} and indirectly through $\boldsymbol{\alpha}$; hence

$$\frac{\partial \mathbf{e}}{\partial \beta_j} = \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha} + \mathbf{E} \frac{\partial \boldsymbol{\alpha}}{\partial \beta_j}. \quad (\text{A17})$$

To find $\partial \boldsymbol{\alpha} / \partial \beta_j$ we take the derivative of both members of (A5). This leads to

$$\mathbf{E}^T \mathbf{E} \frac{\partial \boldsymbol{\alpha}}{\partial \beta_j} = \frac{\partial \mathbf{E}^T}{\partial \beta_j} (\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}) - \mathbf{E}^T \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha} \quad (j=1, \dots, q). \quad (\text{A18})$$

The practical solution of (A18) exploits the triangular decomposition $\mathbf{E}^T \mathbf{E} = \mathbf{R}^T \mathbf{R}$, \mathbf{R} having already been computed in the QR factorization (A6) devised to solve (A5). For an ideal model the term in (A18) containing the residual vector $\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}$ 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 semilinear minimization of Φ : Start the outer iterations from a guessed value of $\boldsymbol{\beta}$, and select suitable initial values for λ and Δ . For each outer iteration, solve the linear subproblem (A4) for $\boldsymbol{\alpha}$ and calculate Φ . Compute the Jacobian elements $\partial \mathbf{e} / \partial \beta_j$ from (A18) and (A17), and form \mathbf{G} and \mathbf{D} . Then enter an inner loop to 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 Φ has proved to be stationary.

When implementing our semilinear algorithm, there is a practical difficulty in handling $\partial \mathbf{e} / \partial \beta_j$ in (A17). For each data point we must evaluate a $p \times q$ matrix of scalar derivatives. This means altogether $n \times p \times q$ values, and the magnitude of this number may tax the memory capacity of a small computer (this is most likely in the RESOLUTION model). For this reason we use a packed ("sparse-matrix") scheme for storing only the nonzero derivatives. Furthermore, when the storage demand exceeds some critical amount, a buffered storing of the derivatives on a temporary disk file is activated.

The numerical solution of many of the linear-algebraic and optimization subproblems in our algorithm is accomplished by software from the standard packages LINPACK [33] and MINPACK-1 [43].

Constraints

It is important to be able to impose constraints on the free variation of the model parameters. In principle such 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 and widths). Should an iteration step make such a parameter negative, a message “SIGN EXCURSION” is issued, and a new iterate is determined by halving the correction vector from the old one. As a rule, many SIGN EXCURSIONs mean an inadequate model parametrization. No sign checks are made on the linear parameters. For example, negative intensities may occur in a sensible way, when making a composite ACAR fitting analysis.

Incorporation of general equality constraints is possible in the framework of our least-squares procedure. We realized, however, that 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 procedures developed previously.

Let m linear constraints on the p components of $\boldsymbol{\alpha}$ be expressed as the matrix equation

$$\mathbf{H}\boldsymbol{\alpha} = \boldsymbol{\gamma}. \quad (\text{A19})$$

The matrix \mathbf{H} is an $m \times p$ constraint matrix with constant elements, and $\boldsymbol{\gamma}$ is a constant m -vector. We assume that the m constraints are independent and that $m \leq p$. It follows that the rank of \mathbf{H} is m , so we can construct a nonsingular matrix by selecting m independent columns from \mathbf{H} . A suitable permutation will bring these columns to the m first positions. This can be expressed in terms of a p th-order permutation matrix \mathbf{P} by

$$\mathbf{HP} = \mathbf{H}' =_m \begin{pmatrix} m & p-m \\ \mathbf{B} & \mathbf{N} \end{pmatrix}. \quad (\text{A20})$$

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”. As \mathbf{P} is orthogonal, $\mathbf{PP}^T = \mathbf{I}$, (A19) can be written

$$\mathbf{H}'\boldsymbol{\alpha}' = \boldsymbol{\gamma} \quad (\text{A21})$$

with

$$\boldsymbol{\alpha}' = \mathbf{P}^T\boldsymbol{\alpha}. \quad (\text{A22})$$

Eq. (A21) has the complete solution

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha}'_0 + \mathbf{Y}'\mathbf{t}, \quad (\text{A23})$$

where

$$\boldsymbol{\alpha}'_0 = \begin{matrix} m \\ p-m \end{matrix} \begin{pmatrix} \mathbf{B}^{-1}\boldsymbol{\gamma} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{Y}' = \begin{matrix} m \\ p-m \end{matrix} \begin{pmatrix} -\mathbf{B}^{-1}\mathbf{N} \\ \mathbf{I} \end{pmatrix}, \quad (\text{A24})$$

and $\mathbf{t} \in \mathcal{R}^{p-m}$. From (A22) we get the complete solution of (A19):

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_0 + \mathbf{Y}\mathbf{t}, \quad (\text{A25})$$

where

$$\boldsymbol{\alpha}_0 = \mathbf{P}\boldsymbol{\alpha}'_0 \quad (\text{A26})$$

and

$$\mathbf{Y} = \mathbf{P}\mathbf{Y}' \quad (\text{A27})$$

For the numerical computation it is practical to partition \mathbf{P} in column sections as follows:

$$\mathbf{P} = \begin{matrix} m & p-m \\ p \end{matrix} \begin{pmatrix} \mathbf{P}_1 & \mathbf{P}_2 \end{pmatrix}. \quad (\text{A28})$$

Then (A26) and (A27) are expressed as follows:

$$\boldsymbol{\alpha}_0 = \mathbf{P}_1\mathbf{B}^{-1}\boldsymbol{\gamma}, \quad \mathbf{Y} = (\mathbf{I} - \mathbf{P}_1\mathbf{B}^{-1}\mathbf{H})\mathbf{P}_2 \quad (\text{A29})$$

(the columns of \mathbf{Y} form a basis of the null space or kernel of \mathbf{H} , and we have $\mathbf{H}\mathbf{Y} = \mathbf{0}$.) When we insert (A25) into (A4), the constrained problem transforms to the following unconstrained one:

$$\min \{ \Phi(\mathbf{t}) \equiv \|\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}_0 - \mathbf{E}\mathbf{Y}\mathbf{t}\|^2 : \mathbf{t} \in \mathcal{R}^{p-m} \}. \quad (\text{A30})$$

This is solved as was (A4). One simply replaces $\boldsymbol{\alpha}$ by \mathbf{t} , \mathbf{z} by $\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}_0$ and \mathbf{E} by $\mathbf{E}\mathbf{Y}$. The result is that (A5) is replaced by

$$(\mathbf{E}\mathbf{Y})^T \mathbf{E}\mathbf{Y}\mathbf{t} = (\mathbf{E}\mathbf{Y})^T (\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}_0), \quad (\text{A31})$$

which is solved by now decomposing $\mathbf{E}\mathbf{Y} = \mathbf{Q}\mathbf{R}$. Moreover, by combining (A17) and (A25), we get

$$\frac{\partial \mathbf{e}}{\partial \beta_j} = \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha} + \mathbf{E}\mathbf{Y} \frac{\partial \mathbf{t}}{\partial \beta_j}, \quad (\text{A32})$$

and, by taking the derivative of (A31),

$$(\mathbf{E}\mathbf{Y})^T \mathbf{E}\mathbf{Y} \frac{\partial \mathbf{t}}{\partial \beta_j} = \frac{\partial (\mathbf{E}\mathbf{Y})^T}{\partial \beta_j} (\mathbf{z} - \mathbf{E}\boldsymbol{\alpha}) - (\mathbf{E}\mathbf{Y})^T \frac{\partial \mathbf{E}}{\partial \beta_j} \boldsymbol{\alpha}. \quad (\text{A33})$$

The remaining part of the least-squares procedure is unaffected by the constraints.

A.2 Statistics

We shall here address the question of statistical fluctuations to be expected in our parameter estimations, and also discuss some interpretations of the results.

Analysis

The following statistical analysis involves the parameter vector \mathbf{b} as obtained from the solution of the least-squares problem (A1), and also Φ_{\min} , the minimum value of Φ in (A1). Our primary goal is to estimate the “dispersion matrix”

$$\Sigma(\mathbf{b}) = \{\sigma_{jj'}\} \quad (\text{A34})$$

of \mathbf{b} (also called the covariance matrix). This matrix contains the parameter variances in the diagonal, while the off-diagonal entries (after normalization) contain the correlation coefficients. We shall use the symbol $E[\cdot]$ for the expected value, and $\text{Var}[\cdot]$ for variance.

In computing $\Sigma(\mathbf{b})$ we must imagine that an infinite ensemble of similar experimental spectra be given. Each spectrum, $\mathbf{y} = \{y_i\}$, gives rise to an estimate $\mathbf{b}(\mathbf{y})$ of the parameter vector through the solution of (A1). As the minimum value is attained where $\nabla\Phi(\mathbf{b}) = \mathbf{0}$, $\mathbf{b} = \mathbf{b}(\mathbf{y})$ also satisfies the equation system

$$\sum_{i=1}^n w_i (y_i - f_i(\mathbf{b})) \frac{\partial f_i}{\partial b_j} = 0 \quad (j = 1, \dots, k). \quad (\text{A35})$$

We introduce the “ensemble-mean” spectrum $\boldsymbol{\eta} = \{\eta_i\}$ defined by

$$\boldsymbol{\eta} = E[\mathbf{y}], \quad (\text{A36})$$

and the corresponding hypothetic estimate

$$\mathbf{b}_0 = \mathbf{b}(\boldsymbol{\eta}). \quad (\text{A37})$$

The analysis is approximate and relies on five assumptions:

1. Small fluctuations in y_i , hence $\text{Var}[y_i]$ small.
2. An ideal model, which means that

$$\eta_i = f_i(\mathbf{b}_0) \quad (i = 1, \dots, n). \quad (\text{A38})$$

3. “Statistical weighting”, which means that the weights w_i in (A1) are chosen as

$$w_i = 1/\text{Var}[y_i]. \quad (\text{A39})$$

4. The y_i are independent.
5. The y_i have a Gaussian distribution.

Assumption 1 is understood in the relative meaning. It is a fair approximation to our applications, provided the counts y_i are not too small.

Assumption 2 expresses that our model “explains” the observed data perfectly, apart from the inevitable statistical noise.

To fulfill Assumption 3, we need (an estimate of) $\text{Var}[y_i]$, see later. Statistical weighting is a convenient means to equalize the impact from the individual observations on the fit.

Assumption 4 is natural in most applications, and Assumption 5, which is needed only in the last part of the analysis, holds in the limiting case of Poisson statistics with large counts.

By Assumption 1 we infer, that for a single outcome $\mathbf{b} = \mathbf{b}(\mathbf{y})$, the deviation of the parameter vector

$$\Delta \mathbf{b} = \{\Delta b_j\} = \mathbf{b} - \mathbf{b}_0 \quad (\text{A40})$$

is likely to be small. This admits a linearized perturbation analysis based on (A35), and thus enables us to derive a linear relation between $\Delta \mathbf{b}$ and the deviation of the data vector,

$$\Delta \mathbf{y} = \{\Delta y_i\} = \mathbf{y} - \boldsymbol{\eta}. \quad (\text{A41})$$

Expanding from $\boldsymbol{\eta}$ and \mathbf{b}_0 and using Assumption 2, we get

$$y_i - f_i(\mathbf{b}) \approx \Delta y_i - \sum_{j=1}^k \frac{\partial f_i}{\partial b_j} \Delta b_j. \quad (\text{A42})$$

Let the perturbation domain in the parameter space be S . Then S includes \mathbf{b}_0 as well as \mathbf{b} , and moreover the partials $\partial f_i / \partial b_j$ are constant in S . If we normalize (A41) by introducing $\Delta \mathbf{u} = \{\Delta u_i\}$ given by

$$\Delta u_i = w_i^{1/2} \Delta y_i, \quad (\text{A43})$$

we find that (A35) is approximated by the homogeneous linear system

$$\mathbf{T}^T \mathbf{T} \Delta \mathbf{b} = \mathbf{T}^T \Delta \mathbf{u}, \quad (\text{A44})$$

where $\mathbf{T} = \{t_{ij}\}$, which we could call the differential design matrix, is defined by

$$t_{ij} = w_i^{1/2} \partial f_i / \partial b_j \quad (\text{A45})$$

(compare the normal-equation system (A5) for the semilinear subproblem.) The solution of (A44) is

$$\Delta \mathbf{b} = (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T \Delta \mathbf{u}. \quad (\text{A46})$$

Now, according to Assumptions 3 and 4, we have

$$\boldsymbol{\Sigma}(\Delta \mathbf{u}) = \mathbf{I}_n, \quad (\text{A47})$$

(\mathbf{I}_n = unit matrix of the order n), so finally

$$\boldsymbol{\Sigma}(\mathbf{b}) = (\mathbf{T}^T \mathbf{T})^{-1}. \quad (\text{A48})$$

Still in the linear approximation, we insert the solution (A46) in the expression (A42) for the residuals and thus approximate Φ_{\min} by a quadratic form

$$\Phi_{\min} \approx \Delta \mathbf{u}^T \mathbf{C} \Delta \mathbf{u} \quad (\text{A49})$$

with the coefficient matrix

$$\mathbf{C} = (\mathbf{I}_n - \mathbf{E})^2, \quad (\text{A50})$$

where

$$\mathbf{E} = \mathbf{T}(\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T. \quad (\text{A51})$$

We observe that $\mathbf{E}\mathbf{T} = \mathbf{T}$ and hence deduce that \mathbf{C} has $n - k$ eigenvalues equal to unity and k equal to zero. Consequently there exists an orthogonal transformation that transforms Φ_{\min} into a sum of squares of $n - k$ independent and standardized variates z_i :

$$\Phi_{\min} = \sum_{i=1}^{n-k} z_i^2, \quad \mathbb{E}[z_i] = 0, \quad \text{Var}[z_i] = 1. \quad (\text{A52})$$

At this point we invoke Assumption 5 to deduce that Φ_{\min} has a χ^2 -distribution with $n - k$ degrees of freedom (see also for example [36]).

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} . When constraints are included, the analysis still holds for a “basic subset” of k_{free} free parameter components. Thus in the distribution for Φ_{\min} we should replace k by k_{free} . To obtain $\Sigma(\mathbf{b})$, we must express the remaining components (deterministically) in terms of the free ones. In our case there are the linear constraints (A19), and instead of (A48) we get the dispersion matrix

$$\Sigma(\mathbf{b}) = \mathbf{Z}((\mathbf{T}\mathbf{Z})^T \mathbf{T}\mathbf{Z})^{-1} \mathbf{Z}^T, \quad (\text{A53})$$

where \mathbf{Z} is given by

$$\mathbf{Z} = \begin{matrix} & \begin{matrix} p-m & q \end{matrix} \\ \begin{matrix} p \\ q \end{matrix} & \begin{pmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \end{matrix}, \quad (\text{A54})$$

with the matrix \mathbf{Y} defined by (A29).

The practical computation of $\Sigma(\mathbf{b})$ can be done via the QR decomposition

$$\mathbf{T}\mathbf{Z} = \mathbf{Q}_2 \mathbf{R}_2, \quad (\text{A55})$$

leading to

$$\Sigma(\mathbf{b}) = \mathbf{Z}\mathbf{R}_2^{-1}(\mathbf{Z}\mathbf{R}_2^{-1})^T. \quad (\text{A56})$$

However, in ill-conditioned problems the diagonal row of \mathbf{R}_2 occasionally contains very small elements, which would render the evaluation of $\Sigma(\mathbf{b})$ by (A56) completely erratic. As explained in the LINPACK documentation [33] a variant of the QR decomposition with column scaling and pivoting admits a judicious discarding rule for insignificant elements in the \mathbf{R}_2 -diagonal. Hence we prefer to replace (A55) with

$$\mathbf{T}\mathbf{Z}\Sigma_0\mathbf{P}_2 = \mathbf{Q}_2\mathbf{R}_2, \quad (\text{A57})$$

where Σ_0 is a diagonal scaling matrix, \mathbf{P}_2 a permutation matrix, and the diagonal elements of \mathbf{R}_2 are in non-increasing order of magnitude. The entries in Σ_0 are chosen as the inverse Euclidean norms of the column vectors of $\mathbf{T}\mathbf{Z}$ and might be called “uncoupled standard deviations”. Instead of (A56) we obtain

$$\Sigma(\mathbf{b}) = \mathbf{Z}\Sigma_0\mathbf{P}_2\mathbf{R}_2^{-1}(\mathbf{Z}\Sigma_0\mathbf{P}_2\mathbf{R}_2^{-1})^T. \quad (\text{A58})$$

Only the upper “significant” part of \mathbf{R}_2 enters in this formula. The variance of possibly deleted parameters are estimated by their uncoupled standard deviations, while the covariance calculation for such parameters are abandoned.

If the parameter vector \mathbf{b} is transformed to another vector \mathbf{b}_1 before the output is presented, the dispersion matrix of \mathbf{b}_1 is then computed as

$$\Sigma(\mathbf{b}_1) = \mathbf{J}\Sigma(\mathbf{b})\mathbf{J}^T, \quad (\text{A59})$$

where

$$\mathbf{J} = d\mathbf{b}_1/d\mathbf{b} \quad (\text{A60})$$

is the Jacobian of the transformation.

In PATFIT-88 we use only simple transformations when passing from \mathbf{b} to \mathbf{b}_1 , or no transformation at all. Examples are lifetimes τ_j in ns instead of annihilation rates λ_j in channels⁻¹, and widths in FWHM instead of in standard deviations. These give rise to trivial diagonal elements in \mathbf{J} . On the other hand, the presentation of relative intensities I_j instead of absolute intensities J_j induces a diagonal block in the upper-left corner of \mathbf{J} with the (j, j') -entry $I_j(\delta_{jj'} - I_j)/J_j$.

When setting up the “statistical weighting” (A39) (Assumption 3) we do not know $\text{Var}[y_i]$ à priori. In the usual case the y_i are unnormalized counts with Poisson statistics, so that $\text{Var}[y_i] \approx y_i$, and (A39) may be replaced by

$$w_i = 1/y_i. \quad (\text{A61})$$

(In extreme cases y_i may become very small or even zero; when this happens we still make a least-squares fit, by replacing the zero counts by 1, but of course the preconditions for the statistical analysis no longer hold.)

In a POSITRONFIT analysis with source correction we use the preliminary model-predicted values f_i of y_i as estimates of $\text{Var}[y_i]$ in the second cycle of the fit:

$$w_i = 1/f_i. \quad (\text{A62})$$

This is slightly better than (A61), because f_i does not fluctuate as does y_i .

In ACARFIT the y_i may be normalized, in which case estimates of $\text{Var}[y_i]$ (or rather the standard deviations) must be given as an input table, see Section 3.4 and Appendix C.3.

Interpretations

We have seen, that under the five given assumptions, the minimum value Φ_{\min} of the weighted sum of squared errors Φ in (A1), has a χ^2 -distribution with f degrees of freedom, i.e.

$$\Phi_{\min} \in \chi^2(f), \quad (\text{A63})$$

where

$$f = n - k + m = n - k_{\text{free}}, \quad (\text{A64})$$

and $k_{\text{free}} = k - m$ is the effective number of free parameters in the estimation. We shall therefore speak of Φ_{min} as the “chi-square”. From the well-known properties of χ^2 we infer

$$\text{E}[\Phi_{\text{min}}] = f \quad (\text{A65})$$

and

$$\text{Var}[\Phi_{\text{min}}] = 2f. \quad (\text{A66})$$

As the result of a χ^2 -test, we compute the probability that the observed chi-square exceeds a χ^2 -distributed variable with f degrees of freedom. A value close to 100% indicates some systematic deviation from the assumed model, and we use the phrase “significance of imperfect model” for this probability.

We also compute the normalized chi-square

$$V = \Phi_{\text{min}}/f, \quad (\text{A67})$$

for which we have

$$\text{E}[V] = 1 \quad (\text{A68})$$

and

$$\text{Var}[V] = 2/f. \quad (\text{A69})$$

We shall call V the “Variance of the Fit”; for a good fit this quantity should be close to unity.

Another consequence of assuming a normal distribution of y_i is that also the parameter estimates will be distributed normally and their joint distribution completely determined by their dispersion matrix $\{\sigma_{jj'}\}$. An estimate of this matrix can be computed as shown in Section A.2.1. We prefer to operate with the set of standard deviations $\{\sigma_j\}$ and the estimated correlation matrix $\{\rho_{jj'}\}$, where $\sigma_j = \sigma_{jj'}^{1/2}$ and $\rho_{jj'} = \sigma_{jj'}/(\sigma_j\sigma_{j'})$.

The natural statistical interpretation of $\{\sigma_{jj'}\}$ or $\{\sigma_j, \rho_{jj'}\}$ is an estimate of the dispersion structure of the computed parameters in a series of repetitions of the spectrum recording under identical physical conditions. This interpretation is perhaps of limited interest, inasmuch as $\{\sigma_{jj'}\}$ might show a good deal of scatter in such a series.

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

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

$$= \left[\sum_{j=1}^k \left(\frac{\partial z}{\partial b_j} \right)^2 \sigma_j^2 + 2 \sum_{j=1}^{k-1} \sum_{j'=j+1}^k \frac{\partial z}{\partial b_j} \frac{\partial z}{\partial b_{j'}} \rho_{jj'} \sigma_j \sigma_{j'} \right]^{\frac{1}{2}}. \quad (\text{A70})$$

This result is established by analogy with the transformation formulas (A59) - (A60).

There is still another significance of the correlation matrix which might be useful in practice. Suppose an analysis of a given spectrum results in an estimated parameter vector $\mathbf{b} = \{b_j\}$, $j = 1, \dots, k$. One may ask: What happens to the remaining components if one of them, say b_1 , is forced to be shifted a small amount Δb_1 , and the analysis is repeated with the same spectrum? It can be shown that the other components will be shifted according to the formula

$$\Delta b_j = (\sigma_j / \sigma_1) \rho_{1j} \Delta b_1 \quad (j = 2, \dots, k). \quad (\text{A71})$$

(For a proof we consider $\Phi = \Phi(\mathbf{b})$ for fixed \mathbf{y} . In the linear approximation we have

$$\Phi(\mathbf{b} + \Delta \mathbf{b}) = \Phi(\mathbf{b}) + \nabla \Phi(\mathbf{b}) \cdot \Delta \mathbf{b} + \Delta \mathbf{b}^T \mathbf{T}^T \mathbf{T} \Delta \mathbf{b}, \quad (\text{A72})$$

\mathbf{T} being defined in (A45). We now fix Δb_1 , and seek the conditional minimum, when the other parameters vary. In the present case $\nabla \Phi(\mathbf{b}) = \mathbf{0}$, and we find

$$\frac{\partial \Phi}{\partial \Delta b_j} = 2 \sum_{j'=1}^k \sum_{i=1}^n t_{ij} t_{ij'} \Delta b_{j'} = 0, \quad j = 2, \dots, k, \quad (\text{A73})$$

which forms an equation system in $\Delta b_2, \dots, \Delta b_k$. From (A48), written in the form

$$\mathbf{T}^T \mathbf{T} \boldsymbol{\Sigma}(\mathbf{b}) = \mathbf{I}_k, \quad (\text{A74})$$

we deduce that (A71) is indeed the solution of (A73).

The formula (A71) 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, to the indeterminate nature of the minimization problem, or to strong nonlinearities.

B Details of Mathematical Models

Below we shall try to fill the gap between the rather brief description of the underlying mathematical models in the three fitting programs and the general semilinear least-squares theory in Appendix A.

B.1 POSITRONFIT

In Section 2.1 we gave a short presentation of the theoretical model used in POSITRONFIT. Writing formula (3) as

$$f(t) = \sum_{j=1}^{k_0} \sum_{p=1}^{k_g} \omega_p (a_j * G_p)(t) + B, \quad (\text{B1})$$

we must evaluate the convolution integral

$$(a_j * G_p)(t) = \int_{-\infty}^{\infty} a_j(v) G_p(t-v) dv, \quad (\text{B2})$$

where a_j and G_p were defined in (4) and (6), respectively. Henceforward, we prefer to describe the decay of a lifetime component in terms of the annihilation rate

$$\lambda_j = 1/\tau_j \quad (\text{B3})$$

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

$$(a_j * G_p)(t) = \frac{1}{2} A_j \varphi(t - T_0 - \Delta t_p, \lambda_j, \sigma_p). \quad (\text{B4})$$

The function φ acts as a building block for our lifetime spectral modelling and is defined by

$$\varphi(u, \lambda, \sigma) = \exp(-\lambda u + \frac{1}{2} \lambda^2 \sigma^2) \operatorname{erfc}\left(\frac{\lambda \sigma^2 - u}{\sqrt{2} \sigma}\right), \quad (\text{B5})$$

where erfc stands for the complementary error function

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x), \quad (\text{B6})$$

and erf in turn is defined by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt. \quad (\text{B7})$$

Inserting (B4) in (B1) we get

$$f(t) = \frac{1}{2} \sum_{j=1}^{k_0} A_j \sum_{p=1}^{k_g} \omega_p \varphi(t - T_0 - \Delta t_p, \lambda_j, \sigma_p) + B. \quad (\text{B8})$$

Finally, we compute the integrated model-predicted count f_i over channel i defined by equation (9) in Section 2.1. We use the identity

$$\int \varphi(u, \lambda, \sigma) du = -\frac{1}{\lambda} \left(\varphi(u, \lambda, \sigma) + \operatorname{erfc}\left(\frac{u}{\sqrt{2} \sigma}\right) \right) \quad (\text{B9})$$

and obtain

$$f_i = \sum_{j=1}^{k_0} F_{ij} + B = \sum_{j=1}^{k_0} \alpha_j f_{ij} + B, \quad (\text{B10})$$

where

$$\alpha_j = A_j/\lambda_j = A_j\tau_j \quad (\text{B11})$$

is the area or “absolute intensity” ascribed to lifetime component j ,

$$f_{ij} = \frac{1}{2} \sum_{p=1}^{k_g} \omega_p \{ \varphi(t_{ip}, \lambda_j, \sigma_p) - \varphi(t_{i+1,p}, \lambda_j, \sigma_p) \\ + \operatorname{erfc}(t_{ip}/(\sqrt{2}\sigma_p)) - \operatorname{erfc}(t_{i+1,p}/(\sqrt{2}\sigma_p)) \}, \quad (\text{B12})$$

and where we use the shorthand notation

$$t_{ip} = t_i - T_0 - \Delta t_p. \quad (\text{B13})$$

By now we have arrived at the model expression $f_i = f_i(\mathbf{b})$ entering the least-squares formulation (A1) of the fitting problem given in Appendix A. We also see that (B10) is semilinear as required by (A2); the parameter vector \mathbf{b} splits into a “linear” parameter $\boldsymbol{\alpha}$ and a “nonlinear” one $\boldsymbol{\beta}$ given by

$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{k_0}, B)^T \quad (\text{B14})$$

and

$$\boldsymbol{\beta} = (\lambda_1, \dots, \lambda_{k_0}, T_0)^T \quad (\text{B15})$$

Thus the semilinear fitting theory of Appendix A.1 applies. To perform the computations outlined there, we must evaluate the derivatives of f_{ij} in (B12) with respect to λ_j and T_0 ; this job is facilitated by the following two formulas:

$$\frac{\partial \varphi}{\partial u} = -\lambda \varphi(u, \lambda, \sigma) + \sqrt{\frac{2}{\pi}} \frac{1}{\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right), \quad (\text{B16})$$

and

$$\frac{\partial \varphi}{\partial \lambda} = (\lambda \sigma^2 - u) \varphi(u, \lambda, \sigma) - \sqrt{\frac{2}{\pi}} \sigma \exp\left(-\frac{u^2}{2\sigma^2}\right). \quad (\text{B17})$$

In Section 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 (B14) and (B15) are realized by deleting the corresponding components from $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$. This may apply to B , λ_j , and T_0 . Constraints of the type “fixed relative intensity” are not of this simple type because the relative intensities $\alpha_j / \sum \alpha_{j'}$ are not primary parameters. But obviously such constraints are expressible as linear constraints on the linear parameters α_j , i.e. relations of the form

$$\sum h_{ij} \alpha_j = \gamma_i, \quad (\text{B18})$$

where h_{ij} are known coefficients. The same holds true for constraints of the type “a linear combination of the relative intensities = 0”, as well as the constraint fixing the total area $\sum \alpha_j$ to an input value. The aggregate

of all the imposed linear constraints (B18) are comprised in the single matrix relation (A19), and the technical handling of the constraints now proceeds as in Appendix A.

Regarding the practical computation of (B5), that formula may from a numerical point of view be dangerous to use as it stands. The difficulty arises when $u \ll 0$. Then $\varphi(u, \lambda, \sigma)$ itself is small; nevertheless, the first factor of (B5) is large and may cause an overflow on the computer. At the same time, the second factor is very small and likely to underflow. A better alternative is to restate (B5) as

$$\varphi(u, \lambda, \sigma) = \exp\left(-\frac{u^2}{2\sigma^2}\right) \text{eerfc}\left(\frac{\lambda\sigma^2 - u}{\sqrt{2}\sigma}\right), \quad (\text{B19})$$

where, in the notation of Spanier and Oldham [44], we have introduced the normalized complementary error function

$$\text{eerfc}(x) = e^{x^2} \text{erfc}(x). \quad (\text{B20})$$

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

B.2 RESOLUTION

Although the basic model in RESOLUTION is the same as in POSITRON-FIT, there are certain differences regarding which parameters enter as fitting parameters. First, we notice that the background B in (B1), (B8), and (B10) should be omitted, because it has already been subtracted from the spectrum when the least-squares analysis begins. On the other hand, the shifts Δt_p and the widths σ_p in (B8) are fitting parameters in RESOLUTION. Hence (B14) and (B15) should be replaced by

$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{k_0})^T \quad (\text{B21})$$

and

$$\boldsymbol{\beta} = (\lambda_1, \dots, \lambda_{k_0}, \sigma_1, \dots, \sigma_{k_g}, \Delta t_1, \dots, \Delta t_{k_g}, T_0)^T. \quad (\text{B22})$$

In Section 2, we mentioned the types of constraints which could be imposed on the parameters in RESOLUTION. Some of the parameters of (B22) can be fixed and in that case should be deleted from $\boldsymbol{\beta}$. This may apply to λ_j , σ_p , and Δt_p . Notice that T_0 in RESOLUTION 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 (B16) and (B17), we shall also need the derivative of φ with respect to σ :

$$\frac{\partial \varphi}{\partial \sigma} = \lambda^2 \sigma \varphi(u, \lambda, \sigma) - \sqrt{\frac{2}{\pi}} \left(\lambda + \frac{u}{\sigma^2} \right) \exp\left(-\frac{u^2}{2\sigma^2}\right). \quad (\text{B23})$$

In RESOLUTION we compute shape parameters for the fitted resolution curve. This leads to nonlinear equations involving the φ function of (B5). We use a Newton–Raphson procedure for the numerical solution of these.

B.3 ACARFIT

In the following we give the formula apparatus necessary to handle the ACARFIT model, which was briefly presented in Section 2.3.

If we consider the model function in Equation (16), we see that this can be written in semilinear form

$$F(\theta) = f(\theta, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad (\text{B24})$$

As usual $\boldsymbol{\alpha}$ contains those parameters that enter the model linearly; they are

$$\{J_j^g\}_{j=1}^{k_g}, \{J_j^p\}_{j=1}^{k_p}, \quad (\text{B25})$$

i.e. the Gaussian and the parabolic (absolute) intensities. Note that each intensity refers to a pair of peaks (in the important special case where θ_j^g or θ_j^p is 0 the peak pair coalesce into a single ‘‘central’’ peak.) The vector $\boldsymbol{\beta}$, on the other hand, contains those parameters that enter (16) nonlinearly; they are

$$\{\theta_j^g\}_{j=1}^{k_g}, \{\sigma_j\}_{j=1}^{k_g}, \{\theta_j^p\}_{j=1}^{k_p}, \{\theta_j^c\}_{j=1}^{k_p}, \{\omega_j\}_{j=1}^{k_p}, \theta_0. \quad (\text{B26})$$

Any of the nonlinear parameters may be constrained to a fixed value and in that case should be deleted.

The mathematical treatment of the Gaussians in our model presents no particular difficulties; hence, we shall only give an account of the necessary manipulations of the parabolic terms. First, we have the normalization results

$$\int_{-\infty}^{\infty} P(\theta, \theta^c) d\theta = \int_{-\infty}^{\infty} p(\theta, \theta^c, \omega) d\theta = 1, \quad (\text{B27})$$

telling that the truncated parabola (19) is normalized and expressing the norm-preserving property of the convolution. In order to ease the calculus we give at this place a list of auxiliary formulas. First, we define a function h which is the integral of the Gaussian g defined in (17):

$$h(\theta, \sigma) = \int_0^\theta g(\theta', \sigma) d\theta' = \frac{1}{2} \operatorname{erf}\left(\frac{\theta}{\sqrt{2}\sigma}\right), \quad (\text{B28})$$

where erf is the error function (B7). Then, the following identities hold:

$$\int \theta g(\theta, \sigma) d\theta = -\sigma^2 g(\theta, \sigma), \quad (\text{B29})$$

$$\int \theta^2 g(\theta, \sigma) d\theta = \sigma^2 (h(\theta, \sigma) - \theta g(\theta, \sigma)), \quad (\text{B30})$$

$$\int h(\theta, \sigma) d\theta = \theta h(\theta, \sigma) + \sigma^2 g(\theta, \sigma), \quad (\text{B31})$$

$$\int \theta h(\theta, \sigma) d\theta = \frac{1}{2} ((\theta^2 - \sigma^2) h(\theta, \sigma) + \sigma^2 \theta g(\theta, \sigma)), \quad (\text{B32})$$

$$\int \theta^2 h(\theta, \sigma) d\theta = \frac{1}{3} (\theta^3 h(\theta, \sigma) + \sigma^2 (\theta^2 + 2\sigma^2) g(\theta, \sigma)), \quad (\text{B33})$$

$$\frac{\partial}{\partial \sigma} g(\theta, \sigma) = \frac{\theta^2 - \sigma^2}{\sigma^3} g(\theta, \sigma), \quad (\text{B34})$$

and

$$\frac{\partial}{\partial \sigma} h(\theta, \sigma) = -\frac{\theta}{\sigma} g(\theta, \sigma). \quad (\text{B35})$$

Using (B29) and (B30) it is possible to express p in (18) in terms of g and h :

$$\begin{aligned} p(\theta, \theta^c, \omega) &= \frac{3}{4(\theta^c)^3} [((\theta^c)^2 - \omega^2 - \theta^2)(h(\theta + \theta^c, \omega) - h(\theta - \theta^c, \omega)) \\ &\quad - \omega^2 \{(\theta - \theta^c)g(\theta + \theta^c, \omega) - (\theta + \theta^c)g(\theta - \theta^c, \omega)\}]. \end{aligned} \quad (\text{B36})$$

For computation of the area under the fitted curve, we shall need the integral q of the function p with respect to θ :

$$q(\theta, \theta^c, \omega) = \int_0^\theta p(\theta', \theta^c, \omega) d\theta'. \quad (\text{B37})$$

We observe that

$$p(\theta, \theta^c, \omega) = p_0(\theta, \theta^c, \omega) + p_0(\theta, -\theta^c, \omega), \quad (\text{B38})$$

where

$$\begin{aligned} p_0(\theta, \theta^c, \omega) &= \frac{3}{4(\theta^c)^3} [((\theta^c)^2 - \omega^2 - \theta^2)h(\theta + \theta^c, \omega) \\ &\quad - \omega^2(\theta - \theta^c)g(\theta + \theta^c, \omega)]. \end{aligned} \quad (\text{B39})$$

Then, by using (B31) - (B33) we get

$$q(\theta, \theta^c, \omega) = q_0(\theta, \theta^c, \omega) + q_0(\theta, -\theta^c, \omega), \quad (\text{B40})$$

where

$$\begin{aligned} q_0(\theta, \theta^c, \omega) &= \frac{1}{4(\theta^c)^3} [(2(\theta^c)^3 + 3((\theta^c)^2 - \omega^2)\theta - \theta^3)h(\theta + \theta^c, \omega) \\ &\quad + \omega^2(2((\theta^c)^2 - \omega^2) + \theta^c\theta - \theta^2)g(\theta + \theta^c, \omega)]. \end{aligned} \quad (\text{B41})$$

In the least-squares computations, we shall further need the three partial derivatives of $p(\theta, \theta^c, \omega)$. We find

$$\begin{aligned} \frac{\partial}{\partial \theta} p(\theta, \theta^c, \omega) &= \frac{3}{2(\theta^c)^3} [\theta(h(\theta - \theta^c, \omega) - h(\theta + \theta^c, \omega)) \\ &\quad + \omega^2(g(\theta - \theta^c, \omega) - g(\theta + \theta^c, \omega))], \end{aligned} \quad (\text{B42})$$

$$\begin{aligned} \frac{\partial}{\partial \theta^c} p(\theta, \theta^c, \omega) &= -\frac{3}{4(\theta^c)^2} \left[\left(1 - \frac{3(\omega^2 + \theta^2)}{(\theta^c)^2} \right) (h(\theta + \theta^c, \omega) - h(\theta - \theta^c, \omega)) \right. \\ &\quad \left. + \frac{3\omega^2}{\theta^c} \left\{ \left(1 + \frac{\theta}{\theta^c} \right) g(\theta - \theta^c, \omega) + \left(1 - \frac{\theta}{\theta^c} \right) g(\theta + \theta^c, \omega) \right\} \right], \end{aligned} \quad (\text{B43})$$

and

$$\begin{aligned} \frac{\partial}{\partial \omega} p(\theta, \theta^c, \omega) &= -\frac{3\omega}{2(\theta^c)^3} [h(\theta + \theta^c, \omega) - h(\theta - \theta^c, \omega) \\ &\quad - \theta^c(g(\theta - \theta^c, \omega) + g(\theta + \theta^c, \omega))]. \end{aligned} \quad (\text{B44})$$

The derivation of (B44) was facilitated by (B34) and (B35).

In ACARFIT we compute shape parameters for the fitted ACAR curve. This involves solution of nonlinear equations, and this task is accomplished by Newton-Raphson technique.

C Input data descriptions for the fitting programs

In the following we shall give prescriptions for Control Data input for the three fitting programs POSITRONFIT, RESOLUTION, and ACARFIT. Direct preparation of such input should be rather exceptional because this task will normally be handled by the editor programs POSEDIT, RESEDIT, and ACAREEDIT discussed in Section 3.

The input descriptions to be given are related to the FORTRAN language, in particular its input formatting facilities. We also refer to the identifier names in the FORTRAN programs. Some of the input are formatted, and in that case the input format will be explicitly stated. If a format is not stated, the input will be in “free-format” (“list-directed input” in FORTRAN terminology). In this case the data must be separated by blanks (or commas). A new record always means a new line.

In all three programs, the Control Data are partitioned into a number of data blocks, corresponding to the menus in the -EDIT programs. Each block must be initiated by a so-called block header. Only the first block header contains significant information for the fitting program (it is used to identify a new data set). It must appear exactly as follows:

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

in the case of POSITRONFIT, and similarly for the other two programs.

Note that the input prescriptions given in [1] is by now obsolete: The input for POSITRONFIT and RESOLUTION have been redesigned since 1981, and ACARFIT is a new program.

C.1 POSITRONFIT input

Sample Data

A sample Control Data Set (the one produced by POSEDIT as described in Section 3.2) is printed below.

```
POSITRONFIT DATA BLOCK: OUTPUT OPTIONS
11111
POSITRONFIT DATA BLOCK: SPECTRUM
  512
(F7.0/(10F7.0))
GR34800.DAT
34800
  1
34800 CU,RTNS II,61.4 DPA,T30, K=153                07:18:16 29-JUN-87
  75590.   34800.      3.      6.      11.      5.      7.      8.
    8.      9.      8.      5.      14.      7.      11.     10.
    6.      8.      5.      6.      8.      6.      8.      8.
    11.     7.      9.      10.     4.      4.      7.      3.
```

9.	3.	6.	5.	8.	8.	6.	10.
4.	16.	16.	2.	16.	10.	9.	6.
7.	10.	7.	6.	11.	11.	7.	7.
3.	2.	8.	12.	12.	4.	14.	4.
9.	5.	6.	5.	8.	6.	15.	13.
6.	15.	8.	9.	12.	4.	8.	7.
6.	7.	6.	8.	5.	6.	12.	15.
19.	35.	59.	87.	148.	223.	361.	633.
1165.	1889.	3259.	5846.	10072.	16435.	25267.	36083.
47387.	57966.	65200.	69771.	69971.	65937.	60585.	53345.
47128.	40655.	34425.	29200.	24704.	21012.	17870.	14902.
12750.	10805.	9202.	7822.	6591.	5600.	4760.	3979.
3529.	3024.	2502.	2201.	1821.	1610.	1450.	1151.
971.	829.	806.	686.	604.	486.	411.	380.
351.	281.	239.	225.	202.	173.	158.	158.
116.	104.	110.	92.	89.	77.	82.	50.
62.	67.	44.	59.	43.	42.	32.	30.
30.	33.	27.	19.	19.	26.	21.	22.
27.	33.	21.	24.	16.	19.	19.	25.
10.	18.	12.	18.	15.	12.	16.	18.
11.	13.	17.	15.	16.	9.	11.	12.
14.	12.	13.	14.	12.	14.	8.	14.
8.	12.	14.	10.	10.	11.	13.	11.
9.	9.	11.	8.	10.	8.	10.	11.
9.	11.	4.	15.	8.	8.	11.	7.
11.	10.	8.	11.	6.	10.	10.	10.
10.	5.	11.	13.	9.	15.	6.	6.
14.	9.	8.	9.	10.	13.	8.	8.
8.	10.	9.	4.	4.	11.	12.	12.
11.	13.	8.	9.	7.	6.	8.	2.
9.	12.	12.	5.	7.	6.	5.	8.
13.	7.	4.	7.	8.	7.	5.	14.
13.	8.	11.	13.	7.	5.	8.	15.
10.	13.	10.	12.	11.	7.	7.	4.
4.	4.	8.	10.	5.	13.	10.	9.
11.	7.	8.	5.	4.	15.	12.	8.
5.	8.	7.	4.	8.	11.	6.	13.
11.	10.	11.	9.	7.	12.	4.	11.
13.	4.	7.	9.	11.	8.	5.	9.
13.	10.	18.	6.	7.	5.	4.	6.
7.	8.	11.	9.	13.	2.	5.	5.
8.	3.	10.	11.	13.	6.	7.	11.
9.	6.	5.	9.	13.	3.	12.	12.
7.	8.	16.	4.	10.	12.	11.	7.
11.	4.	7.	9.	7.	10.	13.	10.
9.	12.	14.	9.	11.	10.	5.	9.
8.	7.	9.	7.	8.	17.	8.	10.
8.	8.	8.	6.	10.	6.	7.	11.
12.	9.	3.	5.	8.	4.	4.	7.
9.	5.	7.	8.	12.	12.	7.	10.
5.	8.	5.	8.	3.	5.	4.	7.
5.	8.	6.	9.	6.	5.	5.	12.
6.	8.	4.	10.	8.	5.	4.	8.
11.	14.	9.	7.	4.	11.	8.	5.
4.	9.	2.	10.	3.	8.	5.	14.
6.	12.	3.	8.	9.	9.	6.	11.
8.	4.	5.	9.	10.	10.	8.	4.
16.	10.	3.	8.	8.	7.	9.	11.
10.	7.	10.	7.	7.	13.	9.	12.
6.	8.	7.	6.	4.	9.	14.	6.

POSITRONFIT DATA BLOCK: CHANNEL RANGES. TIME SCALE. TIME-ZERO.

3

511

```

100
300
0.0311
G
105.7000
POSITRONFIT DATA BLOCK: RESOLUTION FUNCTION
3
0.2010 0.3050 0.2050
14.0000 14.0000 72.0000
0.0960 -0.0340 0.0000
POSITRONFIT DATA BLOCK: LIFETIMES AND INTENSITY CONSTRAINTS
4
GGFF
0.1500 0.2500 0.1800 2.0000
-2
0.5000 -0.5000 0.0000 0.0000
0.0000 0.0000 1.0000 -100.0000
POSITRONFIT DATA BLOCK: BACKGROUND CONSTRAINTS
1
400
500
POSITRONFIT DATA BLOCK: AREA CONSTRAINTS
1
10
510
POSITRONFIT DATA BLOCK: SOURCE CORRECTION
2
0.1800 0.3560
60.2400 39.7600
8.3000
1
3
GGF
0.1500 0.2500 2.0000
0

```

Output Options

Block 1 contains output options. Apart from the block header there is only one record. It contains five integer keys (IPRINT(I), I=1,5), format 5I1. Each key is either 0 or 1. The value 1 causes some output action to be taken, whereas 0 omits the same action. The five actions are:

1. Print Input Echo
2. Print outcome of each iteration
3. Print Residual Plot
4. Print Correlation Matrix
5. Produce Communication File

Regardless of the setting of these keys, POSITRONFIT always prints the Main Output.

Spectrum

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 the FORTRAN format (identifier FORM, format A80), enclosed in parentheses, e.g. (9F8.0). Note that I-formatting

must not be used, even though the spectrum contains integers. After this, two text records follow (format A80). In the first a name is stated (identifier SPCFIL) of a spectrum file. Even when $INSPEC = 1$ (see below) this name must be present, but is not used by the program in that case. In the other record an identification label (identifier SLABEL) of the spectrum is given. The next record contains an integer $INSPEC$ taking the value either 0 or 1. $INSPEC = 1$ means that the spectrum is an intrinsic part of the present Control Data file. In this case the next record should be a text line (format A80) with a description of the spectrum. The subsequent records are supposed to hold the spectrum (CT(I), I=1,NCH). On the other hand, $INSPEC = 0$ means that the spectrum is expected to reside on an external file with the name (SPCFIL) entered above. The program tries to open this file and scans it for a record whose start matches the identification label (SLABEL) up to the last nonblank character in this. After a successful match, a text line and the spectrum itself is 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 two records (after the block header) contain two channel numbers ICHA1 and ICHA2. These numbers are lower and upper bounds for the definition of a total area range. The next two records contain also two channel numbers ICHMIN and ICHMAX. These define in the same way the channel range which is used in the least-squares analysis. The next record contains CHATIM, which is the channel width (in ns). The last two records in this block deal with T_0 (time=0 channel number). First comes a constraint flag T0FLAG (format A1) 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 T0INIT which is the initial (guessed or fixed) value of T_0 .

Block 4 contains input for definition of the resolution function. The first record (after the block header) contains the number KG of Gaussian components in the resolution function. Each of the next three records contains KG numbers. In the first we have the full widths at half maxima of the Gaussians (in ns), (FWHM(J), J=1,KG), in the second their relative intensities (in percent), (OMEGA(J), J=1,KG), and in the third their peak displacements (in ns), (DT(J), J=1,KG).

Block 5 contains data for the lifetime components in the lifetime spectrum as well as constraints on their relative intensities. The first record (after the block header) holds the number K0 of lifetime components assumed in the model. Each of the next two records contains K0 data. In the first we have the constraint flags (G=guessed, F=fixed) for the lifetimes, (TAFLAG(J), J=1,K0), format 6A1. The other record contains the initial values (guessed or fixed) of the lifetimes, (TA0(J), J=1,K0). After this comes a record with an integer M telling the number and type of intensity constraints. |M| is the number of constraints. If $M = 0$ there is no further input data in this block. If $M > 0$, M of the relative intensities are fixed. In this case the next data item is a pair of records with the numbers (JL(L),

Ranges
Time scale
Time-zero

Resolution
Function

Lifetime
Components.

Constraints.

$L=1,M$) and $(RINTF(L), L=1,M)$; $JL(L)$ is the term number (the succession agreeing with the lifetimes on the previous record) associated with constraint number L , and $RINTF(L)$ is the corresponding fixed relative intensity (in percent). If $M < 0$, $|M|$ linear combinations of the intensities each equal zero. In this case $|M|$ records follow, each containing the $K0$ coefficients $(COMB(L,J), J=1,K0)$ to the intensities of one of the $|M|$ linear combinations.

Background

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

Area

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 called for. If $KAR = 2$ the area is fixed to an input value $SAREA0$, and this is entered on the next record.

Source Correction

Block 8 contains source correction data. The first record (after the block header) contains an integer $KCOR$ denoting the number of components in the source correction spectrum. $KCOR = 0$ means no source correction, in which case the present block contains no more data. The next record contains the lifetimes $(TACOR(J), J=1,KCOR)$, and the following the relative intensities $(RINTCR(J), J=1,KCOR)$ for the source correction terms. On the next record is $ALPHA$. The sum of $RINTCR(J)$ times $ALPHA$ equals the percentage of positrons that annihilate in the source (cf. Section 3.2, Menu 9). Then there follows a record with an integer $ISEC$. When $ISEC = 0$ the new iteration cycle after the source correction starts from lifetime guesses equal to the converged values from the first (correction-free) cycle. $ISEC = 1$ tells that the second cycle starts from new input data. These 2nd-cycle input data are now entered in exactly the same way as the 1st-cycle data in Block 5.

With the end of Block 8 the entire input for $POSITRONFIT$ is completed, and the program is ready to run. However, $POSITRONFIT$ accepts multiple data sets, starting all over with Block 1 again, etc. The data stream is stopped with an end-of-file.

C.2 RESOLUTION input

A sample Control Data Set (the one produced by RESEDIT as described in Section 3.3) is printed below. **Sample Data**

RESOLUTION DATA BLOCK: OUTPUT OPTIONS

11111

RESOLUTION DATA BLOCK: SPECTRUM

512

(F7.0/(10F7.0))

B:GR35050.DAT

35090

1

```
35090 CU-ANN.#6, CU-ANN.#5, CU-ANN.#3, K=156          13:01:35 09-JUL-88
 79196.  35090.   49.   51.   47.   45.   51.   63.
  44.    49.   43.   62.   60.   58.   46.   55.
  47.    57.   67.   46.   46.   53.   56.   38.
  61.    52.   60.   59.   47.   33.   53.   44.
  47.    55.   49.   52.   44.   44.   59.   48.
  48.    64.   60.   54.   56.   49.   59.   53.
  45.    46.   56.   49.   51.   53.   41.   51.
  47.    47.   62.   45.   54.   48.   50.   59.
  44.    46.   50.   56.   68.   50.   51.   50.
  56.    43.   62.   41.   48.   47.   59.   66.
  56.    53.   65.   56.   57.   58.   55.   58.
  68.    80.  109.  140.  254.  426.  676. 1239.
 2146.   3810.  6857. 12207. 21374. 35351. 56427. 83199.
115374. 147328. 176373. 195642. 203719. 199834. 185443. 164860.
141862. 117458.  96273.  77445.  61588. 48313. 38308. 30146.
23663. 18672. 14889. 11527.  9306.  7668.  6193.  5128.
 4176.   3529.  2805.  2468.  2049.  1888.  1675.  1357.
 1167.   1091.  1101.   875.   819.   701.   689.   643.
  554.   541.   451.   442.   406.   396.   333.   312.
  322.   314.   293.   262.   224.   220.   192.   197.
  193.   159.   173.   155.   145.   129.   126.   135.
  125.   110.   114.   118.   106.   97.   81.   105.
  96.   85.   84.   82.   83.   72.   81.   60.
  75.   73.   70.   83.   71.   62.   65.   67.
  67.   56.   60.   52.   64.   55.   56.   54.
  57.   54.   53.   64.   52.   54.   53.   56.
  60.   51.   55.   44.   60.   60.   46.   61.
  59.   47.   72.   62.   51.   51.   55.   58.
  61.   62.   41.   49.   50.   42.   50.   43.
  54.   42.   58.   55.   50.   50.   52.   43.
  54.   69.   42.   51.   50.   58.   46.   51.
  51.   42.   51.   48.   60.   46.   48.   49.
  42.   57.   56.   48.   55.   53.   52.   54.
  42.   64.   48.   44.   44.   53.   53.   40.
  39.   48.   51.   67.   57.   49.   48.   28.
  58.   51.   48.   43.   44.   47.   47.   50.
  55.   54.   56.   47.   61.   42.   47.   32.
  52.   51.   51.   55.   52.   49.   49.   52.
  68.   56.   51.   56.   54.   50.   46.   53.
  51.   38.   54.   61.   49.   33.   49.   41.
  35.   48.   58.   49.   47.   49.   58.   49.
  55.   51.   52.   60.   46.   47.   59.   55.
  66.   50.   52.   54.   55.   57.   52.   57.
  60.   48.   48.   55.   55.   56.   50.   47.
  50.   32.   57.   54.   51.   46.   48.   48.
  50.   57.   51.   56.   52.   56.   52.   51.
  55.   54.   57.   53.   44.   44.   47.   51.
  51.   42.   49.   47.   48.   44.   40.   47.
```

```

50.    51.    48.    50.    56.    53.    59.    50.
45.    51.    45.    49.    50.    41.    50.    54.
47.    57.    53.    50.    55.    47.    36.    60.
56.    52.    41.    62.    37.    56.    50.    50.
61.    54.    55.    60.    61.    37.    48.    57.
42.    41.    55.    51.    52.    53.    50.    33.
53.    37.    55.    36.    39.    53.    52.    45.
53.    46.    52.    47.    48.    53.    40.    56.
57.    49.    47.    60.    58.    48.    45.    56.
48.    47.    31.    51.    54.    65.    54.    52.
48.    69.    49.    41.    53.    41.    49.    52.
43.    51.    54.    44.    52.    53.    38.    46.
65.    62.    53.    51.    48.    48.    53.    45.
43.    49.    64.    38.    35.    38.    42.    48.
63.    44.    48.    43.    47.    44.    51.    46.
49.    46.    50.    28.    49.    49.    47.    69.
RESOLUTION DATA BLOCK: CHANNEL RANGES. TIME SCALE. TIME-ZERO.
  3
  511
  95
  250
  0.0311
  107.0000
RESOLUTION DATA BLOCK: RESOLUTION FUNCTION
  3
GGG
  0.2000    0.3000    0.2200
  12.0000   13.0000   75.0000
GGF
  0.1000   -0.0500    0.0000
RESOLUTION DATA BLOCK: LIFETIME AND INTENSITY CONSTRAINTS
  4
GGFF
  0.1100    0.4500    0.1800    2.0000
  1
  3
  5.0000
RESOLUTION DATA BLOCK: BACKGROUND RANGE
  411
  512

```

Output Options

Block 1 contains output options. It is identical to the corresponding block in the POSITRONFIT input (but of course the name RESOLUTION must appear in the block header).

Spectrum

Block 2 contains the spectrum. It is identical to the corresponding block in the POSITRONFIT input.

Ranges

Time scale

Time-zero

Block 3 contains information related to the measuring system. The first two records (after the block header) contain two channel numbers ICHA1 and ICHA2. These numbers are lower and upper bounds for the definition of a total area range. The next two records contain also two channel numbers ICHMIN and ICHMAX. These define in the same way the channel range which is used in the least-squares analysis. The next record contains CHATIM, which is the channel width (in ns). The last record in this block contains T0INIT which is the initial (guessed) value of T_0 .

Resolution

Block 4 contains input for definition and initialization of the resolution

function. The first record (after the block header) contains the number KG of Gaussian components in the resolution function. Each of the next two records contains KG data. In the first we have the constraint flags (G=guessed, F=fixed) for the Gaussian widths, (FWFLAG(J),J=1,KG), format 6A1. The second contains the initial values (guessed or fixed) of the full widths at half maxima of the Gaussians (in ns), (FWHM0(J), J=1,KG). The next record contains the KG Gaussian peak intensities in percent, (OMEGA(J),J=1,KG). The last two records in the block contain again KG data each. First, we have the constraint flags (G=guessed, F=fixed) for the Gaussian shifts, (DTFLAG(J),J=1,KG), format 6A1; notice that not all the shifts can be free. Next, we have the initial (guessed or fixed) peak displacements (in ns), (DT0(J), J=1,KG).

Function

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

**Lifetime
Components.
Constraints.**

Block 6 contains data related to the background. Apart from the block header this block contains two records. In these are given the channel numbers ICHBG1 and ICHBG2 between which the background shall be determined.

Background

This completes the RESOLUTION input. Multiple data set can be handled in the same way as in POSITRONFIT. It is even possible (for both POSITRONFIT and RESOLUTION) to process a "mixed" job stream with Control Data Sets for both programs; only the data for the relevant program are interpreted.

C.3 ACARFIT input

A sample Control Data Set (the one produced by ACAREEDIT as described in Section 3.4) is printed below.

Sample Data

```
ACARFIT DATA BLOCK: OUTPUT OPTIONS
11111
ACARFIT DATA BLOCK: SPECTRUM, ANGLE-ZERO, ANGLE SCALE.
110
    55
(12F6.2)
(12F6.0)
(12F6.0)
01591.DAT
01591
    1
01591 COPPER,5N,CW 70%                                06-AUG-86
-17.24  -15.64  -14.23  -13.00  -11.90  -10.91  -10.05
-9.32   -8.70   -8.09   -7.47   -6.92   -6.37   -5.82
-5.26   -4.71   -4.16   -3.67   -3.17   -2.74   -2.31
-1.95   -1.58   -1.21   -0.84   -0.47   -0.10   0.27
0.63    1.00    1.37    1.74    2.11    2.48    2.85
3.28    3.71    4.20    4.69    5.24    5.79    6.35
6.90    7.45    8.01    8.62    9.23    9.85    10.59
11.45   12.43   13.53   14.76   16.18   17.77
```

```

369.0   586.0   847.0   1274.0   1653.0   2087.0   2521.0
3000.0  3405.0  3907.0  4375.0  4960.0  5669.0  6482.0
7371.0  8662.0  9959.0  11066.0  12177.0  12923.0  13550.0
14224.0 14456.0 14918.0 15195.0 15270.0 15276.0 15322.0
15220.0 14988.0 14899.0 14425.0 13827.0 13359.0 12689.0
11918.0 10948.0 9790.0 8684.0 7391.0 6451.0 5632.0
4946.0 4388.0 3936.0 3453.0 2987.0 2623.0 2246.0
1854.0 1428.0 1054.0 702.0 477.0 303.0
11.0   13.0   16.0   19.0   22.0   24.0   27.0
29.0   31.0   33.0   35.0   37.0   40.0   43.0
45.0   49.0   53.0   56.0   58.0   60.0   61.0
63.0   63.0   64.0   65.0   65.0   65.0   65.0
65.0   65.0   64.0   63.0   62.0   61.0   60.0
58.0   55.0   52.0   49.0   45.0   42.0   40.0
37.0   35.0   33.0   31.0   29.0   27.0   25.0
23.0   20.0   17.0   14.0   12.0   10.0
G
0.0000
ACARFIT DATA BLOCK: FIT AND NORMALIZATION RANGES
-17.0000 17.0000
13.0000
ACARFIT DATA BLOCK: COMPONENTS AND CONSTRAINTS
5
PARA FFF
0.0000 5.5010 2.5191
GAUS FF
0.0000 16.3685
GAUS FF
0.0000 4.7605
GAUS FG
0.0000 8.0000
PARA GGG
6.0000 2.0000 3.0000
-2
2 1 2.7286 3 27.4036
2 1 -75.3250 2 27.4036
ACARFIT DATA BLOCK: BACKGROUND
0.0000

```

Output Options

Block 1 contains output options. It is identical with the corresponding block in the POSITRONFIT input (but of course the name ACARFIT must appear in the block header).

Spectrum

Block 2 deals with the spectrum and a few related quantities. The first record (after the block header) contains three integer keys (MODES(I), I=1,3), format 3I1. Each key is either 0 or 1. The value 1 for key No. i causes a corresponding action to be taken. The actions are:

1. Include an angular table in the input.
2. Include a standard deviation table in the input.
3. Give number of data points in spectrum, NPT, just before the spectrum itself.

If the key is 0 the action is skipped. Therefore the following input depends on the setting of these keys:

The first record to follow contains the number of points NPT. It is skipped if $\text{MODES}(3) = 1$.

Next there are three text records, format A80. They contain FORTRAN formats for the angular table, for the spectrum itself, and for the standard deviation table. They are enclosed in parentheses e.g. (12F6.2). The identifier names are FORMX, FORMY, and FORMS. The FORMX record is skipped if $\text{MODES}(1) = 0$. The FORMY record is mandatory. The FORMS record is skipped if $\text{MODES}(2) = 0$. I-formatting is not allowed in any of these format records.

After the format records two text records follow, format A80. In the first a name (identifier SPCFIL) of a spectrum file is stated. Even when $\text{INSPEC} = 1$ (see below) this name must be present, but in this case is not used by the program. In the other record an identification label (identifier SLABEL) of the spectrum is given. The next record contains an integer INSPEC taking the value either 0 or 1. $\text{INSPEC} = 1$ means that the spectrum is an intrinsic part of the present Control Data file. In this case, the next record should be a text line (format A80) with a description of the spectrum. If $\text{MODES}(3) = 1$ there will next be a record with the number of points, NPT. At this place the input may contain three tables each with NPT values. First we have the angular table, then the spectrum itself, and finally the standard deviation table. The angular table is omitted if $\text{MODES}(1) = 0$, and the standard deviation table is omitted if $\text{MODES}(2) = 0$. The spectrum is always present. If no angular table is given ($\text{MODES}(1) = 0$), the spectrum is supposed to be defined on an equidistant angular grid (“channels”).

On the other hand $\text{INSPEC} = 0$ means that the spectrum is expected to reside on an external file with the name (SPCFIL) entered above. ACARFIT tries to open this file and scans it for a record whose start matches the identification label (SLABEL) up to the last nonblank character in this. After a successful match, a heading text line, the number of points (if $\text{MODES}(3) = 1$), the angular table (if $\text{MODES}(1) = 1$), the spectrum, and the standard deviation table (if $\text{MODES}(2) = 1$), are read from the subsequent records, all in exactly the same way as in the case $\text{INSPEC} = 1$.

The remaining records in this block contain some auxiliary quantities related to the spectrum. First come two records dealing with θ_0 (the angle-zero). In the first one is a constraint flag T0FLAG (format A1) being either a G or an F. G stands for guessed (i.e. free) θ_0 , F for fixed θ_0 . The other record contains T00 which is the initial (guessed or fixed) value of θ_0 . The unit for T00 is mrad if $\text{MODES}(1) = 1$ but “channels” if $\text{MODES}(1) = 0$. The last record in the block is present only if $\text{MODES}(1) = 0$. It contains the number CHARAD, which is the separation of the channels in mrad. Together with the zero T00 it defines a unique mrad scale on the equidistant spectrum grid.

Angle-zero

Angular scale

Ranges

Block 3 contains data for the definition of angular ranges. There are two records (after the block header). The first one contains the numbers TFMIN and TFMAX; they cause the angular range for the least-squares analysis to be confined to the interval from TFMIN to TFMAX. The other record contains the number TNMAX. It defines the normalization range, extending from $-TNMAX$ to $+TNMAX$. All the quantities in this block are given in mrad.

Components and Constraints

Block 4 contains information on the various components in the model, as well as possible constraints on their relative intensities. The first record (after the block header) holds the total number KT of components (Gaussians and folded parabolas). Next come KT record pairs, one for each component. The first record in such a pair contains a text descriptor for the component. The text in position 1–4 should be either GAUS or PARA according to which component type is intended. If GAUS is stated, positions 6–7 in the same record contain two constraint flags (G for guessed, F for fixed), defining the fitting status for the position and width, respectively, of the Gaussian. If, on the other hand, PARA is stated, positions 6–8 in the record contain three constraint flags. They define likewise the constraint status for the position, cutoff angle, and broadening, respectively, of a folded-parabola term. The other record in the pair contains initial values (guessed or fixed) corresponding to the constraint flags: In the GAUS case there are two numbers: position and width; in the PARA case three numbers: position, cutoff-angle, and broadening. (The widths and broadenings are defined by their full width at half maximum. All angles are in mrad.) After these KT record pairs comes a record with an integer M telling the number and type of intensity constraints. $|M|$ is the number of constraints. If $M = 0$ there is no further input data in this block.

If $M > 0$, M of the relative intensities are fixed. In this case there follows M constraint records; record # L contains two numbers JTL(L) and RINTF(L). The first one, JTL(L), is the term number associated with constraint #L (the succession being defined previously by the input sequence of the KT record pairs), and RINTF(L) is the corresponding fixed intensity (in percent).

If $M < 0$, $|M|$ linear combinations of the intensities each equal zero. In this case $|M|$ records follow; record # L describes linear-combination constraint # L in terms of its nonzero coefficients. Specifically, the record is entered as NEL(L), (JTCO(IEL,L), CON(IEL,L), IEL=1,NEL(L)). Here NEL(

D Subroutines in PATFIT-88

In this Appendix we list, in tabular form, the FORTRAN subprograms making up the six PATFIT-88 programs, together with a short description of their tasks. Also the subroutines in the collection MATHPACK, which serves as a common subroutine library for the three fitting programs, are listed, as well as the nonstandard FORTRAN subroutines making up the EXTRA file. PATGRAPH is not discussed in this Appendix.

D.1 POSITRONFIT

List of POSITRONFIT subprograms:

Name	Task
MAIN	POSITRONFIT driver program.
CORRCT	Computes the source-correction spectrum.
DATAIO	Reads input data for POSITRONFIT. Also prints Input Echo and produces part of the Communication File.
DISP	Produces and prints a residual plot for POSITRONFIT.
JACOBI	Computes the Jacobian for transforming internal fitting parameters to output parameters in POSITRONFIT.
LINCON	Constructs the linear constraints matrix for POSITRONFIT.
MODEL	Evaluates model terms in POSITRONFIT and prepares for derivatives.
PACDER	Establishes pointer arrays with packed derivative locations for the POSITRONFIT model.
POSIT	Master subroutine of POSITRONFIT.
RESULT	Prints the main output from POSITRONFIT. Also produces part of the Communication File.

D.2 RESOLUTION

List of RESOLUTION subprograms:

Name	Task
MAIN	RESOLUTION driver program.

DATAIO	Reads input data for RESOLUTION. Also prints Input Echo and produces part of the Communication File.
DISP	Produces and prints a residual plot for RESOLUTION.
JACOBI	Computes the Jacobian for transforming internal fitting parameters to output parameters in RESOLUTION.
LINCON	Constructs the linear constraints matrix for RESOLUTION.
MODEL	Evaluates model terms in RESOLUTION and prepares for derivatives.
PACDER	Establishes pointer arrays with packed derivative locations for the RESOLUTION model.
PEAK	Computes shape parameters for the resolution curve.
RESOL	Master subroutine of RESOLUTION.
RESULT	Prints the main output from RESOLUTION. Also produces part of the Communication File.

D.3 ACARFIT

List of ACARFIT subprograms:

Name	Task
MAIN	ACARFIT driver program.
ACARF	Master subroutine of ACARFIT.
DATAIO	Reads input data for ACARFIT. Also prints Input Echo and produces part of the Communication File.
DISP	Produces and prints a residual plot for ACARFIT.
JACOBI	Computes the Jacobian for transforming internal fitting parameters to output parameters in ACARFIT.
GA	Computes a Gaussian probability density.
HG	Computes the integral of a Gaussian probability density.
LINCON	Constructs the linear constraints matrix for ACARFIT.
MODEL	Evaluates model terms in ACARFIT and prepares for derivatives.
NIT	Returns number of parameters (2 or 3) for an ACAR term.
PACDER	Establishes pointer arrays with packed derivative locations for the ACARFIT model.
PEAK	Computes shape parameters for the ACAR curve.
RESULT	Prints the main output from ACARFIT. Also produces part of the Communication File.

D.4 POSEDIT

List of POSEDIT subprograms:

Name	Task
POSEDI	POSEDIT driver program.
AREA	Edits POSITRONFIT area constraints.
BACKGR	Edits POSITRONFIT background constraints.
CAPS	Converts lower-case letters to upper case.
CHANCK	Checks consistency of channel limits for POSITRONFIT.
DATFIL	Writes POSITRONFIT data to file.
EFFLEN	Finds the effective length of a string ignoring trailing blanks.
FCOPY	Line-by-line copying between files.
FILDAT	Reads POSITRONFIT data from file.
KEYIN	Reads and interpretes an input line from the keyboard.
LIFINT	Edits initial lifetime and intensity-constraint data for POSITRONFIT.
MMLIST	Shows the main menu list in POSEDIT.
NEWDAT	Exports a Control File for POSITRONFIT input.
OLDDAT	Imports a Control File for POSITRONFIT input.
OUTOPT	Edits output options for fitting program.
RASCAZ	Edits area and fit ranges, time scale, and time-zero, for lifetime spectra.
READSP	Edits a lifetime spectrum.
RESFUN	Edits resolution function for POSITRONFIT.
SIZECK	Checks the problem size for POSITRONFIT.
SOCORR	Edits source-correction data.

Finally, a BLOCK DATA subprogram initializes FORTRAN unit numbers for files, FORTRAN carriage control character, POSEDIT escape character, name of restart file, and menu texts.

D.5 RESEDIT

List of RESEDIT subprograms:

Name	Task
RESEDI	RESEDIT driver program.
BACKGR	Edits RESOLUTION background constraints.
CAPS	Converts lower-case letters to upper case.
CHANCK	Checks consistency of channel limits for RESOLUTION.
DATFIL	Writes RESOLUTION data to file.
EFFLEN	Finds the effective length of a string ignoring trailing blanks.
FCOPY	Line-by-line copying between files.
FILDAT	Reads RESOLUTION data from file.
KEYIN	Reads and interpretes an input line from the keyboard.
LIFINT	Edits initial lifetime and intensity-constraint data for RESOLUTION.
MMLIST	Shows the main menu list in RESEDIT.
NEWDAT	Exports a Control File for RESOLUTION input.
OLDDAT	Imports a Control File for RESOLUTION input.
OUTOPT	Edits output options for fitting program.
RASCAZ	Edits area and fit ranges, time scale, and time-zero, for lifetime spectra.
READSP	Edits a lifetime spectrum.
RESFUN	Edits resolution function for RESOLUTION.
SIZECK	Checks the problem size for RESOLUTION.

Finally, a BLOCK DATA subprogram initializes FORTRAN unit numbers for files, FORTRAN carriage control character, RESEDIT escape character, name of restart file, and menu texts.

D.6 ACAREEDIT

List of ACAREEDIT subprograms:

Name	Task
ACARED	ACAREEDIT driver program.

BACKGR	Edits ACARFIT background subtraction.
CAPS	Converts lower-case letters to upper case.
DATFIL	Writes ACARFIT data to file.
DELLIC	Updates linear-combination pointers after deleting an ACAR component.
EFFLEN	Finds the effective length of a string ignoring trailing blanks.
FCOPY	Line-by-line copying between files.
FILDAT	Reads ACARFIT data from file.
INSLIC	Updates linear-combination pointers after inserting an ACAR component.
KEYIN	Reads and interpretes an input line from the keyboard.
MAP	Prints a map of ACAR components.
MMLIST	Shows the main menu list in ACAREEDIT.
NEWDAT	Exports a Control File for ACARFIT input.
NIT	Returns number of parameters (2 or 3) for an ACAR term.
OLDDAT	Imports a Control File for ACARFIT input.
OUTOPT	Edits output options for fitting program.
RANGES	Edits angular ranges for ACARFIT.
READSP	Edits an ACAR spectrum.
SIZECK	Checks the problem size for ACARFIT.
TERMIN	Edits ACAR components and constraints.

Finally, a BLOCK DATA subprogram initializes FORTRAN unit numbers for files, FORTRAN carriage control character, ACAREEDIT escape character, name of restart file, and menu texts.

D.7 MATHPACK

MATHPACK is a collection of mathematical and other general-purpose subprograms. Its serves as a library module for the three fitting programs. MATHPACK is partly made up of Risø software (including routines from “Risø Computer Library”), and partly of routines from the two collections LINPACK [33] for solving problems in linear algebra and MINPACK-1 [43] for nonlinear unconstrained optimization.

List of MATHPACK routines originating from Risø:

Name	Task
------	------

BUFREA	Makes buffered file reading of part of packed derivative matrices.
BUFWRI	Makes buffered file writing of part of packed derivative matrices.
CHABOO	Converts boolean constraint flag to character.
CONMUL	Multiplies constraint matrix by a column.
CTRMUL	Multiplies transposed constraint matrix by a column.
DERIVA	Computes the derivatives of the model function with respect to the nonlinear parameters.
EERFC	Computes the normalized complementary error function.
EFFLEN	Finds the effective length of a string ignoring trailing blanks.
ELIMIN	Reduces linear constraint matrix to canonical form.
ERFC	Computes the complementary error function.
FCOPYL	Line-by-line copying between files, with allowance for a carriage-control character.
FITNON	Computes nonlinear parameter vector by QR technique.
PCHISQ	Probability routine for chi-square test.
SEMIFT	Executes the semilinear least-squares procedure.
STATIS	Performs the statistical analysis.
TIMDAT	Formats the current time and date.

List of MATHPACK routines originating from LINPACK:

Name	Task
DAXPY	Updates a vector by a constant times a vector.
DDOT	Forms the scalar product of two vectors.
DPOSJ	Solves triangularized positive-definite equation system.
DSCAL	Scales a vector by a constant.
DTRDI	Computes the inverse of a triangular matrix.

List of MATHPACK routines originating from MINPACK-1:

Name	Task
DPMPAR	Provides double-precision machine parameters (locally adapted).
ENORM	Computes Euclidean norm of a vector.
LMPAR	Computes the Levenberg-Marquardt parameter.
QRFAC	Computes a QR factorization of a matrix.
QRSOLV	Computes least-squares solution from QR factorization.

D.8 EXTRA

Each of the “EXTRA”-files contains four nonstandard subroutines written for a specific FORTRAN compiler (see Section 5).

List of EXTRA subroutines:

Name	Task
DATJOB	Returns current date.
MCLOCK	Timing routine.
OPENCL	Opens new files without FORTRAN carriage control (acomodates for a DEC peculiarity).
STONUM	Tells how many bytes corresponds to one numerical storage unit when using unformatted disk I/O under direct access.

L) is the number of nonzero coefficients in constraint # L, JTCO(IEL,L) is the component # corresponding to nonzero coefficient # IEL, and CON(IEL,L) is the value of the corresponding coefficient.

Block 5 contains the background. Apart from the block header there is only one record containing the background BACKGR. A value of zero means that the background is already subtracted.

Background

This completes the ACARFIT input. Multiple data sets can be handled in the same way as in POSITRONFIT.

References

- [1] P. Kirkegaard and M. Eldrup, "POSITRONFIT: A versatile program for analysing positron lifetime spectra," *Computer Phys. Commun.*, vol. 3, pp. 240–255, 1972.
- [2] P. Kirkegaard and M. Eldrup, "POSITRONFIT EXTENDED: A new version of a program for analysing positron lifetime spectra," *Computer Phys. Commun.*, vol. 7, pp. 401–409, 1974.
- [3] P. Kirkegaard and O. E. Mogensen, "PAACFIT: A program for analysing positron annihilation angular correlation spectra," Tech. Rep. Risø-M-1615, Risø National Laboratory, DK-4000 Roskilde, Denmark, 1973.
- [4] P. Kirkegaard and O. E. Mogensen, "PARAFIT: A program for analysing positron annihilation angular correlation curves," in *Proc. 4'th Int. Conf. on Positron Annihilation*, (Helsingør, Denmark), 1976. Paper H12.
- [5] P. Kirkegaard, M. Eldrup, O. E. Mogensen, and N. J. Pedersen, "Program system for analysing positron lifetime spectra and angular correlation curves," *Computer Phys. Commun.*, vol. 23, pp. 307–335, 1981.
- [6] W. K. Warburton, "DBLCON: A version of Positronfit with non-gaussian prompt for analysing positron lifetime spectra," *Comput. Phys. Commun.*, 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," *Comput. Phys. Commun.*, vol. 15, pp. 97–105, 1978.
- [8] W. Puff, "PFPOSFIT: A new version of a program for analysing positron lifetime spectra with non-gaussian prompt curve," *Comput. Phys. Commun.*, vol. 30, pp. 359–368, 1983.
- [9] P. Kirkegaard, N. J. Pedersen, and M. Eldrup, "PC-PATFIT: A program package for fitting positron annihilation spectra on personal computers," in *Proc. 8'th Int. Conf. on Positron Annihilation*, (Gent, Belgium), 1988. World Scientific 1989.
- [10] 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.
- [11] 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.

- [12] 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.
- [13] 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.
- [14] 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.
- [15] H. Sormann, P. Kindl, and G. Reiter, “Numerical evaluation of disturbed positron lifetime spectra,” in *Proc. 8th Int. Conf. on Positron Annihilation*, (Gent, Belgium), 1988. World Scientific 1989.
- [16] 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. 8th Int. Conf. on Positron Annihilation*, (Gent, Belgium), 1988. World Scientific 1989.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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), 1988. World Scientific 1989.
- [23] 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.

- [24] 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.
- [25] 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.
- [26] 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.
- [27] L. C. Smedskjaer and M. J. Fluss, "Experimental methods of positron annihilation for the study of defects in metals," *Methods of Experimental Physics*, vol. 21, pp. 77–145, 1983.
- [28] O. E. Mogensen and M. Eldrup, "Vacancies in pure ice studied by positron annihilation techniques," *J. Glaciology*, vol. 21, pp. 85–99, 1978.
- [29] K. O. Jensen, M. Eldrup, B. N. Singh, and M. Victoria, "Helium bubbles in aluminium studied by positron annihilation: determination of bubble parameters," *J. Phys. F.: Met. Phys.*, vol. 18, pp. 1069–1089, 1988.
- [30] O. E. Mogensen, "Solvated positron chemistry. II. The reaction of hydrated positrons with Cl^- , Br^- and I^- ions," *Chem. Phys.*, vol. 37, pp. 139–158, 1979.
- [31] P. C. Jain, M. Eldrup, N. J. Pedersen, and J. N. Sherwood, "Positron annihilation in pivalic acid. Temperature dependence of angular correlation curves," *Chem. Phys.*, vol. 106, pp. 303–313, 1986.
- [32] "Programming Language FORTRAN." American National Standards Institute, Inc, 1430 Broadway, New York, New York 10018, 1978.
- [33] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPACK, User's guide*. Philadelphia: SIAM, 1979.
- [34] "MetaWINDOW Reference Manual Fortran." METAGRAPHICS SOFTWARE CORPORATION, 4575 Scotts Valley Drive, Scotts Valley, CA 95066, 1986.
- [35] P. Kirkegaard, "A FORTRAN IV version of the sum-of-exponential least-squares code EXPOSUM," Tech. Rep. Risø-M-1279, Risø National Laboratory, DK-4000 Roskilde, Denmark, 1970.
- [36] P. Kirkegaard, "Some aspects of the general least-squares problem for data fitting," Tech. Rep. Risø-M-1399, Risø National Laboratory, DK-4000 Roskilde, Denmark, 1971.

- [37] P. Kirkegaard and M. Eldrup, “The least-squares fitting programme POSITRONFIT: principles and formulas,” Tech. Rep. Risø-M-1400, Risø National Laboratory, DK-4000 Roskilde, Denmark, 1971.
- [38] L. Kaufman, “A variable projection method for solving separable nonlinear least squares problems,” *BIT*, vol. 15, pp. 49–57, 1975.
- [39] 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.
- [40] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *J. Soc. Ind. Appl. Math.*, vol. 11, pp. 431–441, 1963.
- [41] K. Levenberg, “A method for the solution of certain nonlinear problems in least squares,” *Quart. Appl. Math.*, vol. 2, pp. 164–168, 1944.
- [42] 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.
- [43] 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.
- [44] J. Spanier and K. B. Oldham, *An Atlas of Functions*. Washington: Hemisphere Publishing Corporation (a subsidiary of Harper & Row, Publishers, Inc.), 1987.

Bibliographic Data Sheet

Title and author(s)

PATFIT-88: A DATA-PROCESSING SYSTEM FOR POSITRON ANNIHILATION SPECTRA ON MAINFRAME AND PERSONAL COMPUTERS

Peter Kirkegaard, Niels Jørgen Pedersen, and Morten Eldrup

ISBN	Dept. or group
87-550-1491-7	ITA
Date	
February 1989	

Groups own reg. number(s)	Sponsorship
---------------------------	-------------

Project/contract No.

Pages	Tables	Illustrations	References
121	0	5	44

Abstract (Max. 2000 char.)

A data processing system has been developed for analyzing positron annihilation lifetime and angular correlation spectra on mainframe and Personal Computers (PCs). The system is based on the PATFIT programs previously developed for use on mainframe computers. It consists of the three fitting programs POSITRONFIT, RESOLUTION and ACARFIT and three associated programs for easy editing of the input data to the fitting programs, as well as a graphics program for the display of measured and fitted spectra. They can be used directly on any IBM-compatible PC. The PATFIT-88 software is available from Risø National Laboratory.

Descriptors

ANGULAR CORRELATION; ANNIHILATION; COMPUTER PROGRAM DOCUMENTATION; CORRELATIONS; DISPLAY DEVICES; LEAST SQUARE FIT; LIFETIME; MICROCOMPUTERS; MINIMIZATION; NON-LINEAR PROBLEMS; P CODES; POSITRONS; REGRESSION ANALYSIS; SPECTRA UNFOLDING

Available on request from:

Information Service Department, Risø National Laboratory
(Afdelingen for Informationsservice, Forskningscenter Risø)
P.O. Box 49, DK-4000 Roskilde, Denmark
Phone (+45) 46 77 46 77, ext. 4004/4005 · Fax (+45) 46 77 40 13
E-mail: risoe@risoe.dk